Constructing a reference standard for sports science and clinical movement sets using IMU-based motion capture technology

Thomas Jamin Gilbert

A dissertation submitted for the degree of

Doctor of Philosophy

University College London

Department of Electronic and Electrical Engineering

Under the supervision of

Professor Sally Day

ACKNOWLEDGEMENTS

Firstly, I would like to express my sincerest gratitude to University College London and my supervisor, Professor Sally Day, for giving me this amazing opportunity and supporting me through this journey.

I am also thankful to my friends and family for putting up with me throughout this never-ending PhD. Especially, my wonderful mother who helped me push through the darkest of times when I fully considered becoming a banker.

Above all, I am eternally grateful to my loving girlfriend.

Ilana, if it wasn’t for your constant support, your encouragement and foremost, your humour I would never have finished this PhD. You are the best.

DECLARATION

I, Thomas Jamin Gilbert declare the work presented in this is my own. The information derived from other sources has been acknowledged in the text and a list of references provided.

____

Signature

____

Date

ABSTRACT

Motion analysis has improved greatly over the years through the development of low-cost inertia sensors. Such sensors have shown promising accuracy for both sport and medical applications, facilitating the possibility of a new reference standard to be constructed.

Current gold standards within motion capture, such as high-speed camera-based systems and image processing, are not suitable for many movement-sets within both sports science and clinical movement analysis due to restrictions introduced by the movement sets. These restrictions include cost, portability, local environment constraints (such as light level) and poor line of sight accessibility.

This thesis focusses on developing a magnetometer-less IMU-based motion capturing system to detect and classify two challenging movement sets: Basic stances during a Shaolin Kung Fu dynamic form, and severity levels from the modified UPDRS (Unified Parkinson's Disease Rating Scale) analysis tapping exercise.

This project has contributed three datasets. The Shaolin Kung Fu dataset is comprised of 5 dynamic movements repeated over 350 times by 8 experienced practitioners. The dataset was labelled by a professional Shaolin Kung Fu master. Two modified UPDRS datasets were constructed, one for each of the two locations measured. The modified UPDRS datasets comprised of 5 severity levels each with 100 self-emulated movement samples. The modified UPDRS dataset was labelled by a researcher in neuropsychological assessment.

The errors associated with IMU systems has been reduced significantly through a combination of a Complementary filter and applying the constraints imposed by the range of movements available in human joints. Novel features have been extracted from each dataset. A piecewise feature set based on a moving window approach has been applied to the Shaolin Kung Fu dataset. While a combination of standard statistical features and a Durbin Watson analysis has been extracted from the modified UPDRS measurements.

The project has also contributed a comparison of 24 models has been done on all 3 datasets and the optimal model for each dataset has been determined. The resulting models were commensurate with current gold standards.

The Shaolin Kung Fu dataset was classified with the computational costly fine decision tree algorithm using 400 splits, resulting in: an accuracy of 98.9%, a precision of 96.9%, a recall value of 99.1%, and a F1-score of 98.0%. A novel approach of using sequential forward feature analysis was used to determine the minimum number of IMU devices required as well as the optimal number of IMU-devices.

The modified UPDRS datasets were then classified using a support vector machine algorithm requiring various kernels to achieve their highest accuracies. The measurements were repeated with a sensor located on the wrist and finger, with the wrist requiring a linear kernel and the finger a quadratic kernel. Both locations achieved an accuracy, precision, recall, and F1-score of 99.2%. Additionally, the project contributed an evaluation to the effect sensor location has on the proposed models.

It was concluded that the IMU-based system has the potential to construct a reference standard both in sports science and clinical movement analysis. Data protection security and communication speeds were limitations in the system constructed due to the measured data being transferred from the devices via Bluetooth Low Energy communication. These limitations were considered and evaluated in the future works of this project.

IMPACT STATEMENT

Motion analysis is a significant field of research with funding coming from the medical, the entertainment and the sporting industry. While current gold standards are capable of addressing many motion analysis applications, as the field continues to expand the limitations of these gold standard have become more apparent.

The research recorded in this thesis outlines the procedure required when constructing a new reference standard. Portability and price amongst other aspects of the current gold standards were found to be limitations for many applications. Therefore, an IMU-based solution focusing on these aspects was constructed and compared against the gold standards.

Outlining the procedure to constructing a reference standard using machine learning will allow the process to be replicated by future academics and non-academics alike. Therefore, increasing the number of potential applications feasible by the new standards.

The reference standard developed in this thesis could be used in several applications both in sports science and in the clinic. Many of these applications stem from the constructed system not requiring any external devices or information, allowing usage both indoors and outdoors. The three-dimensional data would allow a professional to precisely monitor the user's movement. Depending on the professional the following application in sports science and in the clinic would be possible:

- Outpatients for physiotherapists
- Teaching in sports
- Injury monitoring in a sporting environment and quotidian environment

Within this thesis a collaboration with a researcher in neuropsychological assessment, Sara Pisani, has allowed the system to be tested on an emulation of the modified-UPDRS tapping task. The test which usually requires the Parkinson's disease (PD) patient to come to the clinic can be done at home using the system described in this thesis.

To determine the effectiveness of the system in a sporting environment, a collaboration with a Shaolin Kung fu school allowed a movement set to be constructed and evaluated by professional Shaolin Kung Fu practitioners.

Acronym Table/List of Abbreviations

AUC	Area Under the Curve
BIF	Biologically Inspired Features
DOF	Degree of Freedom
DTW	Dynamic Time Warping
EKF	Extended Kalman Filter
FN	False Negative
FP	False Positive
GPS	Global Positioning Systems
GUI	Graphic User Interface
HMM	Hidden Markov Model
I2C	Inter-Integrated Circuit
IC	Integrated Circuit
IMU	Inertia Measurement Unit
IO	Input Output
KNN	K-Nearest-Neighbours
LED	Light Emitting Diode
MDS	Movement Disorder Society
MEMS	Micro-Electro-Mechanical System
MMR	MetaMotionR
OMC	Optical Motion Capture
PCA	Principal Component Analysis
PD	Parkinson’s Disease
RBF	Radial Basis Function
RMSE	Root Mean Square Error
ROC	Receiver Operating Characteristics
ROM	Range of Motion
RPM	Rotation per Minute
RTC	Real Time Clock
SPI	Serial Peripheral Interface
SVM	Support Vector Machine
TN	True Negative
TP	True Positive
UKF	Unscented Kalman Filter
UPDRS	Unified Parkinson’s Disease Rating Scale

TABLE OF CONTENTS

List of tables

List of figures

1 INTRODUCTION

The biomechanics of human movement has attracted interest from the earliest days of science and medicine. Even before physicians began to get involved in the way humans moved, society has always been fascinated about how we move our bodies. This obsession can in fact be considered a survival instinct; from the early days of humanity judgements had to be made to decide who should hunt and who should gather. Although primitive, these judgements were possibly some of the first movement analyses made. Moving forward several thousand years, mankind begins to develop sports. Modern day sports can be considered one of the largest areas of research when talking about movement analysis, with it even having its own subcategory of research, Performance Analysis. Performance Analysis is used by athletes to increase their understanding and improve their tactics, techniques, and movements. Sports can be considered one of the first analytical methods used to assess a person's movement. Some historians have suggested that sports were first used in the military to determine if an individual was useful for service [1]. The next big development for movement analysis was when physicians began to study it, forming the basis for modern movement analysis.

Modern movement analysis started with clinical observation of patterns of movement. Over time observation transitioned to measurements and as time progressed so did the tools being used to make these measurements. As technology advanced the tools being used progressed to video recordings allowing movement to be viewed repeatedly and in slow motion. Today, technology allows 3D motion capture where movement can be studied in greater detail and allows objective measurements to be made. Movement analysis has many applications including sports science, medical research, and the clinical management of patients with movement disorders [2].

Movement analysis is enhanced using objective data that describes a person's movement. In a clinical setting this data is used alongside medical examinations to give physiotherapists a more informed diagnosis of movement disorders. An example of an established motion capture technique used in the clinic is the Gait Analysis Laboratory, with a devoted a journal (Gait and Posture) and a society (Gait and Clinical Movement Analysis Society) [3]. Clinical gait analysis is used to help understand pathology that affects gait [4]. Pathologies such as: cerebral palsy, spina bifida, stroke, amputees, and Parkinson's disorder. The resulting information can be used in making decisions about the patient's diagnosis and treatment.

Research into human movement is greatly enhanced using an objective tool such as movement analysis, allowing repeated assessment under different conditions. It encompasses many areas including fundamental studies of muscle activity and function, as well as more applied areas such as rehabilitation, development of prostheses and ergonomics. As mentioned previously, sport and exercise science are another major area with studies of human movement in dance, martial arts and other activities.

While there are many areas where gold standards like the Gait Analysis laboratory have been established, there are still many examples in both the clinic and in sports where no gold standard is available. A Rutjes et al. have done a review of clinical tests where no gold standard is available and found that in the majority of cases an attempt to construct a reference standard is preferred [5]. In these cases, professionals in the area must use their knowledge of the field to make educated analysis of the clinical test. In many of the cases reviewed a gold standard was not available because the tests were either new or small (low funding or low number of test results/patients). In these cases, some attempts have been made to adjust similar tests' gold standards and results to form a paradigm.

However, for the majority of the cases reviewed this was not possible resulting in an attempt to construct a reference standard.

Rutjes et al. determined that the attempts to construct a reference standard all following the following steps of logic [5]. Step one, a deterministic predefined rule must be established, in the case of movement analysis this rule would be predefined movement sets. Step two, a consensus procedure amongst experts must be made to categorise the tests within the predefined rules. Step three, a statistical model based on these categorised tests must be established allowing future tests to be compared against, i.e. constructing a reference standard.

The main challenge in creating these reference standards is the process of moving from step one to step three. The issue here is recording the data from the predefined tests accurately enough that a well-developed statistical model is able correlate the data to the expert consensus. In the case of this project to record these tests or performances (in the case of sports science) a motion capturing system must be used. In section 1.1, a quick review of motion capturing systems has been done, outlining their limitations and comparative accuracies.

Before reviewing the available motion capturing systems available a movement set without a gold standard must be chosen. In fact, to thoroughly test the system developed in this project, two movement sets have been chosen, each presenting their own challenges. These movement sets have been specifically chosen to test the proposed system's flexibility and highlight which parts of the system can be generalised to any movement set.

The first movement set has been selected from the Movement Disorder Society's modified Unified Parkinson's Disease Rating Scale (UPDRS) assessment; more specifically the finger tapping assessment [6]. This movement set has been chosen as it is a clinical analysis concerned with quality of movement (QOM) for one location of the body. A lot of work has been done with the old UPDRS assessments giving some standards to compare this project against.

The second movement set consists of the five basic stances of Shaolin Kung Fu within the form Wu Bu Quan. This movement set has been chosen as it is a sports science analysis concerned with precise movement recognition of the entire body. Not much research has been done within in this area of sports science, however, similar martial arts movement sets have been analysed extensively.

Detailed descriptions of both movement sets will be given in chapter 2.

1.1 General background

The following descriptions of motion capturing systems and their associated limitations has been adapted from van der Kruk et al. state-of-the-art literature review on human motion capture systems [7].

1.1.1 Optoelectronic measurement system (OMS)

An OMS detects light and uses this detection to estimate the 3D position of a marker via time-of-flight triangulation. The accuracy of the systems is dependent on the following parts of the experimental set-up: the locations of the cameras relative to each other, the distance between the cameras and the markers, the position, number and type of the markers in the field, and the motion of the markers within the capture volume [8]. This capture volume is dependent on the number of cameras, the field of view and the required movement resolution. Areas above 800m^2 have been recorded with 24 cameras using the Vicon MX13 measurement system [9]. This number of cameras results in significant practical difficulties regarding cost, portability, calibration, synchronisation, labour, and set-up. Further limitations of the system are the necessity of line-of-sight, which means that the data output will be interrupted when the cameras lose sight of the markers [10] [9]. Furthermore, the systems are highly sensitive to alterations in the set-up, e.g. due to accidental shifting of a camera [11]. The systems are mostly used in dark areas (indoors) because bright light interferes with the measurements [9].

1.1.2 Electromagnetic measurement system (EMS)

EMS find the unknown positions of the measurement transponders by means of time-of-flight of the electromagnetic waves travelling from the transponder to the base stations [12]. EMS provides large capture volumes but are less accurate than OMS [7]. Unlike an OMS, no line-of-sight is necessary to find the positions of the transponders; also, the human body is transparent for the field applied [13]. Limitations of the system, related to the experimental set-up, are the sensitivity for ferromagnetic material in the environment, which decrease the accuracy of the data [14]. When the distance between the base station and the transponder is increased, noise increases and the quality of the signal decreases [14]. EMS have low sample frequencies which is a limitation when considering sports analysis. The sample frequency is inversely proportional to the number of markers used.

1.1.3 Image processing system (IPS)

IPS generally have better accuracy compared to the EMS, and an improved range when compared to the OMS. In IPSs captured films or photos are digitally analysed. Unlike the other motion capturing methods described this method does not require markers, using high quality cameras and computer vision algorithms to perform the motion capturing. This marker-less tracking can be a big advantage in sports, such as for event-detection [15]. IPSs have the following drawbacks: real-time recognition is computationally costly, high quality camera with high resolution and frame rate is required. The accuracy is also dependent on the experimental set-up, namely the position of the camera/s in relation to the object trajectory, the properties of the object in relationship to the background and the number of cameras [16].

1.1.4 Inertia measurement unit (IMU)

An IMU is a module consisting of an accelerometer, gyroscope, and often a magnetometer. By combining the information from the accelerometer, acceleration, with the data from the gyroscope, angular velocity, the orientation of the device can be determined [17]. The magnetometer can be combined with the inertia sensors to track the magnetic north, to determine the heading of the IMU. There are many commercially available IMUs on the market.

As stand-alone system, the device cannot determine its (global) position, and therefore is not added to the chart. In principle, the accelerometer could be used to determine position by performing a double integration, but the data will suffer from large integration drifts. The position in global space can for example be estimated when an IMU is combined with a rigid-body model of a human [18] [19]. Hereby the IMUs are placed on body segments to determine the global orientation. IMUs do not have a base station and are therefore the most mobile of all available measurement systems. Additionally, the system is capable of detecting very rapid motion [20] and is non-invasive for the user, which makes it an attractive system in sports (e.g. gymnastics [20], swimming [21]. A drawback to using magnetometers with the inertia sensors is that the system is susceptible to measurement errors due to nearby ferromagnetic interference. Moreover, the sensor-fusion algorithms are sensitive to linear accelerations.

1.2 REQUIREMENT ANALYSIS

The requirements of this project have been established by determining the limitations of current gold standards whilst specifically considering the two movement sets chosen for this project. Some of the aims will be more relevant to the Parkinson's modified UPDRS movement set while others the Shaolin Kung Fu basic stances movements set when this is so it will be indicated in the parentheses.

Requirements:

1. **Requires no information of the recorded environment.** This includes the levels of ferromagnetic disturbance, size of the working environment, ambient light levels, and ambient sound levels.
2. **Cost effective.** *(Shaolin Kung Fu movement set)* No budget has been established for such tests by the associated Shaolin school however considering their funding current gold standards are too expensive: Vicon's starting price of $12,800 [22] and Xsens' starting price of $12,500 [23]. The reference standard constructed in this project should minimise costs both for the hardware used to capture the movements and to postprocess/analysis them. A budget of approximately $2000 (£1500) has been suggested by the head of the Shaolin Kung Fu school.
3. **Low computational cost.** Considering both movement sets will not inherently have access to high computational power, it is important that the postprocessing done in this project requires only low processing power. This only refers to the processing required to use the system and not the processing required to construct the system. Constructing the system is a one-off task and can either: be delegated to a 3rd-party host if the budget permits it or done sequentially on a weaker CPU as the extended processing time is less impactful due to it being one-off.
4. **Non-invasive on the movement set.** Collins et al. have determined a large factor to many gold standards not being applicable to clinical tests is that the measurements are invasive to the tests, requiring the tests to adapt to the gold standard [24]. Therefore, the reference standard constructed in this project should not have this drawback.
5. **Portable.** *(Parkinson's modified UPDRS movement set)* A large benefit of having a reference standard over an expert's opinion is that an analysis can be done outside the clinic (in the case of the Parkinson's movement set). Therefore, the system should be fully portable allowing the movement data to be recorded on a secure cloud storage and accessed by the clinic for postprocessing and review. An aspect of the system's portability is its usability. It is not enough for the system to be physically portably; it should also be designed to allow non-experts (patients or sports people) to use it with limited guidance.
6. **Flexible for a variety movement sets.** Reference standards can be designed specifically to work for one task only however this sort of approach is inefficient engineering practise. The two movement sets have been selected to test that the constructed reference standard has not designed specifically for one approach but instead can be adapted to work for a variety of movement sets.
   a. The Parkinson's movement set should test the following:
      i. The effectiveness of using a single point of interest for motion analysis.

        - **ii.** To distinguish between the same movements with different quality of motion.
    - **b.** The Shaolin Kung Fu's movement set should test the following:
        - **i.** The effectiveness of using multiple points of interest for motion capture.
        - **ii.** The ability to recognise certain motions within a sequence of movements.

7. **Data remains secure.** *(Parkinson's modified UPDRS movement set)* The data collected for the Parkinson's movement set will be considered Protected Health Information (PHI) meaning it will have to comply with the Health Insurance Portability and Accountability Act (HIPAA) [25]. This act requires the data to be secure during measurements and storage alike.

1.3 THESIS APPROACH

By comparing the requirements of the project and the limitations highlighted in section 1.2, certain motion capturing systems have been eliminated as an option for this project.

The first aim of environmental awareness, in fact, eliminates many options already. The performance of both OMSs and EMSs rely heavily on the working environment, when considering light level and ferromagnetic interference respectively. While the working environment can have some impact on IPSs and IMUs including magnetometers.

The second aim of cost effective, eliminates OMSs and IMU fusion systems, such as Xsens [23]. Depending on the motions being analysed IPSs can also become expensive.

The aim to have the system portable, eliminates OMSs and EMSs as they require many external components to be setup around the analysis space. IPSs only require a single camera to make its measurements, however the orientation of the action in relation to the camera is important and may be a challenge to some users to do correctly. IMUs are portable, can be plug and play, and would only require the user to place the sensor on the location of interest (the wrist or finger in the case modified UPDRS movement set).

Considering all of this, it has been decided to work with IMUs without using magnetometers. This approach does not contradict any of the project's aims. The aims that will be hardest to achieve when considering an IMU setup are low computational cost and match accuracy with similar project's standards. Due to the noise and errors associated with standalone IMU systems complex fusion filters will be required to increase the accuracy of the system. These fusion filters can often be computationally costly; therefore, this project must weigh accuracy against cost to determine the best approach to meet the aims.

1.4 RESEARCH AIM AND OBJECTIVES

Aim: To construct and outline a system using magnetometer-less IMU motion capture and machine learning capable of motion recognition for the two movement sets chosen.

Objectives:

1. To identify the criteria for selecting a commercial IMU device for human motion capture.
2. To construct a new dataset for each of the two chosen movement sets large enough for machine learning analysis.
3. To select features for each dataset capable of full distinguishing the classes when applied to a machine learning model.

4. To determine the optimal machine learning model for each dataset such that it is commensurate with similar studies.
5. To determine the minimum number of IMU devices required for the Shaolin Kung Fu machine learning model to remain commensurate with similar studies.
6. To evaluate the affect location of the IMU device has on the modified UPDRS tapping test model. Meaning two datasets will be need for the modified UPDRS movement set.

1.5 Related work

Many studies have been published on motion capture in both sport science and clinical tests, however it has only been in recent years that IMU-based techniques have been developed. This can clearly be seen in *Pueo and Olmedo's* literature review regarding motion capture technology for sport performance analysis [26]. Of the 81 papers reviewed only 9 used IMU-based technology, compared to 53 that used OMC technology.

Whilst searching for publications that have analysed similar challenging movement sets the following five papers were found to set a standard to be compared against. Three of the papers (*Bianco*, *Samiullah et al.* and *Hachaj*) have used challenging martial arts movement sets which will be the standard this project compares against for the Shaolin Kung Fu movement set. While the other two papers (*Printy et al.* and *Huo et al.*) have used datasets based on the UPDRS tests. Below are summaries and critical analyses of these five papers.

- *Bianco* proposed a system made up of four steps [27]. The first step is to convert the depth data received by the Kinect into a global skeleton representation. This step is used to map the skeleton data to a set of features that enable effective recognition of karate movements. The second step is the pose classification, that models each move as a sequence of key poses, this achieved using a one-versus-all multi-class Support Vector Machine (SVM). The third step uses Dynamic Time Warping (DTW) temporally aligns the incoming move with the set of exemplar moves and recognises which of them has been performed. Once the move has been recognised, the fourth step gives a score representing how well the athlete performed the movement compared to the exemplar one. This score is based on the distances measured during the DTW step as well as human judgments. Bianco tests their system both statically and dynamically on a set of different Karate punches, kicks and blocks. The experiment required a Microsoft Kinect to be setup in the dojo and for the athletes to position themselves in from of the Kinect such that their entire body was sensed. 10 athletes, including both males and females, of varying skill levels were selected for the experiment. The system obtained an accuracy of 97% in the case of static stances, and 96.25% in the case of sequences with different starting and ending stances.
    - Despite *Bianco's* proposed system using markerless motion capture via a Microsoft Kinect, it will still be compared against as it uses machine learning on features extracted from motion capture technology on a similarly challenging movement set as the Shaolin Kung Fu movement set. This project will show that a similar approach to *Bianco's* can be implemented for an IMU-based system.
- *Samiullah et al.* describes and implements a system which uses a single IMU device to capture 8 key positions of Brazilian Jiu-Jitsu [28]. The proposed system was tested on 3 practitioners. The single IMU device is mounted below the practitioner's sternum whilst keeping the orientation of the sensor consistent throughout the recordings. Standard

statistical features (mean, median, max/min, standard deviation, tilt, magnitude, root mean square, root sum square) are extracted from the raw measurements and fed into a Random Forest classifier. A video recording was used to label the data and a 10-fold cross-validation was used to validate the Random Forest. To reduce over-fitting and high number of false positive results, the Random Forest algorithm was combined a Multinomial Hidden Markov Model. The resulting system obtained an accuracy of 72%±20.

  - This project will expand on this research and extract similar statistical features however to windowed and quantise data rather than raw data allowing poses to be extracted from temporal data. Additionally, this project's proposed system will begin with 17 sensor locations and evaluate the minimum number of sensors required to classify the movement set.

- *Hachaj* proposes a method for improving human motion classification by applying bagging and symmetry to Principal Component Analysis (PCA)-based features [29]. The proposed classification method was evaluated on a motion capture recording dataset of martial arts techniques performed by professional karate practitioners. The recordings were done in a dojo which had been verified to have arbitrary magnetic interference. The dataset consisted of 360 recordings in 12 motion classes. The experimental results show that applying the proposed classifiers' bagging procedure increased the accuracy of the Nearest-Neighbour (NN) and Support Vector Machine (SVM) classifiers by more than 5% and 3%, respectively. It is concluded that symmetry information might be helpful in situations in which it is not possible to optimize the decision borders of the classifier (for example, when we do not have direct information about class labels). The proposed PCA-feature set uses nearly all the tracked joints. The proposed SVM model achieved an accuracy of 94%.
  - This project's proposed system will not use PCA-based features as this results in the effect of the sensor locations to be unclear due to the incomprehensibility of PCA-based features. Additionally, this project's proposed model will not use magnetometer data as this limits the working environment of the system.
- *Printy et al.* have developed an iPhone application that collects kinematic data from a small cohort of Parkinson's disease patients during guided movement tasks based on the UPDRS test and extracted quantitative features using signal processing techniques [30]. These features were used in a classification model trained to differentiate between overall motor impairment of greater and lesser severity using standard clinical scores provided by a trained neurologist. Using a support vector machine classifier, a classification accuracy of 94.5% was achieved under 6-fold cross validation, and several features were shown to be highly discriminatory between more severe and less severe motor impairment by area under the receiver operating characteristic curve (AUC > 0.85). Accurate classification for discriminating between more severe and less severe bradykinesia was not achieved with these methods.
  - Despite *Printy et al.* proposed system using the touchscreen of a mobile, it will still be compared against as it uses machine learning on features extracted from motions based on the old UPDRS test. Additionally, the proposed system will be able to distinguish the different levels of severity posed by the modified UPDRS test.
- *Huo et al.* have created a system using 3 IMU devices, one force sensor and 4 custom mechanomyography (MMG) sensors [31]. The system was tested in its capacity to predict Unified Parkinson's Disease Rating Scale (UPDRS) scores based on quantitative assessment

of bradykinesia, rigidity and tremor in PD patients. 23 PD patients were tested with the sensor system in parallel with exams conducted by treating clinicians and 10 healthy subjects were recruited as a comparison control group. Standard statistical features extracted from the IMUs and MMGs were applied to multiple machine learning algorithms. Results prove the system accurately predicts UPDRS scores for all symptoms (85.4% match on average with physician assessment) and discriminates between healthy subjects and PD patients (96.6% on average).

  - *Huo et al.* have used multiple sensors to successfully distinguish the UPDRS scores, including at least one MMG sensor. This project's proposed system will extract features from a single IMU device to distinguish the UPDRS score of emulated data.

1.6 Outline of the Thesis

The thesis begins with a detailed description of the movement sets in chapter 2. The chapter goes onto detailing the importance of orientation representation and gimbal lock. Finally, the chapter gives human joint limitations and describes how these can be used to put a constraint on the system for reducing error proneness.

In chapter 3 the theory of the functionality of MEMS sensors is outlined, with a heavy focus on the noise attributes of both MEMS accelerometers and gyroscopes. Possible methods for tackling the errors within MEMS sensors have been suggested. The chapter also introduces the IMU module that has been chosen for this project, giving its technical specifications.

Chapter 4 gives a brief literature review on possible methods for evaluating a motion capturing system's performance. Benchmarks, that can be used in conjugation with the IMU system allowing direct comparisons, are outlined. Methods focused on using precise actuators are also suggested within this Chapter.

Chapter 5 outlines fusion filtering techniques that can be used to reduce the stochastic noise described in chapter 2. The theory behind orthogonal filtering is summarised, giving the reader a comprehensive understanding of the available fusion filters. The filters will then undergo testing using the L-611 rotary stage; the results will then be compared to determine which filter meets the aims of the project most closely.

The experimental setup starting from the filtered raw data and ending with extracted features is given in chapter 6. The chapter starts with a summary of the setup before going into full detail for both movement sets separately.

The chapter is split into two starting with the modified UPDRS datasets and finishing with the Shaolin Kung Fu dataset. Both sets of features will be applied to several machine learning models using MATLAB's Classification app [32], the models will then be cross validated, and the features will be analysed giving a full understanding of the system's performance. The chapter then concludes with a discussion of the results presented, comparing them to related studies.

Finally, the thesis will conclude highlighting the main contributions of the work done in this project as well as the limitations of the work. This will be followed by looking back at the project's aim, objectives and requirements. The impact and usability of the system developed will be highlighted before outlining the future work for this project.

2 MOVEMENT DESCRIPTIONS

As mentioned previously two movement sets will be measured to encompass clinical movement analysis and movement analysis in sports science. To gain further understanding of the use of inertia-based analysis one dataset will be formed using multiple modules while the others will consist of a single module. In this chapter the chosen movements will be described in full. Following this the mathematics for Euler angles and quaternions orientations will be outlined, highlighting the importance of each representation. Finally human joint limitations will be described as well as how these limitations can be used to reduce errors in the system.

2.1 MOVEMENT SET DESCRIPTIONS

2.1.1 Shaolin Kung Fu

2.1.1.1 Introduction

The first set of movements will come from practitioners of Shaolin Kung Fu, these movements will form the dataset for the multiple modules sports science system. Shaolin Kung Fu was chosen due to the precise movements required. The practitioners used for the dataset have been studying the movements for several years and are required to dynamically transition between them with little variation. Shaolin Kung Fu masters, official judges and high-level practitioners can distinguish these variations determining a movements success by a few degrees.

2.1.1.2 Basic Stances

There are 5 basic stances in Shaolin Kung Fu: Ma Bu, Gong Bu, Xie Bu, Pu Bu and Xu Bu. Early practitioners are traditionally required to be able to hold each stance for 5 minutes before they can start their first form. The 5 basic stances focus on building leg strength and flexibility, each with a distinct “shape”. There are some variations in the basic stances depending on the style of the school teaching them, however these are only minor and could be considered by the system being developed. The descriptions of the stances below have been given by experts at the associated Shaolin Kung Fu school as well as some literature about Chinese Wushu [33].

Ma Bu (Horse Stance)

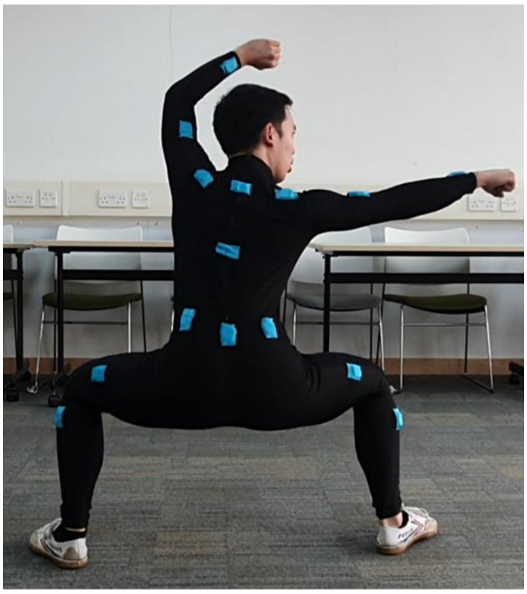


Figure 1 Shaolin Kung Fu practitioner performing Ma Bu stance during the Wu Bu Quan form.

The figure (Figure 1) above should be used in conjugation with the description below to facilitate in the understand of the Ma Bu stance.

- The upper body should be near vertical, leaning slightly forward to keep centre of gravity above the toes.
- Prevent back from curving, keeping the lower, middle and upper back on the same plane.
- Stand with your feet approximately two shoulder widths apart and keep your feet approximately 15 degrees outward.
- Squat down until thighs are approximately horizontal resulting in a right angle between the thighs and calves.
- Open hips allowing the knees to sit above the feet.
- The lower body should be symmetrical.
- The arms and head position change depending on the form being executed.

Gong Bu (Bow Stance)

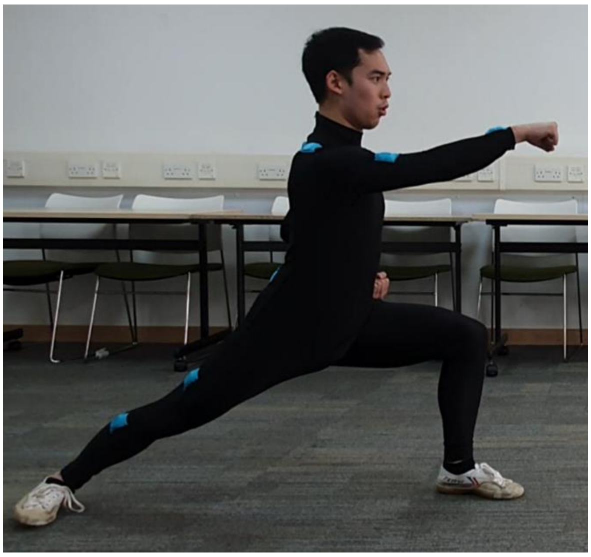


Figure 2 Shaolin Kung Fu practitioner performing Gong Bu stance during the Wu Bu Quan form.

The figure (Figure 2) above should be used in conjugation with the description below to facilitate in the understand of the Gong Bu stance.

- The upper body should be vertical keeping shoulders above the hips.
- Prevent back from curving, keeping the lower, middle and upper back on the same plane.
- The shoulders and hips should be perpendicular to the front leg.
- The feet should be two shoulder widths apart.
- The back foot should be twisted 45 degrees towards the front leg and the front foot should be twisted inward 15 degrees. Both heels have stay on the ground.
- The back leg is fully extended backwards.
- The front leg is bent at a right angle such that the thigh is horizontal and the calf vertical above the heel.
- The back leg is fully extended.
- The arms position change depending on the form being executed.

Xie Bu (Rest Stance)

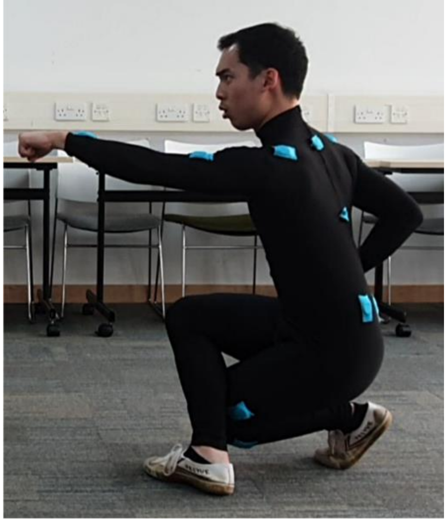


Figure 3 Shaolin Kung Fu practitioner performing Xie Bu stance during the Wu Bu Quan form.

The figure (Figure 3) above should be used in conjugation with the description below to facilitate in the understand of the Xie Bu stance.

- The upper body should be vertical keeping shoulders above the hips.
- Prevent back from curving, keeping the lower, middle and upper back on the same plane.
- Sit down with the back knee under the front knee.
- The back knee may touch the ground however should not be rested on.
- Raise the heel of the back leg.
- The front foot should be 45 degrees outward.
- The arms and head position change depending on the form being executed.

Pu Bu (Flat Stance)

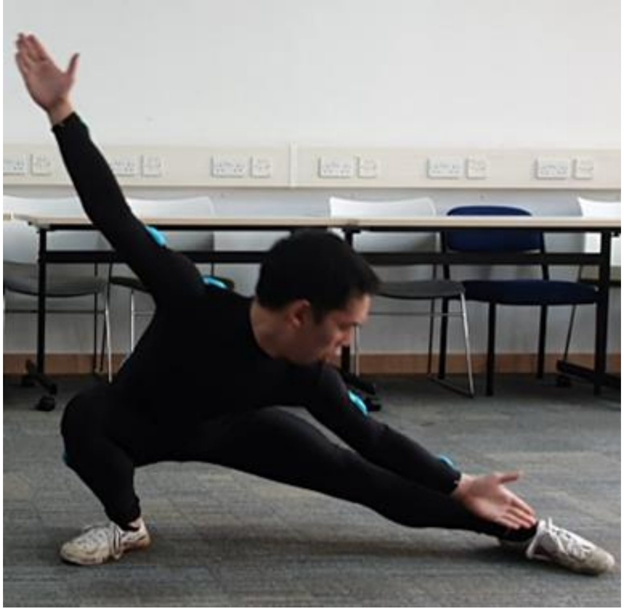


Figure 4 Shaolin Kung Fu practitioner performing Pu Bu stance during the Wu Bu Quan form.

The figure (Figure 4) above should be used in conjugation with the description below to facilitate in the understand of the Pu Bu stance.

- The upper body should lean forward to keep the centre of gravity over the toes.
- Prevent back from curving, keeping the lower, middle and upper back on the same plane.
- The feet should be two shoulder widths apart.
- The feet should be approximately 15 degrees outward
- Squat down on the back leg until your hips are below the back-knee's height. The back knee should be in line with the back foot.
- The hips should be fully open allowing the front leg to fully extend in line with the hips.
- The arms and head position change depending on the form being executed.

Xu Bu (Cat Stance)

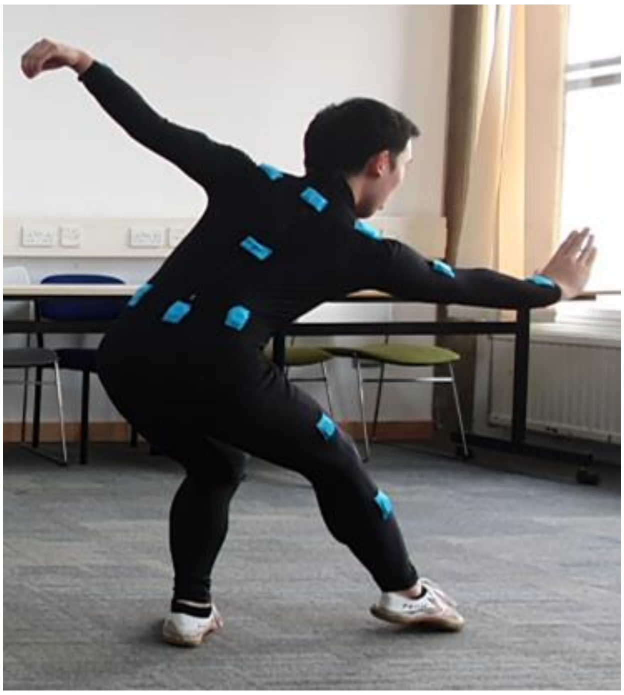


Figure 5 Shaolin Kung Fu practitioner performing Xu Bu stance during the Wu Bu Quan form.

The figure (Figure 5) above should be used in conjugation with the description below to facilitate in the understand of the Xu Bu stance.

- The upper body should be leaning forward to keep the centre of gravity above the back toes.
- Prevent back from curving, keeping the lower, middle and upper back on the same plane.
- The shoulders and hips be aligned facing 45 degrees to the back leg.
- Squat on your back leg such that the thigh is approximately horizontal to the ground and the knee is over the toes.
- The back foot should be 45 degrees outward with the heel on the ground.
- The front leg should be bent allowing the front toes to touch the ground. No weight should be placed on the front foot.
- The hips should be closed keeping the knees near one another.
- The arms and head position change depending on the form being executed.

2.1.2 Wu Bu Quan (5 Stance Form)

Wu Bu Quan is the first form taught to many Shaolin Kung Fu practitioners as it proves their ability to dynamically move through the five basic stances. To successfully execute the form the practitioner must transition through the five-stance holding each stance for a split second to prove stability. Figure 6 shows an action shot of a practitioner performing Wu Bu Quan.

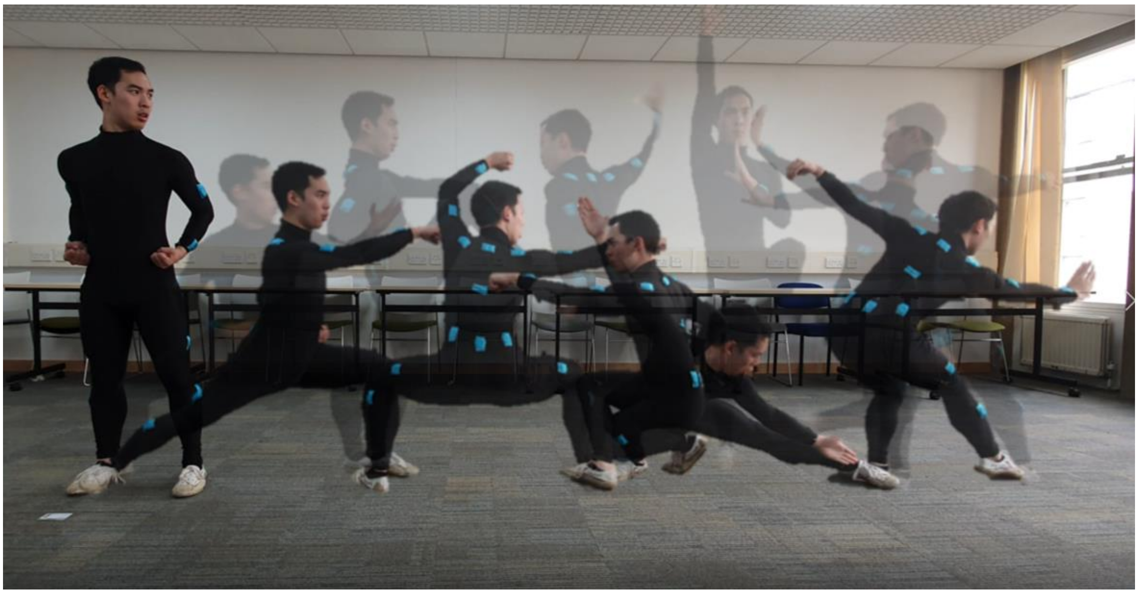


Figure 6 A Shaolin Kung Fu practitioner performing the first form taught to students at the associated Shaolin Kung Fu school, Wu Bu Quan.

2.1.3 Parkinson’s Disease

2.1.3.1 Introduction

The second set of movements will come from a study on Parkinson’s Disease (PD), these movements will form a clinical dataset using a single module for each of the two chosen locations. PD is a progressive nervous system disorder that affects movement. Symptoms start gradually, sometimes with a barely noticeable tremor in just one hand. Tremors are common, but the disorder also commonly causes stiffness or slowing of movement.

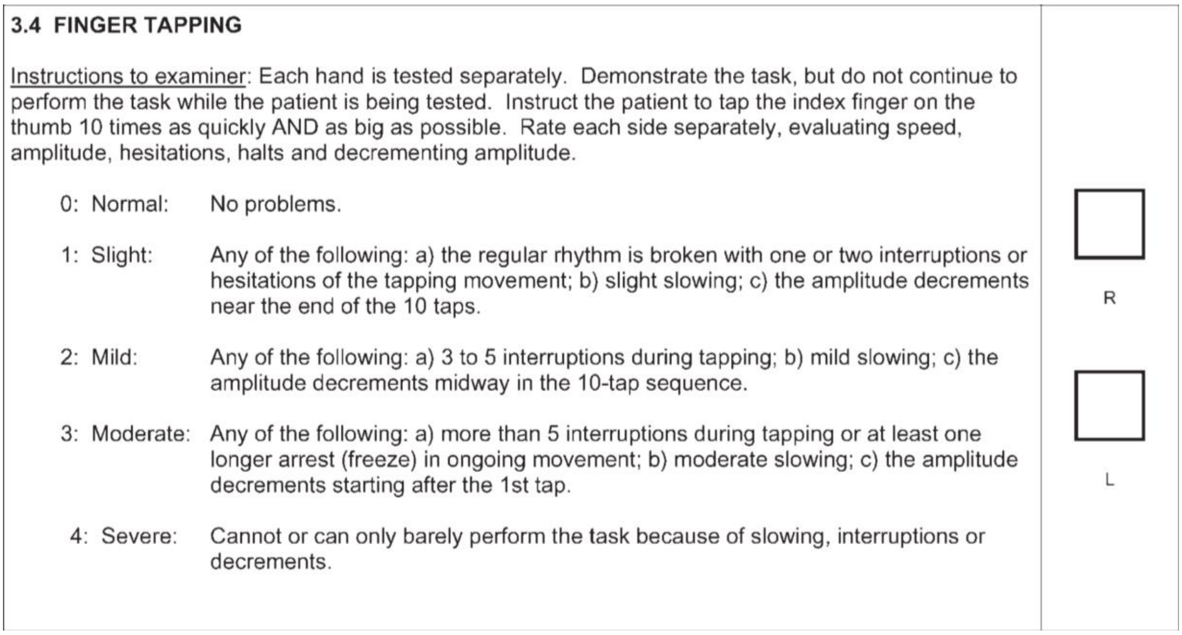
2.1.3.2 Movement Disorder Society - Unified Parkinson's Disease Rating Scale (MDS-UPDRS)

The Unified Parkinson’s Disease Rating Scale (UPDRS) was originally developed in the 1980s and has become the most widely used clinical rating scale for PD [6]. In 2001, the Movement Disorder Society (MDS) sponsored a critique of the UPDRS resulting in the formation of the MDS-UPDRS [6]. Both the original UPDRS and the modified MDS-UPDRS have an entire section on “Motor Examination” however the new MDS-UPDRS have specific instructions for the required tasks. The part in question has 18 different tasks for the patient to attempt and are scored separately with the integer scale 0-4 (0: Normal, 1: Slight, 2: Mild, 3: Moderate and 4: Severe).

2.1.3.3 Finger Tapping

The finger tapping task has been chosen from the 18 tasks in Part III of the MDS-UPDRS due to it being the first task with a specific part of the body being assessed, making it the ideal task to be performed with one module. The beginning of the task has the assessor demonstrate the exercise holding their hand out in front of them, as seen in Figure 8. As can be seen in Figure 7, the assessor must evaluate the patient's movement in the following ways, keeping in mind when these features change in the task:

- Speed and consistence
- Number of interruptions/hesitations
- Number of longer arrests/freezes
- Amplitude of the taps



3.4 FINGER TAPPING

Instructions to examiner: Each hand is tested separately. Demonstrate the task, but do not continue to perform the task while the patient is being tested. Instruct the patient to tap the index finger on the thumb 10 times as quickly AND as big as possible. Rate each side separately, evaluating speed, amplitude, hesitations, halts and decrementing amplitude.

0: Normal: No problems.

1: Slight: Any of the following: a) the regular rhythm is broken with one or two interruptions or hesitations of the tapping movement; b) slight slowing; c) the amplitude decrements near the end of the 10 taps.

2: Mild: Any of the following: a) 3 to 5 interruptions during tapping; b) mild slowing; c) the amplitude decrements midway in the 10-tap sequence.

3: Moderate: Any of the following: a) more than 5 interruptions during tapping or at least one longer arrest (freeze) in ongoing movement; b) moderate slowing; c) the amplitude decrements starting after the 1st tap.

4: Severe: Cannot or can only barely perform the task because of slowing, interruptions or decrements.

□ R

□ L

Figure 7 Instructions to the assessor for the finger tapping task of the MDS-UPDRS. [6]

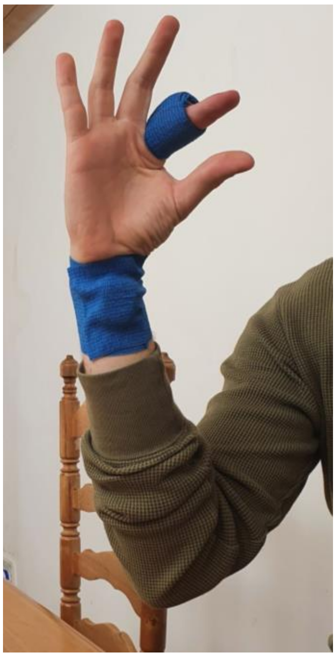


Figure 8 Example of the hand position used for the finger tapping task of the MDS-UPDRS.

2.2 Orientation Representations

Most IMU modules give multiple options for representing the inertia data they record raw accelerometer/gyroscope/magnetometer data, Euler angles, and Quaternions. This chapter will focus on understanding both Euler angles and Quaternions and their pros and cons. Euler angles are an intuitive and mathematically simple method to use however suffers from Gimbal lock which can cause the system to stop entirely. Quaternions are difficult to conceptualise, and the mathematics can get quite complicated, however do not suffer Gimbal lock.

2.2.1 Euler Angles

Euler angles provide a representation of 3D orientation of an object using a combination of three rotations about different axis. To intuitively understand the mathematics defining Euler angles the three rotations will each result in a new frame. These frames will be labelled during this explanation as “frame 0”, “frame 1”, “frame 2”, and “frame 3”.

Frame 0

The "frame 0" is an Earth set of axes that is used as a constant fixed reference. Note that the order of rotation when considering an orientation is important. For this explanation the first rotation will be about the Z axis (Yaw), followed by the Y axis (Pitch) and the final rotation will be about the X axis (Roll), refer to Figure 9. This combination has been chosen as it is the common method on IMU sensors. As this consists of rotations of three distinct axes it is technically not Euler angles but instead Tait-Bryan angles however few references make this distinction. There are six possible Tait-Bryan angles combinations (XYZ, XZY, YZX, YXZ, ZXY, ZYX) and six Euler angle combinations (ZXZ, ZYZ, YZY, YXY, XZX, XYX).

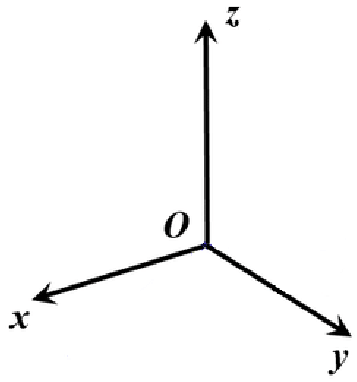


Figure 9 Euler angle calculation – Frame 0

Frame 1

As shown in Figure 10, yaw represents rotation about the frame 0 Z-axis by an angle γ. The yaw rotation produces a new coordinate frame where the z-axis is aligned with frame 0 and the x and y axes are rotated by the yaw angle γ. For this explanation the new frame is called frame 1. The orientation of frame 1 after yaw rotation is shown in Figure 10. The frame 1 axes are coloured blue, while frame 0 axes remain black.

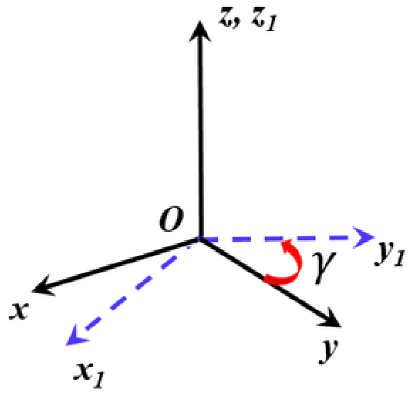


Figure 10 Euler angle calculation – Frame 1

Rotation of a vector from frame 0 to frame 1 can be performed by multiplying the vector by the rotation matrix.

$$R_0^1(\gamma) = \begin{pmatrix} \cos(\gamma) & \sin(\gamma) & 0 \\ -\sin(\gamma) & \cos(\gamma) & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{6.1}$$

Frame 2

Pitch represents rotation about Frame 0 y'-axis by an angle β as shown in Figure 11. For clarity, frame 0 axes are not shown. The frame 2 axes are shown in blue, and frame 1 axes are shown in black. It is important to note that pitch is NOT rotation about the frame 0 Y-axis.

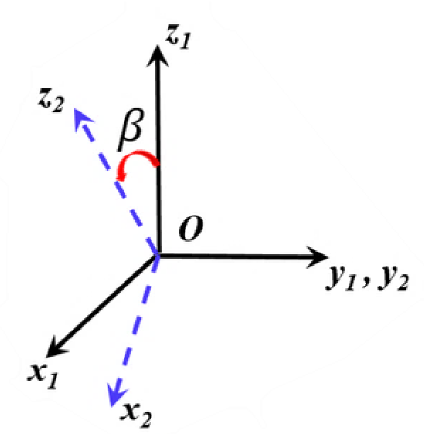


Figure 11 Euler angle calculation – Frame 2

The rotation matrix for moving from frame 1 to frame 2 is given by:

$$R_1^2(\beta) = \begin{pmatrix} \cos(\beta) & 0 & -\sin(\beta) \\ 0 & 1 & 0 \\ \sin(\beta) & 0 & \cos(\beta) \end{pmatrix} \tag{6.2}$$

The rotation matrix for moving from frame 0 to frame 2 consists simply of the yaw matrix multiplied by the pitch matrix:

$$R_0^2(\beta, \gamma) = R_1^2(\beta) R_0^1(\gamma) \tag{6.3}$$

Frame 3

Frame 3 is the final frame of the orientation calculation and therefore can be considered to the sensor orientation. This means frame 3 would line up with the MEMS sensor frame.

Frame 3 is obtained by performing a rotation by the angle α around frame 2 x_2-axis as shown in Figure 12. For clarity, frame 0 and frame 1 axes are not shown. Frame 2 axes are shown in black, while frame 3 axes are shown in blue.

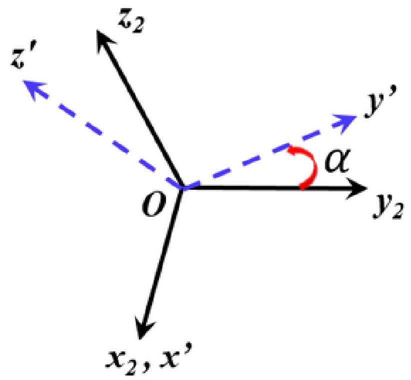


Figure 12 Euler angle calculation – Frame 3

The rotation matrix for moving from the vehicle-2 frame to the body frame is given by:

$$R_2^3(\alpha) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha) & \sin(\alpha) \\ 0 & -\sin(\alpha) & \cos(\alpha) \end{pmatrix} \tag{6.4}$$

The complete rotation matrix for moving from frame 0 to frame 3 is given by:

$$R_0^3(\alpha,\beta,\gamma) = R_2^3(\alpha)R_1^2(\beta)R_0^1(\gamma)$$

$$= \begin{pmatrix} \cos(\gamma)\cos(\beta) & \cos(\beta)\sin(\gamma) & -\sin(\beta) \\ \cos(\gamma)\sin(\alpha)\sin(\beta) - \cos(\alpha)\sin(\gamma) & \cos(\alpha)\cos(\gamma) + \sin(\alpha)\sin(\gamma)\sin(\beta) & \cos(\beta)\sin(\alpha) \\ \sin(\alpha)\sin(\gamma) + \cos(\alpha)\cos(\gamma)\sin(\beta) & \cos(\alpha)\sin(\gamma)\sin(\beta) - \cos(\gamma)\sin(\alpha) & \cos(\alpha)\cos(\beta) \end{pmatrix} \tag{6.5}$$

The rotation matrix for moving the opposite direction, from frame 3 to frame 0, is given by:

$$R_3^0(\alpha,\beta,\gamma) = R_0^1(-\gamma)R_1^2(-\beta)R_2^3(-\alpha)$$

$$= \begin{pmatrix} \cos(\gamma)\cos(\beta) & \cos(\gamma)\sin(\alpha)\sin(\beta) - \cos(\alpha)\sin(\gamma) & \sin(\alpha)\sin(\gamma) + \cos(\alpha)\cos(\gamma)\sin(\beta) \\ \cos(\beta)\sin(\gamma) & \cos(\alpha)\cos(\gamma) + \sin(\alpha)\sin(\gamma)\sin(\beta) & \cos(\alpha)\sin(\gamma)\sin(\beta) - \cos(\gamma)\sin(\alpha) \\ -\sin(\beta) & \cos(\beta)\sin(\alpha) & \cos(\alpha)\cos(\beta) \end{pmatrix} \tag{6.6}$$

2.2.1.1 Practical use of Euler angles

IMU modules inertia sensors are aligned to the body of the module, therefore if the earth frame or frame 0 is needed the outputs of the module must be converted from frame 3 to frame 0. This will give the system a reference for the modules orientation as well as its linear acceleration, allowing the system to gauge displacement as well as articulating movement compared to the earth axes.

Let a_s be the accelerometer data recorded by the sensor, and a_e be the earth frame acceleration, the following is true:

$$a_e = R_3^0(\alpha, \beta, \gamma)a_s \tag{6.7}$$

Vector a_e gives the measured acceleration of the module with respect to the earth frame however this is not the physical acceleration of the device but instead the device's acceleration plus acceleration due to gravity. Therefore, it is more accurate to saying the following if the physical acceleration of the device is desire. Let a_p be the physical acceleration of the device with respect to the earth frame and g be acceleration due to gravity, the following is true:

$$a_p = R_3^0(\alpha, \beta, \gamma)a_s + \begin{pmatrix} 0 \\ 0 \\ g \end{pmatrix} \tag{6.8}$$

Magnetometer data can be converted in the same way as accelerometer data, however gyroscope data cannot as it is measured as a rate and not a flat value.

2.2.1.2 Gimbal lock

Gimbal lock occurs when the orientation of the sensor cannot be uniquely represented using Euler Angles. The exact orientation at which gimbal lock occurs depends on the order of rotations used. On most IMU modules, the order of operations results in gimbal lock when the pitch angle is 90 degrees.

Intuitively, the cause of gimbal lock is that when the pitch angle is 90 degrees, yaw and roll cause the sensor to move in the same fashion.

An orientation sensor that uses Euler Angles will always fail to produce reliable estimates when the pitch angle approaches 90 degrees. This is a fundamental problem of Euler Angles and can only be solved by switching to a different representation method.

2.2.2 Quaternions

Quaternions provide an alternative measurement technique that does not suffer from gimbal lock. Quaternions are less intuitive than Euler Angles and the math can be a little more complicated.

A quaternion is a four-element vector that can be used to encode any rotation in a 3D coordinate system. Technically, a quaternion is composed of one real element and three complex elements, and it can be used for much more than rotations. In this application note we'll be ignoring the theoretical details about quaternions and providing only the information that is needed to use them for representing the attitude of an orientation sensor.

The orientation quaternion estimated by IMU modules orientation sensors encodes rotation from the earth frame (frame 0) to the sensor frame (frame 3). Like in the previous explanation the sensor frame is the coordinate frame that always remains aligned with the sensor. Unlike Euler Angle estimation, only the sensor frame and the earth frame are needed when quaternions are used for estimation.

Let the vector q_e^s be defined as the unit-vector quaternion encoding rotation from the earth frame to the sensor frame.

$$q_e^s = (a\ b\ c\ d)^T \tag{6.9}$$

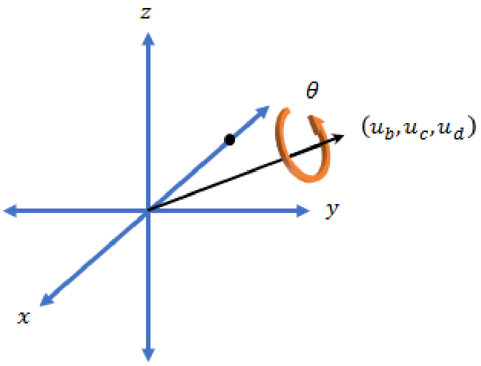


Figure 13 Rotational frame for Quaternion orientation representation

The elements b, c, and d are the vector part of the quaternion and can be thought of as a vector about which rotation should be performed. The element a is the scalar part that specifies the amount of rotation that should be performed about the vector part. Specifically, if θ is the angle of rotation and the vector $(u_b, u_c, u_d)^T$ is a unit vector representing the axis of rotation, refer to Figure 13, then the quaternion elements are defined as:

$$\begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix} = \begin{pmatrix} \cos(0.5\theta) \\ u_b \sin(0.5\theta) \\ u_c \sin(0.5\theta) \\ u_d \sin(0.5\theta) \end{pmatrix} \tag{6.10}$$

2.2.2.1 *Practical use of Quaternions*

The orientation quaternion can be used to rotate a vector,v_e, from the earth frame to the sensor frame using the following formula:

$$v_s = q_e^s \begin{pmatrix} 0 \\ v_e \end{pmatrix} (q_e^s)^{-1} \tag{6.11}$$

That is, a vector can rotate by treating it as a quaternion with zero real-part and multiplying it by the orientation quaternion and its inverse. The inverse of a quaternion is equivalent to its conjugate, which means that all the vector elements are negated. The rotation also uses quaternion multiplication, which has the following definition:

If $q_1 = (a_1 \quad b_1 \quad c_1 \quad d_1)^T$ and $q_2 = (a_2 \quad b_2 \quad c_2 \quad d_2)^T$ the $q_1 q_2$ is given by:

$$q_1 q_2 = \begin{pmatrix} a_1 a_2 - b_1 b_2 - c_1 c_2 - d_1 d_2 \\ a_1 b_2 + b_1 a_2 + c_1 d_2 - d_1 c_2 \\ a_1 c_2 - b_1 d_2 + c_1 a_2 + d_1 b_2 \\ a_1 d_2 + b_1 c_2 - c_1 b_2 + d_1 a_2 \end{pmatrix} \tag{6.12}$$

To rotate a vector from the body frame to the inertial frame, two quaternion multiplies as defined above are required. Alternatively, the attitude quaternion can be used to construct a 3x3 rotation matrix to perform the rotation in a single matrix multiply operation. The rotation matrix from the inertial frame to the body frame using quaternion elements is defined as:

$$R_e^s(q_e^s) = \begin{pmatrix} a^2 + b^2 - c^2 - d^2 & 2bc - 2ad & 2bd + 2ac \\ 2bc + 2ad & a^2 - b^2 + c^2 - d^2 & 2cd - 2ab \\ 2bd - 2ac & 2cd + 2ab & a^2 - b^2 - c^2 + d^2 \end{pmatrix} \tag{6.13}$$

Then the rotation from the inertial frame to the body frame can be performed using the matrix multiplication

$$v_s = R_e^s(q_e^s) v_e \tag{6.14}$$

Regardless of whether quaternion multiplication or matrix multiplication is used to perform the rotation, the rotation can be reversed by simply inverting the attitude quaternion before performing the rotation. By negating the vector part of the quaternion vector, the operation is reversed.

2.2.3 Converting from Quaternions to Euler Angle

The exact equations for converting from quaternions to Euler Angles depends on the order of rotations. Most IMU modules move from the earth frame to the sensor frame using first yaw, then pitch, and finally roll. This results in the following conversion equations:

$$\alpha = \arctan\left(\frac{2(ab + cd)}{a^2 - b^2 - c^2 + d^2}\right) \tag{6.15}$$

$$\beta = -\arcsin\big(2(\mathrm{bd} - \mathrm{ac})\big) \tag{6.16}$$

$$\gamma = \arctan\left(\frac{2(ad + bc)}{a^2 + b^2 - c^2 - d^2}\right) \tag{6.17}$$

When converting from quaternions to Euler Angles, the arctan2 function should be used instead of arctan so that the output range is correct. Note that when converting from quaternions to Euler Angles, the gimbal lock problem still manifests itself. The difference is that since the estimator is not using Euler Angles, it will continue running without problems even though the Euler Angle output is temporarily unavailable.

2.2.4 Converting from Euler angle to Quaternion

Comparing the orthogonal matrices of Euler angles and unit orientation quaternions the following can be said:

$$q = \begin{pmatrix} \cos\left(\frac{\alpha}{2}\right)\cos\left(\frac{\beta}{2}\right)\cos\left(\frac{\gamma}{2}\right) + \sin\left(\frac{\alpha}{2}\right)\sin\left(\frac{\beta}{2}\right)\sin\left(\frac{\gamma}{2}\right) \\ \sin\left(\frac{\alpha}{2}\right)\cos\left(\frac{\beta}{2}\right)\cos\left(\frac{\gamma}{2}\right) - \cos\left(\frac{\alpha}{2}\right)\sin\left(\frac{\beta}{2}\right)\sin\left(\frac{\gamma}{2}\right) \\ \cos\left(\frac{\alpha}{2}\right)\sin\left(\frac{\beta}{2}\right)\cos\left(\frac{\gamma}{2}\right) + \sin\left(\frac{\alpha}{2}\right)\cos\left(\frac{\beta}{2}\right)\sin\left(\frac{\gamma}{2}\right) \\ \cos\left(\frac{\alpha}{2}\right)\cos\left(\frac{\beta}{2}\right)\sin\left(\frac{\gamma}{2}\right) - \sin\left(\frac{\alpha}{2}\right)\sin\left(\frac{\beta}{2}\right)\cos\left(\frac{\gamma}{2}\right) \end{pmatrix} = \begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix} \qquad (6.18)$$

2.3 HUMAN MOVEMENT LIMITATION

A significant part of this project is limiting the range of results classified as possible by the system. By limiting the system, it is possible to adjust results that fall outside the identified range and label outlying results as errors. For this project the limits of the system are governed by the limits of human motion. The human joint limits have been given in Appendix A.

Demirdjian et al. have proposed a system that imposes kinematic constraints on motion capture data from previous temporal positions and joint limitations derived from 150,000 poses of training data [34]. Due to time constraints, it was decided not to implement *Demirdjian et al.* system instead published joint limitations would be simply used as hard constraints on the Shaolin Kung Fu dataset. This hard-constraints approach would simply label any samples outside the limits given by Human joint limits in Appendix A as corrupt data.

2.4 CHAPTER SUMMARY

This chapter has outlined the two movement sets that will be used in this project, indicating the description the datasets will be labelled against. The chapter continues to discuss the importance in choosing the correct orientation representation. Finally, the chapter describes the joint limitation technique that will be used to remove any potential outliers from the Shaolin Kung Fu dataset.

3 MEMS BACKGROUND

This chapter will outline the specifications of the IMU modules selected for this project, including the possible range of measurements, data formatting, memory and battery life. These are all important aspects to consider when designing a system. The chapter will then move onto the errors associated with MEMS sensors, giving an outline of the errors and their integrals, methods to measure the errors and compensation techniques.

3.1 INERTIAL MEASUREMENT UNIT (IMU) - MEMS SENSORS

MEMS or micro-machined electromechanical system sensors have become more and more popular over the last 30 years, seeing their way into most of global positioning systems (GPS) and smartphone devices. These devices are far more popular than their predecessors, such as the Vibratory gyroscope, due to the complexity of the manufacturing process before the MEMS sensors were introduced. By taking advantage of the chemical etching process used to create integrated circuits, the MEMS sensor does not require the intricate assembly techniques conventionally associated with mechanical sensors. In fact, by moving the assembly process to chemical etching many of the difficulties associated with manufacturing mechanical devices may be overcome, resulting in a much cheaper and faster manufacturing process. [35]

Another difference between MEMS sensors and their predecessors is the size, by reducing the number of parts and using chemical etching, the MEMS sensors could be made much smaller. This reduction in size has several benefits when considering the practical applications; however, it also affects the performance of the sensors; introducing addition errors and noise issues to those already associated with mechanical devices [36]. Including that standard to integrated circuits such as the thermal sensitivity concerns, from the working environment and possible leakage current. Methods have been developed to tackle these errors in the MEMS sensor's measurements, using both existing algorithms and algorithms specifically designed for MEMS sensors. These algorithms will be discussed thoroughly in Chapter 5.

Through the development of compensation techniques, the accuracy of MEMS sensors has increased such that they are on par with their precursors (i.e. ring laser gyroscope and fibre optic gyroscope), allowing, for gyroscope accuracies up to 0.01°/h angular rate measurements, and for accelerometers accuracies up to 1mg specific force measurement [37]. Such accuracies are not required for the aims of this project however shows a good indication of what is possible when using MEMS sensors.

The cost, accuracy and range of measurements of a MEMS sensor depends on the type of technology being used. Despite having the same manufacturing process, the technology for MEMS gyroscopes and MEMS accelerometers differs and therefore should be considered separately.

Most IMU modules include 3-axis magnetometers which can be a valuable asset when calculating absolute orientation however they are also very suspectable to noise. Magnetometers are easily distorted by ferromagnetic materials near the sensor. There are many techniques that can reduce this noise, however these techniques makes many assumptions about the environment around the sensor, such as the environment has an homogeneous magnetic field [38] or estimating the disturbance using a filter to compensate for it [39]. The problem with these methods is that it requires prior knowledge about the environment the sensors will be taking measurements, and considering the portability aim of the project it would not be possible to make such assumptions. Considering this and the fact that methods that do not rely on magnetometer data exists, it has been decided not to use magnetometer sensor to aid in the orientation calculations.

3.2 IMU Device Selection

Initially during the first two years of this project a new IMU module was being developed to meet the project's requirements. However, during the development stages of the IMU module several commercial modules became available with specifications that met the project's requirements, therefore the direction of the project shifted to making use of these commercial systems. The work done previously in developing a new IMU module can be seen in Appendix B.

During this project 7 fully integrated commercial IMU devices were compared against the criteria seen in Table 1.

Criteria	Reasoning
The device can record at least 20 minutes of data onboard.	The Shaolin Kung Fu movement set requires 5-20 seconds recording time plus a setup time which has been estimated to be 15 minutes for a full bodysuit. It is stated the motor assessment part of the modified UPDRS test should take on average 15 minutes to complete. Therefore, including a setup time of 5 minutes for the single sensor used, a maximum of 20 minutes per session is required from the device.
The length dimension of the device must be less than 40mm, while the width dimension must be less than 30 mm.	The dimension delimitations were chosen having spoken to the two experts selected for this project. The 40 mm length delimitation was chosen considering the locations the device will be placed on the Shaolin Kung Fu bodysuit, specifically the wrist was a limiting location. While the 30mm width delimitation was chosen considering the device would be required to be placed across two fingers for the modified UPDRS test. More on device locations in Chapter 6.
Maximum sampling rate should be at least 200Hz.	The minimum maximum sampling rate was chosen considering orientation calculation would be required for the Shaolin Kung Fu dataset. 200 Hz has been stated to be the minimum effective sampling rate when using orientation calculations from accelerometer and gyroscope data [40].
A real time clock (RTC) is present on the device.	As multiple IMU devices will be used in conjugation for the Shaolin Kung Fu dataset an RTC is required to synchronise the devices during post processing.
The device has gyroscope, accelerometer and magnetometer.	Despite the magnetometer not being used in this research project, it was initially decided to select a device which includes a magnetometer to have the option for future research paths.
The device must be less than £90.	Given the price constraints given by the Shaolin Kung Fu school and this project's aim to tailor to these constraints, and the fact that at least 17 devices will be required for the bodysuit, this price criteria will be considered.

Table 1 Criteria used to select an IMU device. The green highlighted criteria were mandatory, while the orange highlighted criteria were only desirable

Table 2 below highlights the main specifications of interest for the 7 IMU devices compared throughout the project. The green highlighted features meet the criteria, the red highlighted features do not meet the criteria, while the orange highlighted features either just meet the criteria or are not specific to the criteria. It should be noted that all 7 of these devices include an RTC, therefore meeting that criterion. As can be seen from this table, only two of the 7 devices do not have any red features. Both the Mbientlab's MMR and the Portabiles' NilsPod meet the criteria, additionally the NilsPod also includes wireless charging which would increase its usability for the Parkinson's patients. However, the NilsPod does not include a magnetometer and only became available commercially towards the end of this project. Therefore, the Mbientlab's MMR device was chosen. For a more comprehensible comparison of these IMU devices please refer to [41].

Specifications	**MMR**	**QuantiMotion**	**NilsPod**	**Move 4**	**Physilog 5**	**EXL-s3**	**Shimmer 3**
Dimensions (mm)	36x27x10	36.5x32x13.5	28x23x11.5	62.3x23x11.5	47.5x26.5x10	54x33x14	51x34x14
Memory	8MB	32MB	250MB	4GB	8GB	1GB	8GB
Battery Capacity	100mAh	250mAh	120mAh	380mAh	140mAh	200mAh	450mAh
Maximum sampling rate	800Hz	100Hz	1024Hz	64Hz	512Hz	200Hz	1024Hz
Accelerometer	YES	YES	YES	YES	YES	YES	YES
Gyroscope	YES	YES	YES	YES	YES	YES	YES
Magnetometer	YES	YES	NO	NO	NO	YES	YES
Wireless charging	NO	NO	YES	NO	NO	NO	NO
Max recording time limitation	Memory	Memory	Memory	Battery	Battery	Battery	Battery
Max recording time (hours)	0.73	2.23	45.5	168	12.65	3	36.49

Table 2 Specifications of the 7 commercial devices compared during this project. The green highlighted features meet the criteria, the red highlighted features do not meet the criteria, while the orange highlighted features either just meet the criteria or are not specific to the criteria.

Table 3 summarises the sensor specifications of the MMR, while Table 4 summarises the IMU technical specifications.

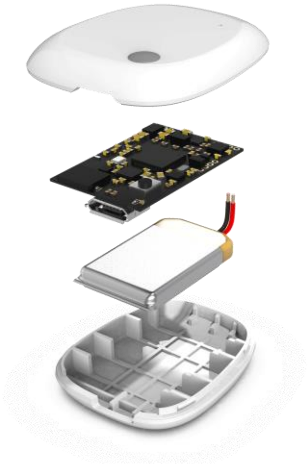


Figure 14 Cross section breakout of the MMR 9 axis IMU and environmental sensor [42]

Sensor	Range	Resolution	Sample rate
Gyroscope	± 125, ± 250, ± 500, ± 1000, ± 2000°/s	16bit	0.001Hz – 100Hz stream – 800Hz log
Accelerometer	± 2, ± 4, ± 8, ± 16 g	16bit	0.001Hz – 100Hz stream – 800Hz log
Magnetometer	±1300µT (x,y-axis), ±2500µT (z-axis)	$0.3\mu T$	0.001Hz – 25Hz
Temperature	-40°C – 85°C	0.01° C	0.001Hz – 50Hz
Ambient Light	0.01 lux – 64k lux	16bit	0.001Hz – 50Hz
Barometric Pressure	300 – 1100 hPa	0.01 hPa (< 10 cm)	0.001Hz – 50Hz
Additional Features			
Sensor Fusion	Quaternion, Rotation Matrix, Euler Angles (Robust Heading / Yaw, Pitch, Roll), Linear Acceleration, Earth Acceleration (Gravity)		
Analog and Digital Inputs	2 10bit (with 4 additional IOs under the board)		
Auxiliary Serial Interface	I2C and SPI		
Memory	8MB NOR Flash memory – store up to 1M sensor data entries		
Peripherals	Programmable Push-button, battery monitor, micro-USB battery charger, and programmable LED		

Table 3 MMR's sensor specifications. Feature highlight in green will be used in this project, features highlighted in orange are considered in this project, features highlighted in red are not used in this project. [42]

Weight: 0.2oz	Miniature form factor: 36mm × 27mm x 4mm in case
Synchronized timestamp: RTC allowing multiple devices to be synchronised	Low power consumption: Sleep mode supports 6 months idle time
LiPo battery: 70-100mAH micro-USB rechargeable. Charging time: 2h	Usage modes: Streaming mode: 8h-24h, **Recording mode: 2h-48h**
Splash Resistant Materials: IP40 plastic case	Data Transfer: Bluetooth Low Energy Smart®
Regulatory Compliance: CE, FCC, IC, RoHS	Secure Cloud platform

Table 4 MMR module's technical specification outlining features other than the sensors in the module. Feature highlight in green will be used in this project, features highlighted in orange are considered in this project. [42]

3.3 MEMS IMU Noise and Errors

In this section, the errors and noise present in MEMS technology will be characterised and discussed. Specifically looking at the error's sources for both MEMS gyroscopes and accelerometers individually. The effect the error has on the integrated MEMS signal will also be discussed due to its relevance in determining orientation. Errors in MEMS measurements can be broken down into two main categories: deterministic and stochastic. The deterministic errors can be subcategorised into the following: biases, scale-factors and misalignment error.

3.3.1 Deterministic Errors

3.3.1.1 Bias

Biases can be further broken down into three types: fixed, stability, and instability. The fixed biases, or offsets, in the MEMS measurements, $\vec{b}_a$ and $\vec{b}_g$, are the deterministic biases that can be calibrated out.

Bias in accelerometers

The bias of an accelerometer is the offset of the output signal from the true value, measured in m/s^2. The bias can be determined by measurement the long-term average of the accelerometer's output, whilst it is undergoing no acceleration. This is quite difficult to achieve as acceleration due to gravity will constantly be applied to the device, resulting in a bias including the component of gravity. To remove this component, it is necessary to know the exact orientation of the accelerometer in respect to the gravitational field.

When double integrating the constant bias error ϵ, an error in position which grows quadratically in time is produced:

$$s(t) = \frac{\epsilon t^2}{2} \tag{2.1}$$

The accelerometer fixed bias can be calculated using the equation below. The + and – superscripts denote the positively and negatively aligned subscript axis, respectively. Just the average value of the measurements on the aligned axes are used in this calculation.

$$\vec{b}_a = \frac{\begin{bmatrix} f_x^+ + f_x^- \\ f_y^+ + f_y^- \\ f_z^+ + f_z^- \end{bmatrix}}{2} \tag{2.2}$$

This test works with the assumption that the fixed bias is the same regardless of the orientation or forces on the sensor. By adding the two opposing measurements together, the opposite gravity measurements will cancel each other out and twice the fixed bias will be left behind.

Bias in gyroscopes

The bias of a gyroscope can be determined by taking an average of its output over an extended period, whilst the gyroscope undergoes zero rotation. Ideally the gyroscope should have no bias and therefore its reading over time should remain 0°/h, however in reality the bias of most MEMS gyroscopes is significantly high. This bias is often referred to as gyroscope drift.

Let ε be the constant bias, if integrating this constant in terms of time the integral becomes $\theta(t) = \epsilon.t$, which represents an angular error that's directly proportional to time with a gradient equal to the constant bias.

Intuitively, it seems a gyroscope's fixed bias could be calculated by fastening it in a fixed orientation and averaging the values measured on each sense axis, because the gyroscope is not rotating. This, however, does not work because gyroscopes are inertial sensors and are rotating inertially due to earth's rotation. High-resolution, low noise, gyroscopes can be used to measure the earth's rotation rate while sitting still. To offset the effect of earth rate, even if it cannot be measured on the MMR, the sense-axis is once again aligned in opposite orientations to find the bias.

$$\vec{b}_g = \frac{\begin{bmatrix} \omega_x^+ + \omega_x^- \\ \omega_y^+ + \omega_y^- \\ \omega_z^+ + \omega_z^- \end{bmatrix}}{2} \qquad (2.3)$$

3.3.1.2 *Scale factor*

Scale factor or sensitivity is the ratio between the measured output and the change in sense input. In general, scale factor is the slope of a straight line fitted by a least-squares method on a plot of the sensor output over the full input range. Scale-factor error is the ratio of the output error (deviation from the fitted straight-line slope) over the input and is typically expressed as a percentage or ppm (parts per million). In both accelerometer and gyroscopes, the scale-factor error will usually not be as large a contributor to total error as the bias error.

Scale factor in accelerometers

The effect of accelerometer scale-factor error is like that of bias error, except that the error term increases as the input acceleration increases. Therefore, the scale-factor error effect is worse in conditions where the acceleration range is wide.

The equation below can be used to calculate the scale factor error in an accelerometer.

$$\begin{bmatrix} s_{ax} & 0 & 0 \\ 0 & s_{ay} & 0 \\ 0 & 0 & s_{az} \end{bmatrix} = \frac{\begin{bmatrix} f_x^+ - f_x^- & 0 & 0 \\ 0 & f_y^+ - f_y^- & 0 \\ 0 & 0 & f_z^+ - f_z^- \end{bmatrix}}{2g} - \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad (2.4)$$

By subtracting the two measurement averages, the effects of the fixed bias are removed, and the sensor's output due to gravity are doubled. Because the input force, local gravity, is known, the input-to-output ratio can be calculated. The scale factor error is derived from this ratio.

Scale factor in gyroscopes

Gyro scale-factor error contributes significant errors to the angle tracking of the IMU during dynamic motion. The scale-factor error term is worse under large angular velocities; for example, a 0.1% scale-factor error on a 90°/s angular velocity can cause a 0.09°/s error. The effects of the resulting scale-factor error term can be treated like gyro bias error.

The equation below can be used to calculate the scale factor error in a gyroscope.

$$\begin{bmatrix} s_{gx} & 0 & 0 \\ 0 & s_{gy} & 0 \\ 0 & 0 & s_{gz} \end{bmatrix} = \frac{\begin{bmatrix} \omega_x^+ - \omega_x^- & 0 & 0 \\ 0 & \omega_y^+ - \omega_y^- & 0 \\ 0 & 0 & \omega_z^+ - \omega_z^- \end{bmatrix}}{2\omega_k} - \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad (2.5)$$

This set of measurements requires rotating the MEMS sensor at a known rate around each axis individually, this can either be done using a precise motor or ideally a rate-table. This known rate,ω_k, is then divided out of the measurements to calculate the input-to-output error.

3.3.1.3 Misalignment error

Misalignment, or cross-coupling errors are due to imperfect construction or alignment of the three sensing axes in the accelerometer or gyroscope. As seen in Figure 15, if the sense-axis is not perfectly aligned with the body axis it is supposed to measure, forces on the other two axes will create false readings on the sense-axis.

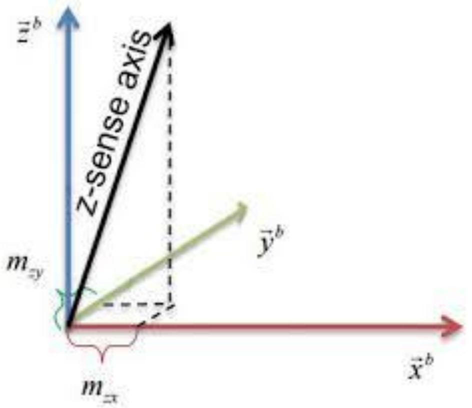


Figure 15 Misalignment error: the black z-sense axis is misaligned to the "expected" sensor axis [43]

Misalignment error in accelerometers

Accelerometer misalignment is calculated in a similar way to the scale factor, but instead of measuring along the gravity aligned axis, the two non-gravity axes are measured to find the cross-coupling. These calculations are shown below:

$$\begin{bmatrix} 0 & m_{axy} & m_{axz} \\ m_{yx} & 0 & m_{yz} \\ m_{zx} & m_{zy} & 0 \end{bmatrix} = \frac{\begin{bmatrix} 0 & f_y^+ - f_y^- & f_z^+ - f_z^- \\ f_x^+ - f_x^- & 0 & f_z^+ - f_z^- \\ f_x^+ - f_x^- & f_y^+ - f_y^- & 0 \end{bmatrix}}{2g} \tag{2.6}$$

The difference between the two measurements averages removes the fixed bias from the calculation and leaves twice the real measurement. Because the force incident on the cross-axis has a magnitude of local gravity, each element of the matrix is divided by $2g$. The arctangent of each element in the matrix is also taken to find the misalignment angle in radians.

Misalignment error in gyroscopes

The IMU is rotated at a known rate, ω_k around the one axis, and the measurements on the other axes are averaged to determine the misalignment angles. Like the accelerometer, the arctangent of each term is taken to determine the misalignment in radians, rather than as a ratio. The equation below adapts the same principles used for the accelerometer misalignment considering the known rotation.

$$\begin{bmatrix} 0 & m_{gxy} & m_{gxz} \\ m_{gyx} & 0 & m_{gyz} \\ m_{gzx} & m_{gzy} & 0 \end{bmatrix} = \frac{\begin{bmatrix} 0 & \omega_y^+ - \omega_y^- & \omega_z^+ - \omega_z^- \\ \omega_x^+ - \omega_x^- & 0 & \omega_z^+ - \omega_z^- \\ \omega_x^+ - \omega_x^- & \omega_y^+ - \omega_y^- & 0 \end{bmatrix}}{2\omega_k} \tag{2.7}$$

3.3.1.4 Calibration errors

Calibration errors may affect the scale factors, alignment, and linearity of the accelerometers. As the name would suggest, these errors occur due to the inaccuracies during the calibration process.

Calibration error in accelerometers

Assuming a relatively accurate calibration process, these errors will only be observed while the device is under acceleration. As the device is constantly being subjected to gravitational acceleration, the errors observed due to calibration inaccuracies will be present even when the device is stationary. The result of double integrating the error present due to calibration inaccuracies is a position drift proportional to the squared rate and duration of acceleration.

Calibration error in gyroscopes

Assuming a relatively accurate calibration process, these errors will only be observed while the device is under motion. In terms of the integrated signal, these errors result in an additional drift on each axis with the magnitude proportional to the rate and duration of the motion.

3.3.2 Stochastic Errors

3.3.2.1 Bias Instability

Due to the flicker noise produced by the electronics within the MEMS gyroscope and the random flicker which occurs in the mechanical component, the bias of the gyroscope drifts over time. This flicker noise is usually dominant at low frequencies and overshadowed by white noise at high frequencies. This noise results in bias instabilities and is often modelled as a random walk.

To measure a sensor's bias instability, the bias of the device is measured in fixed condition over a given period, typically 100s. This measurement is commonly given in terms of a standard deviation over a period, with units of °/h or for less accurate devices °/s. Over this period a random walk is created within the gyroscope's bias, with its standard deviation growing proportionally to the square root of time.

Assuming the bias instability follows the random walk model (first order), the integral would be a second-order random walk in the orientation domain. In fact, the bias instability only follows this model for short time period, which is fortunate, otherwise the stability error would grow without limit as time moves forward.

MEMS accelerometers are also affected by flicker noise causing instability in the bias of the output signal. Using the random walk model, the bias instability results in a second order random walk in the velocity, with uncertainty growing proportionally to $t^{\frac{3}{2}}$, and a third order random walk in position, with uncertainty growing proportionally to $t^{\frac{5}{2}}$.

3.3.2.2 White Noise

White noise in gyroscopes

Due to the mechanical structure of the MEMS gyroscope, some errors will occur because of thermo-mechanical noise. This noise is at a much higher frequency than the sample rate of the sensor, resulting in a white noise present at the sensor's readings. The white noise can be represented by a sequence of zero mean uncorrelated variables. These variables are random and evenly distributed through all frequencies and have a finite variance of σ^2.

By using the rectangular rule to integrate the white noise an understanding of the effect this noise has on the orientation calculation can be determined. Let N represent the white noise sequence with N_i being the ith random variable. As the sequence is considered as white noise the following characteristics can be assumed:

$$E(N_i) = E(N) = 0 \tag{2.8}$$

$$Var(N_i) = Var(N) = \sigma^2 \tag{2.9}$$

$$Cov(N_i, N_j) = 0 \; for \; all \; i \neq j \tag{2.10}$$

The result of integrating the white noise signal $\epsilon(t)$ using the rectangular rule is:

$$\int_0^t \epsilon(\tau)d\tau = \delta t \sum_{i=1}^{n} N_i \tag{2.11}$$

Where n is the number of samples from the gyroscope and δt is the time between successive samples.

$$E(aX + bY) = aE(X) + bE(Y) \tag{2.12}$$

$$Var(aX + bY) = a^2 Var(X) + b^2 Var(Y) + 2abCov(X, Y) \tag{2.13}$$

As the above formulae are true when a and b are constant, and X and Y are random variables, the following can be determined:

$$E\left(\int_0^t \epsilon(\tau)d\tau\right) = \delta t. n. E(N) = 0 \tag{2.14}$$

Therefore, the noise introduces a zero-mean random walk error into the integrated signal, with the standard deviation:

$$\sigma_\theta(t) = \sigma. \sqrt{\delta t. t} \tag{2.15}$$

White noise in accelerometers

As with the MEMS gyroscope, thermo-mechanical noise produces a white noise sequence in the output signal. Previously it was shown that by integrating the white noise present at the output a random walk is produced with a standard deviation growing proportionally to $\sqrt{t}$. By integrating this random walk again, a velocity random walk is produced.

The result of double integrating the white noise signal $\epsilon(t)$ using the rectangular rule is:

$$\int_0^t \int_0^t \epsilon(t)d\tau d\tau = \delta t \sum_{i=1}^{n} \delta t \sum_{j=1}^{i} N_j \tag{2.16}$$

$$= \delta t^2 \sum_{i=1}^{n} (n - i + 1)N_i \tag{2.17}$$

Where n is the number of samples from the accelerometer and δt is the time between successive samples.

The expected error in position is:

$$E\left(\int_0^t \int_0^t \epsilon(\tau) d\tau d\tau\right) = \delta t^2 \sum_{i=1}^{n} (n-i+1) E(N_i) = 0 \tag{2.18}$$

The variance in position is:

$$Var\left(\int_0^t \int_0^t \epsilon(\tau) d\tau d\tau\right) = \delta t^4 \sum_{i=1}^{n} (n-i+1)^2 Var(N_i) \approx \frac{1}{3}.\delta t.t^3.\sigma^2 \tag{2.19}$$

Where the approximation is made assuming δt is short, which is the case with modern MEMS sampling frequency.

Therefore, the white noise present at the output signal of the accelerometer, results in a second order random walk in position, with a zero mean and a standard deviation:

$$\sigma_s(t) \approx \sigma.t^{\frac{3}{2}}.\sqrt{\frac{\delta t}{3}} \tag{2.20}$$

3.3.3 Handling stochastic errors

Compensating or removing stochastic errors is not as straight forward as deterministic errors due to their random nature. Depending on use cases a simple low pass filter can remove the white noise component while a high pass filter can remove the flicker noise, however when the frequency of the stochastic errors lie near the frequency of the recorded movement more complicated filters are required. These filters are described in Chapter 5.

3.3.4 Summary of errors in MEMS sensors

Error Type	Description	Result of single integration	Result of double integration
Bias	Constant bias ϵ	$\theta(t) = \epsilon t$	$s(t) = \frac{\epsilon t^2}{2}$
Scale factor	Directly proportional to time	$\tau(t) = \frac{\epsilon t^2}{2}$	$\varphi(t) = \frac{\epsilon t^3}{6}$
Bias Instability	Fluctuations which arise due to flicker noise	Second order random walk in angle	Second order random walk in velocity and a third order random walk in position
Temperature effects	Residual bias due to change in temperature	Error in orientation which grows linearly with time.	Error in position which grows quadratically with time.
Calibration inaccuracies	Deterministic errors in scale factors, alignments, and sensor linearities	Orientation drift proportional to the rate and duration of motion.	Position drift proportional to the squared rate and duration of acceleration.

White Noise	Random noise having equal intensity at different frequencies	$\sigma_\theta(t) = \sigma.\sqrt{\delta t.t}$	$\sigma_s(t) \approx \sigma.t^{\frac{3}{2}}.\sqrt{\frac{\delta t}{3}}$

Table 5 Summary of errors in MEMS sensors

3.4 CHAPTER SUMMARY

In this chapter the errors associated with using MEMS sensors have been outlined, including descriptions of why the errors occur, its effect on the integral and double integral and how to calculate them. It is clear from Table 5 that the errors can quickly grow out of control, in some cases exponentially, when integrating the MEMS measurements. The deterministic errors have been calculated for all sensors used before doing any measurements allowing each sensor to be uniquely compensated reducing the overall error in the system. Methods to address stochastic errors will be discussed in chapter 3.

4 IMU VALIDATION PROCEDURE LITERATURE REVIEW

Due to the filters required when using MEMS sensors, a test procedure must be designed to assess the inertial sensors system in both static and dynamic conditions. Of the two conditions designing a test procedure for static measurements is a lot simpler, as it just requires the sensor to remain stationary in a known orientation. As most MEMS inertial sensors are capable of measuring accuracies <1^{o} it is required that the orientation is known to an even greater precision.

Specialised devices such as a rotary xyz-stage can be used to accomplish such conditions. The minimum resolution expected in industrial rotary stages are within the arcminute range, however with some of the more specialised devices the resolution can be fractions of an arcsecond [44]. Another piece of equipment that can be used is a vibration turntable; this is used to ensure an exact horizontal surface that will not be affected by vibrations.

Yuan et al could isolate external vibrations from a low-cost MEMS IMU and the MTi-300 using a turntable, allowing a more accurate evaluation of both sensors' static performance [45]. Allowing accelerations as low as 0.01 m/s^{2} to be recorded, it was concluded that these accelerations were due to quaternion noise, and that any external vibrations were too arbitrarily low to be considered even when compared to 0.01 m/s^{2}.

To evaluate the dynamic accuracy of inertial sensors a more elaborate setup is required. These setups can be split into three categories:

- Benchmark comparisons (optical motion capture, video motion capture, electronic goniometer)
- Known movements using precise actuators (robotic arm, rotary stage)
- Simulations (MATLAB)

4.1 BENCHMARK COMPARISONS

This category refers to motion capturing techniques that have previously been established, and that have proven to be sufficiently accurate for the movements recorded. When determining which benchmark to use to measuring a system against, it is important to determine the required accuracies of the system, as well as other characteristics, such as velocity of movements. In recent publications, there have been x benchmark used:

1. Image processing system, 2D or 3D
2. optical motion capture, 3D
3. IMU motion capture [46], 3D
4. electronic goniometer, 2D

4.1.1 Validated using IPS benchmark

In terms of hardware this technique is very simple, all it requires is a standard video camera. The data recorded by the video camera then undergoes some image processing techniques to analyse the motion recorded. The main techniques used in video motion capture all involve robust tracking of features. This technique is used to determine the speed and position of a feature over several captured frames. For a detailed description on using image processing in movement analysis, please refer to *Uke et al.* "Motion tracking system in video based on extensive feature set" [47].

The accuracy of video motion capturing systems depends on many aspects of the systems. The resolution of the camera directly affects the accuracy by setting the limit of minimum movement possible to record. Another aspect which influences the accuracy is the number of cameras used. *Mundermann et al.* have determined the optimal number of cameras and the configuration used for biomechanical analysis, stating that the use of less than 8 cameras for accurate capturing of human motion is questionable and the accuracy fluctuates enormously for different poses and positions [48]. *Rosenhahn et al.* using 8 cameras are in fact able to achieve an accuracy of <2.5° [49]. Another important aspect when considering the accuracy of the system, is the motion being recorded. *Becker* determines that when using a single camera motion with a range of less than 5° are too small to be accurately measured resulting in a correlation of less than 0.5 when captured with the Simi Motion system [50].

Dong et al. determines whether an inertial based system can track and classify daily physical activities in real time to a good accuracy [51]. During their test procedure, inertial sensors are strapped to a subject's thigh and shank and asked to perform several intensive activities in a gym, including cycling, running on treadmill, and push ups. The sensor results were then compared to those of a video motion capture. Whilst walking on a treadmill the average difference between the inertial sensor results and the video reference results were approximately 4.1°.

4.1.2 Validated using OMC benchmark

By far the most used motion capturing technology currently available, optical motion capture (OMC) systems can be considered as an upgraded version of video motion capture. Most OMC systems use multiple camera; however, it is possible to use a single camera (monocular) in this approach as well. Unlike video motion capture systems, OMC systems use markers to measure the trajectories of specific point of the object being monitored. Room preparations for OMC systems are like the room preparation require for multiple camera video motion capture systems, except for reference markers which are often placed around the room. Uniform lighting is also required to reduce the noise resulting from shadows and highlights in the scene. This means that several light sources and reflectors sheet need to be arranged around the capture room.

Other than the room the object must also be prepared. If the object of interest is a moving person, then a special body suit must be worn. The suit is covered in markers which have a specific shape and reflective properties. Markers are attached near joints and feature point of the human body, i.e., the chest. Generally, the suit colour and the marker colour are very distinguishable allowing the detection of the markers to be more robust. The body suit is usually a tight-fitting leotard, including gloves, socks, long sleeves, and hood (sometimes including a mask).

Specialised software is required to trigger the cameras as well as acquiring synchronous video data. The computer system used to run the software must have enough processing power to provide the bandwidth for recording as well as the storage for saving multiple video streams.

Guerra-Filho has a thorough and clear description of the theory and implementation of OMC systems [52].

There are several companies which each have their own variant of OMC system, however the leading systems are designed and manufactured by OptiTrack. For monitoring a single person's motions systems cost between $7,000 and $60,000 [53].

Zhang et al. uses Xsens' MTx sensor to test their fusion filter design [54]. To evaluate the performance of the filter, they opted to compare the results of the MTx sensors with the BTS SMART-D optical motion capture system [55]. The setup of the sensors and the optical markers can be seen in Figure 16. Seven high-resolution BTS cameras were setup around the room to get a clear view of the subject at all necessary angles. The BTS SMART application then could reconstruct the 7 2D data sequences into a 3D ground truth. Using this ground truth, Zhang & Wu could compare correlation of the MTx sensors before and after their proposed method.

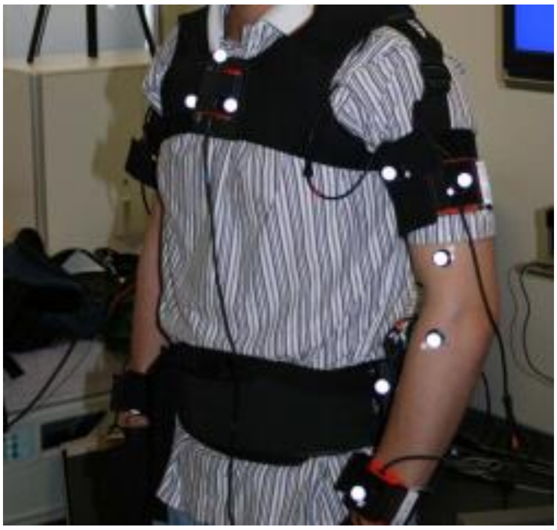


Figure 16 Configuration of MTx inertial sensors and optical markers for human upper limb motion estimation assessment procedure [54]

Baldi et al. estimates the limbs' position during a pose using inertial sensors [56]. Tracking errors were validated with respect to the OptiTrack system. Two different experiments, evaluating the accuracy of the proposed method, have been carried out. The first intends to show the lack of drift, whereas the second demonstrates the accuracy in tracking both slow and rapid body movements. It is important to highlight that the performance of the considered approaches largely depends on the integration capability of the sensors, thus the higher the sampling rate, the more accurate the estimation. Two different sampling rates for each experiment have been considered: i) high sampling rate (1 kHz), and ii) low sampling rate (100 Hz) [56].

Ruffaldi et al. have used 5 IMU devices combined with an Unscented Kalman Filter to estimate the motion of a rower using a rowing machine [57]. The participant was provided with 5 inertial units MPU9150 (Invensense, Borregas Ave Sunnyvale, CA, USA) placed on the back and left and right upper arm and forearm, communicating via Bluetooth to an acquisition pc. The measurements were then validated using a marker based optical capture system. Firstly, the participant was instructed to perform a three-step calibration procedure needed to compute the inertial sensors orientation and calibrate the magnetometers. Then, they were asked to perform multiple sequence of rowing strokes, while being tracked both by the optical and inertial tracking systems.

While most publications are satisfied using OMC benchmarks, *Sy et al.* concludes the use of OMC benchmarks to validate their reduced IMU Extended Kalman filter algorithm introduces a new error associated to the independent motion between the optical trackers and the IMU devices [58]. Therefore, reducing the validity of the OMC benchmark as a ground truth.

4.1.3 Validating using Xsens' IMU benchmark

The use of inertia-based motion capturing systems as a benchmark is a relatively new option. Currently the Xsens systems are the only inertia-based motion capturing systems with the specifications required for research purposes. The Xsens system can either be bought as a full bodysuit or as a set of IMU devices. The bodysuit is designed to allow the major movements of the body to be measured by having inertia sensors at each joint and feature point seen in Figure 17. By using the calculated orientations of individual body segments and the knowledge about the segment lengths, rotations between can be estimated and a position of the segments can be derived under strict assumptions of a linked kinematic chain [59].

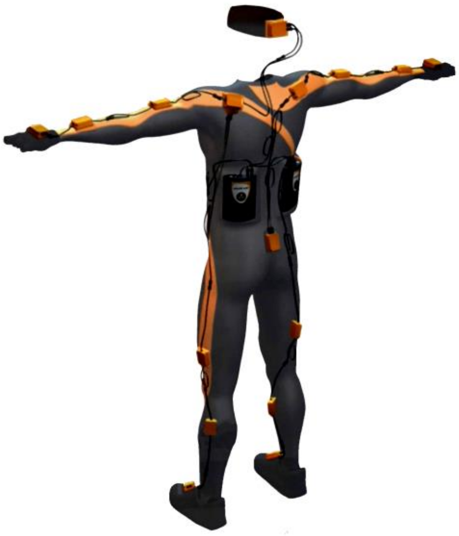


Figure 17 Suit designed for quick use and convenient placement of the Xsens MVN sensors [59]

To use either the sensor straps or the body suit, Xsens has developed their own output studio which allows the user to create a simple segmented animation model of the person or object being monitored. This studio is specifically designed to take the data transmitter by the sensors and have it display in real time in a 3D environment.

External sensors are not required in the Xsens system; however, this means that the motion has no external reference point. If an external reference point is required Xsens has designed a system called MVN MotionGrid which essentially has the equivalent of a local GPS system added to the standard MVN system.

Currently Xsens offers a choice of two series of sensors, MTi 10-series and MTi 100-series. The main difference between these series is the gyro bias stability, 18^{o}/h and 10^{o}/h respectively. The stability is due to the drift present in all MEMS gyroscope sensors. In the MTi 100-series, the MTi-G-710 GNSS is the only sensor Xsens currently offers with position and velocity tracking [59].

Yuan et al. uses a combination of a test turntable and Xsens' MTi-300 to verify the feasibility of using the Allan variance to evaluate the performance of a low-cost wearable MEMS IMU sensor [45]. The low cost IMU sensor consists of a separate accelerometer, gyroscope and magnetometer. Both the low-cost sensor and the MTi-300 were then sampled at 100Hz and the results over an hour were compared. By comparing the Allan variance of the gyroscope over the hour, they could determine the main source of error: <1 second as quaternion noise, the 10s of seconds as angular random walk, and the 100< seconds as bias instability. It was also clear from the results that the MTi-300 had significantly higher stability than the low-cost sensor. By comparing the Allan variance of the accelerometer over the hour, they could determine the main source of error: <1 second as quaternion noise, the 10s as bias instability, and the 100< seconds as velocity random walk. They were also able to determine that the quaternion noise in the low-cost device lasted longer than that of the MTi-300.

4.1.4 Validating using electronic measuring apparatus

The devices described in this section are all designed specifically to measure orientation electronically to a high level of precision.

Vitali et al. uses a coordinate measurement machine (CMM), see Figure 18, as a ground truth for the orientation estimates [60]. The CMM uses embedded optical encoders to measurement three dimensions of rotation to a resolution of 0.004 degrees. Their experimental setup has the IMU mounted to the CMM and rotated along one rotation degree of freedom for 5 seconds while the other joints are locked. This allows a single degree of rotation to be evaluated at a time. Vitali et al. notes that that the CMM results in some time varying magnetic interference requiring some posterior compensation.

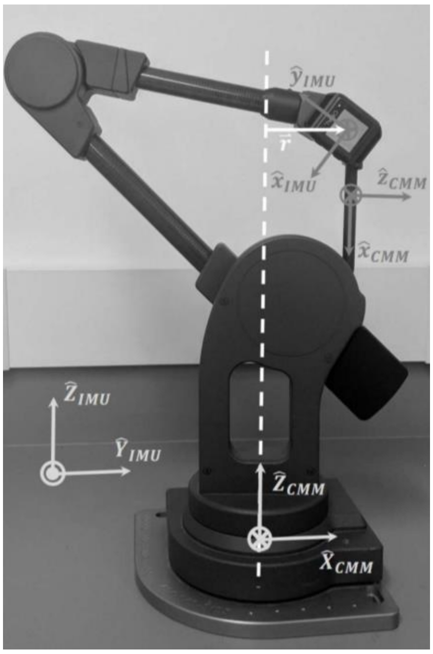


Figure 18 Set-up for validation of IMU-estimated orientation with the IMU attached to the end effector of a coordinate measurement machine (CMM). Body-fixed frames (dark grey) and inertial frames (light grey) for the IMU and end effector are also illustrated. The white dashed line denotes the rotational axis of the CMM's base and the solid white arrow denotes the position vector of the IMU relative to that axis. [60]

Vitali et al. uses two IMU devices attached on the shank and thigh of a leg to estimate the motion of a knee's rotation, once again a CMM is used to validate the experiment [61]. Data from the two IMUs are first time-synchronized to the encoder data from the CMM. The assembly is rotated by hand about the CMM's base (white axis in Figure 19 (a)) with the three knee axes (FE, IE, AA) locked. The angle measured by the optical encoder about the base (dashed white) axis is differentiated with respect to time yielding an angular velocity signal to compare with those measured by the thigh (green) and shank (blue) IMUs. The data from the two IMUs are already time-synchronized, and their synchronization with the data from the CMM follows from measuring (and subsequently subtracting) the time delay between their respective angular rates.

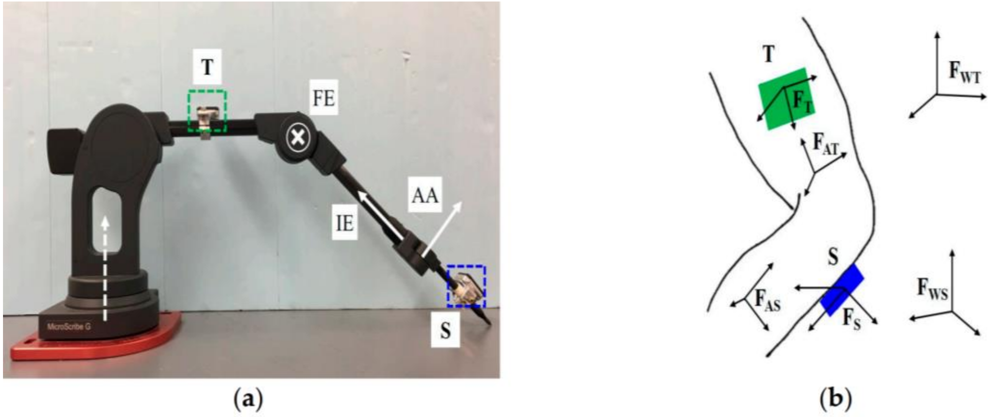


Figure 19 Knee analogue formed by a coordinate measuring machine (CMM). (a) Three anatomical axes for flexion/extension (FE), internal/external rotation (IE), and abduction/adduction (AA) are labelled at the corresponding rotational joints of the CMM. Two labelled IMUs are mounted to the CMM with T (green) analogous to a thigh-mounted IMU and S (blue) analogous to a shank-mounted IMU. (b) Definitions of three frames of reference for a human knee associated with a shank mounted IMU (blue) including the shank IMU frame, FS, the shank anatomical frame, FAS, and the shank IMU's world frame, FWS. Analogous frames of reference are illustrated for the thigh mounted IMU (green). [61]

Betz et al. uses an electronic goniometer to validate their use of IMUs to estimate the inversion angle of an ankle rolling [62]. A test bench is designed to roll the subject's ankle into a specific angle while two IMUs are mounted above and below the ankle, additional a goniometer is mounted across the ankle, see Figure 20. Relative orientations of the IMUs are calculated using the real part of the quaternion representation, more on this in section 2.2, and compared to the goniometer's measurements. The IMUs and goniometer were synchronised using coherence events. It was concluded the goniometer was not a good benchmark to compare against due to its own accuracy only being ±2°; and that its sampling frequency was only 100Hz, which is not sufficient to capture all the aspects of the ankle inversion motion.

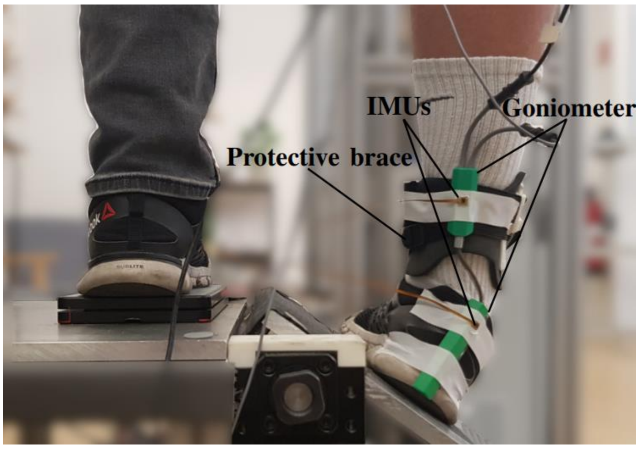


Figure 20 Inversion Kinematic test bench with a rapidly tilting platform element and with inertial sensors and goniometers attached to the foot and shank to capture the inversion motion of the ankle

4.2 Known movements using precise actuators

Any device which requires precise movements, can fall under this category. This can be as simple as a single stepper or as complex as industrial medical robotic arm. To fully assess the accuracy of the system being developed in this project, the number of degrees of freedom should match. As discussed previously the expected number of degrees of freedom for system being developed is 3 DOF, therefore ideally the assessment device should also have 3 DOF. The most important aspect when choosing an assessment device is the repeatability of the movement, and that this repeatability has been thoroughly tested. The next aspect to consider when choosing an assessment device is the velocity and acceleration requirements.

To evaluate the project's system, the sensors being developed would be placed on the moving parts of the assessment device. The exact location of where the sensors need to be placed depends on the motion of the assessment device. It is important that when the sensors are placed on the assessment device, that they are secured in such a manner that sensors are not able to move independently to the assessment device.

Due to the vast number of devices which fall under this category it is difficult to assess the available options. Instead, it would be more reasonable to characterise the devices once the testing procedure begins and an assessment device has been chosen. However, for ease of comparison a simple stepper can be used as an assessment device. A standard stepper motor without any specialised controller or gear box is capable of rotational accuracies up to 0.03°.

Mahmoud El-Gohary et al. compared the joint angles calculated by their inertial tracker sensors with those obtained by the Epson C3 robotic arm [63]. The Epson C3 [64] is an industrial arm designed for medical purposes, having high speed control as well as very high joint angle precision. The 'shoulder joint' is, however, limited to 2 DOF rather than the ideal 3 DOF required to accurately model a human shoulder. Other than the shoulder they could emulate the rest of the arm's movement, surpassing, the maximum range of motion of a human arm, as well as the operating speed. This allowed them to determine the root mean square error (RMSE) between the angles calculated by the inertial sensors and the robotic arm at three set speeds: slow, medium, and fast. By using the robotic arm for both their Unscented Kalman Filter (UKF) and Extended Kalman Filter (EKF) they could determine the error difference between the two filters.

Wang et al. has validated their improved Kalman filter through a simulated and practical turntable experiment [65]. For both experiments a misalignment is set, and three types of Kalman Filter are applied to align the sensors. The misalignment errors were recorded with their mean and standard deviation compared for each Kalman filter. As expected, the errors associated with yaw were greatest for all three Kalman filters.

4.3 Validating using a simulation

Most of the literature reviewed which uses a simulation to validate the experiments done, do this alongside a practical experiment [65] [56] [58]. In these simulated experiments a motion is decided upon as the ground truth. Following this a noise source is added to the ground truth to create the simulated measurements. These measurements are then processed using the proposed filtering technique and the output is compared to the ground truth first established.

Liu uses MATLAB simulations to validate the use of Complimentary filtering on IMU data [66]. The simulation rotates along pitch, roll and yaw separately using varying rotation speed, from 1 radian to 400 radians per second, for 100 seconds. The noise of tri-axis angular rates is set as first order Markov noise derived from Gaussian noise with zero mean value and 100 °/h standard deviation.

4.4 CHAPTER SUMMARY AND CONCLUSION

This chapter has reviewed the validation techniques used when evaluating the performance of IMU-based systems and filtering algorithms. Of the techniques discussed the OMC benchmark, the electronic measuring apparatus and precise actuators have the highest accuracy to compare against. The electronic measuring apparatus and the precise actuators have many devices that fall under these categories and therefore have a large range of resolutions associated to them, for example ±2^{o} for the electronic goniometer and ±0.004^{o} for the CMM. It was established by the experts in each field of the movement sets that the use of an OMC systems would not be possible due to the restrictions these systems impose on the movement. Therefore, a piece of equipment which falls under the electronic measuring apparatus, or the precise actuators will be chosen to validate the filtering techniques used in the following chapter. Ideally the CMM would be used as this piece of equipment has the highest resolution reviewed as well as being able to 3D coordinates, however if a CMM cannot be acquired a comparable device should be used.

5 FUSION FILTERING

This chapter will give the reader an understanding of the available fusion filters suitable to IMU movement analysis. The chapter will begin with outlining the theory behind the different algorithms, indicating advantages and disadvantages of each filter. To begin with the linear Kalman filter will be discussed as a prerequisite to its nonlinear counterparts: the Extended Kalman Filter and the Unscented Kalman Filter. The chapter will also outline the two non-parameter filters: the Complementary filter and the Particle filter. Finally, the filters will be compared using the precise actuator technique described in Chapter 4.

5.1 COMBINING ORTHOGONAL ESTIMATES

Many of the fusion filtering techniques described in this section and in general uses the combination of independent estimates as the centre of their algorithms. Therefore, before describing any specific filters a clear explanation of how two estimates of a variable can be used to form a weighted mean value.

Let start with considering two orthogonal estimates of a value of x, x_1 and x_2. These estimates will have a variance (or covariance in respects to x) of, σ_1^2 and σ_2^2. To combine these two estimations to determine mean estimation, $\hat{x}$, each independent estimation must be weighted with the weights corresponding to the variances.

$$\hat{x} = w_1 x_1 + w_2 x_2 \tag{4.1}$$

Where w_1 and $w_2$are the weighting factoring of the respective estimates, and $w_1 + w_2 = 1$.

The variance of x is defined as $E[\{x - E(x)\}^2]$. If σ^2 denotes the variance of $\hat{x}$, the following is true:

$$\sigma^2 = E\left\{\left(w_1 x_1 + w_2 x_2 - w_1 E(x_1) - w_2 E(x_2)\right)^2\right\} \tag{4.2}$$

$$= E\left\{w_1^2\left(x_1 - E(x_1)\right)^2 + w_2^2\left(x_2 - E(x_2)\right)^2 - 2w_1 w_2\left(x_1 - E(x_1)\right)\left(x_2 - E(x_2)\right)\right\} \tag{4.3}$$

As we have defined x_1 and x_2 as orthogonal, $E\{\left(x_1 - E(x_1)\right)\left(x_2 - E(x_2)\right)\} = 0$. Therefore:

$$\sigma^2 = w_1^2 E\left\{\left(x_1 - E(x_1)\right)^2\right\} + w_2^2 E\left\{\left(x_2 - E(x_2)\right)^2\right\} \tag{4.4}$$

$$= w_1^2 \sigma_1^2 + w_2^2 \sigma_2^2 \tag{4.5}$$

Rewriting $w_2 = w$ and $w_1 = 1 - w$, the variance will be as follows:

$$\sigma^2 = (1-w)^2\sigma_1^2 + w^2\sigma_2^2 \tag{4.6}$$

By differentiating the equation above and equating the answer to 0, a value of w which minimises σ^2 can be determined:

$$\frac{d\sigma^2}{dw} = -2(1-w)\sigma_1^2 + 2w\sigma_2^2 = 0 \tag{4.7}$$

By rearranging the equation above the optimally w value can be found.

$$w = \frac{\sigma_1^2}{\sigma_1^2 + \sigma_2^2} \tag{4.8}$$

By substituting the above equation into 4.1 and 4.6 the following expressions are true:

$$\hat{x} = \frac{\sigma_2^2 x_1 + \sigma_1^2 x_2}{\sigma_1^2 + \sigma_2^2} \tag{4.9}$$

$$\sigma^2 = \frac{\sigma_1^2\sigma_2^2}{\sigma_1^2 + \sigma_2^2} \tag{4.10}$$

The mathematical process described from equation 4.4 to 4.10, show how two orthogonal estimations can been combined to form a weighted mean value. The process also shows how to minimise the variance by determining the optimal weighting factor. By minimising the variance, the probability is maximised. When using fusion filters such as the Kalman Filter in IMU systems, one of the estimations will be determined with equations of motion, and the other estimation will be measurements from the sensors in the system.

The above case is only valid for systems that are single dimensions, for example, a system only interested in the velocity of a ball. For more complex systems, such as, a system interested in the velocity, spin and position of a ball, the mathematical process must be adjusted to handle n-dimensional estimations. The following initial adjustments must be made to the setup before continuing onto the mathematical process:

Single dimension	Multiple Dimensions	Matrix dimensions
$x, \hat{x}, x_1, x_2$	$\boldsymbol{x}, \boldsymbol{\hat{x}}, \boldsymbol{x_1}, \boldsymbol{x_2}$	n-elements
$\sigma^2, \sigma_1^2, \sigma_2^2$	$\boldsymbol{P}, \boldsymbol{P_1}, \boldsymbol{P_2}$	n by n elements
w	$\boldsymbol{W}$	n by n elements

Table 6 Adjustments required for the multidimensional case

Using the same reasoning as with the single dimension case and the adjustments seen in Table 6, the following can be said about the combined weighted mean:

$$\widehat{\boldsymbol{x}} = (\boldsymbol{I} - \boldsymbol{W})\boldsymbol{x}_1 + \boldsymbol{W}\boldsymbol{x}_2 \tag{4.11}$$

$$= \boldsymbol{x}_1 - \boldsymbol{W}(\boldsymbol{x}_1 - \boldsymbol{x}_2) \tag{4.12}$$

Where $\boldsymbol{I}$ is the unit matrix with n by n elements. As can be seen in the above equations only minor adjustments are required when moving from a single dimension system to a multi-dimension system, however this is only the case when both estimates have the same dimensions. In many systems, this is not always the case, in fact, in many systems one of the estimates is only a function of the individual elements of the system state, i.e.:

$$\boldsymbol{y}_2 = \boldsymbol{H}\boldsymbol{x}_2 \tag{4.13}$$

Where $\boldsymbol{y}_2$ is m measurements related to m number of elements of $\boldsymbol{x}_2$, and $\boldsymbol{H}$ is matrix of m by n elements denoting this relationship. In this case the second estimate will have a variance of $\boldsymbol{R}$ and a weighting matric of $\boldsymbol{KH}$, where $\boldsymbol{K}$ is arbitrary weighting matrix.

By applying 4.13 to 4.12, following is true:

$$\widehat{\boldsymbol{x}} = \boldsymbol{x}_1 - \boldsymbol{KH}(\boldsymbol{x}_1 - \boldsymbol{x}_2) \tag{4.14}$$

$$= \boldsymbol{x}_1 - \boldsymbol{K}(\boldsymbol{H}\boldsymbol{x}_1 - \boldsymbol{y}_2) \tag{4.15}$$

$$= (\boldsymbol{I} - \boldsymbol{KH})\boldsymbol{x}_1 + \boldsymbol{K}\boldsymbol{y}_2 \tag{4.16}$$

The variance of $\widehat{\boldsymbol{x}}$, $\boldsymbol{P}$ is defined as:

$$\boldsymbol{P} = E\{[\widehat{\boldsymbol{x}} - E(\widehat{\boldsymbol{x}})]\{\widehat{\boldsymbol{x}} - E(\widehat{\boldsymbol{x}})]^T\} \tag{4.17}$$

By using similar expression for the variance of each estimate, applying 4.16 to 4.17, and considering the estimates being orthogonal the following can be found:

$$\boldsymbol{P} = (\boldsymbol{I} - \boldsymbol{KH})\boldsymbol{P}_1(\boldsymbol{I} - \boldsymbol{KH})^{\boldsymbol{T}} + \boldsymbol{KRK}^{\boldsymbol{T}} \tag{4.18}$$

To determine the value of $\boldsymbol{K}$ which minimises the variance of the estimated state, the following expression, which has been proven Barham et al. [1], can be used:

$$\boldsymbol{K} = \boldsymbol{P}_1\boldsymbol{H}^{\boldsymbol{T}}[\boldsymbol{H}\boldsymbol{P}_1\boldsymbol{H}^{\boldsymbol{T}} + \boldsymbol{R}]^{-\mathbf{1}} \tag{4.19}$$

Substituting 4.18 back into 4.19 the variance allowing the best estimate of the state can be determined:

$$\boldsymbol{P} = \boldsymbol{P_1} - \boldsymbol{KHP_1} \tag{4.20}$$

Several of the expressions used during this mathematical process of determining the optimal weighting for a multidimensional state system are implemented in the Kalman filter and many other filters based on sensor/state fusion.

5.2 KALMAN FILTER

As mentioned previously the Kalman filter, uses the combination of estimations to equate a mean estimation which should in theory have a higher probability of being the actual state of the system. The filtered is designed to work on dynamic linear system with at least one measurement relevant to the state of the system. Before going into the details of how the filter works it is necessary to define the behaviour a linear system. Below is the set of first order differential equations which can represent the dynamic behaviour of a linear system:

$$\frac{d\boldsymbol{x}}{dt} = \boldsymbol{Fx} + \boldsymbol{Gu} + \boldsymbol{Dw} \tag{4.21}$$

Where:

$\boldsymbol{x(t)}$, is a vector of n elements, representing the states of the system,

$\boldsymbol{u(t)}$, is a vector of p elements, representing the deterministic inputs of the system,

$\boldsymbol{w(t)}$, is the systems noise, assumed to have a zero mean and Gaussian distribution, and a power spectral density of $\boldsymbol{Q,}$

$\boldsymbol{F}$, is a n by n matrix known as the system matrix,

$\boldsymbol{G}$, is a n by p matrix known as the system input matrix,

$\boldsymbol{D}$, is the covariance of the noise on the state of the system.

Assuming the system has m measurements, which are a linear combination of the states, and has been corrupt by noise. The following is true:

$$\boldsymbol{y} = \boldsymbol{Hx} + \boldsymbol{n} \tag{4.21}$$

Where:

$\boldsymbol{y(t)}$, is vector of m elements, representing the measurements of the system,

$\boldsymbol{H}$, is a m by n matrix known as the measurement matrix,

$\boldsymbol{n(t)}$, is the measurement noise, assumed to have a zero mean and Gaussian distribution, and a power spectral density of $\boldsymbol{R}$.

With the behaviour of the linear system fully described in the expressions above, the algorithm used in the Kalman filter can be explained properly.

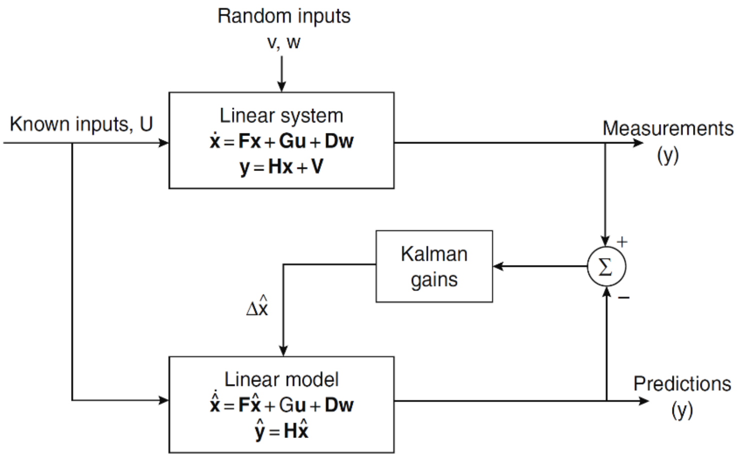


Figure 21 Block diagram of the Kalman filter

The deterministic or measurable inputs are processed by both the system and the model of the system, as shown in the block diagram representation in Figure 21. The measurements of the true system are compared with predictions of those measurements, derived from the latest best estimates of the states provided by the system model. The difference between the true and predicted measurements are fed back through a weighting matrix, the Kalman gain matrix, to correct the estimated states of the model.

The Kalman gains are selected to provide best estimates of the states in a least squares sense. It can be shown that this is equivalent to the best estimate in the maximum likelihood sense in the linear, Gaussian noise system described above. It should be noted that, it is because there is a feedback of a noisy signal in the Kalman filter that the system must be linear and the noise Gaussian distributed. It is only because the distribution of the sum of two normally distributed signals is itself normally distributed, and because a normally distributed signal remains so after passing through a linear system, that a least-squares or maximum likelihood estimation procedure can be applied repeatedly.

Whilst it is usual for the system to be described mathematically in the continuous differential equation form given above, the measurements are in practice provided at discrete intervals of time. To cope with this, and to provide a computationally efficient filtering algorithm, it is customary to express the continuous equations in the form of difference equations as shown below:

$$\boldsymbol{x_k = A_k x_{k-1} + B_k u_k + w_{k-1}} \tag{4.22}$$

$$\boldsymbol{z_k = H_k x_k + v_k} \tag{4.23}$$

Where:

$\boldsymbol{A_k}$, is a n by n matrix known as the transition matrix at time k,

$\boldsymbol{B_k}$, is a p by p matrix known as the input matrix at time k,

$\boldsymbol{H_k}$, is a m by n matrix known as the measurement matrix at time k,

$\boldsymbol{x_k}$, is a vector of n elements, representing the state of the system at time k,

$\boldsymbol{x_{k-1}}$, is a vector of n elements, representing the state of the system at time k,

$\boldsymbol{u_k}$, is a vector of p elements, representing the input of the system at time k,

$\boldsymbol{z_k}$, is a vector of m elements, representing the measurements of the system at time k,

$\boldsymbol{w_k}$ and $\boldsymbol{v_k}$, are vectors, representing the noise of the system at time k, both of which are assumed to be zero mean and Gaussian distributed. $\boldsymbol{Q_k}$ and $\boldsymbol{R_k}$ are used to characterise the variance of the noise at time k.

These equations are used to formulate a recursive filtering algorithm. In such a formulation, it is necessary to consider two distinct sets of equations. The first set is concerned with the prediction of the state of the system based on the previous best estimate, whilst the second involves the updating of the predicted best estimate by combining the prediction with a new measurement.

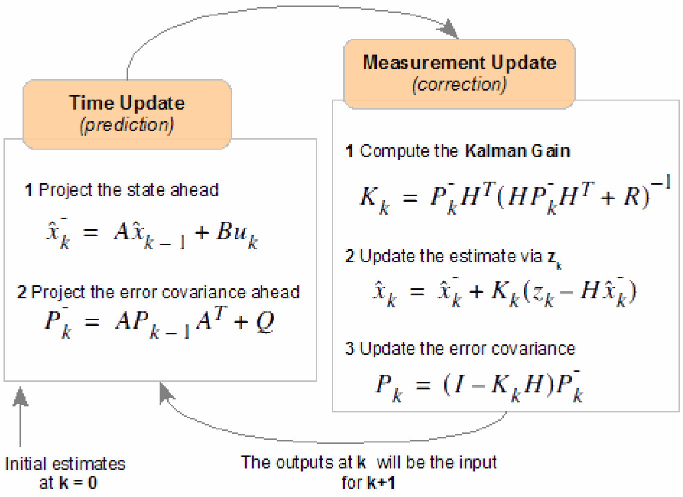

Time Update
(prediction)
1 Project the state ahead
$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_k$
2 Project the error covariance ahead
$P_k^- = AP_{k-1}A^T + Q$
Measurement Update
(correction)
1 Compute the Kalman Gain
$K_k = P_k^- H^T (HP_k^- H^T + R)^{-1}$
2 Update the estimate via z_k
$\hat{x}_k = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-)$
3 Update the error covariance
$P_k = (I - K_k H)P_k^-$
Initial estimates
at k = 0
The outputs at k will be the input
for k+1

Figure 22 Flow of the Kalman filter algorithm using difference equations [67]

5.2.1 Prediction stage

The best estimate at time k is denoted by $\boldsymbol{\hat{x}_k}$. As the system noise, $\boldsymbol{w_k}$, is assumed to have zero mean it can be removed from the difference equation, with its covariance only affecting the system. Therefore, the best estimate of the system is as follows:

$$\boldsymbol{\hat{x}_k^* = A\hat{x}_{k-1} + Bu_k} \tag{4.24}$$

Due to the lack of control commands in the system being considered $\boldsymbol{Bu_k}$ can be dropped simplifying the algorithm, therefore:

$$\widehat{\boldsymbol{x}}_{\boldsymbol{k}}^{*} = \boldsymbol{A}\widehat{\boldsymbol{x}}_{\boldsymbol{k-1}} \tag{4.25}$$

The expected error covariance at time k is given by the following:

$$\boldsymbol{P}_{\boldsymbol{k}}^{*} = \boldsymbol{A}\boldsymbol{P}_{\boldsymbol{k-1}}\boldsymbol{A}^{\boldsymbol{T}} + \boldsymbol{Q} \tag{4.26}$$

5.2.2 Measurement correction stage

On arrival of a new measurement $\mathbf{y}_{k+1}$, at time t_{k+1}, it is compared with the prediction of that measurement derived from the system model. The measurement is then used to update the prediction to generate a best estimate, following the procedure outlined in the previous section. Hence, the best estimate of the state at time t_{k+1} is given by:

$$\widehat{\boldsymbol{x}}_{k} = \widehat{\boldsymbol{x}}_{\boldsymbol{k}}^{*} + \boldsymbol{K}_{\boldsymbol{k}}(\boldsymbol{z}_{\boldsymbol{k}} - \boldsymbol{H}\widehat{\boldsymbol{x}}_{\boldsymbol{k}}^{*}) \tag{4.27}$$

with the covariance given by the following:

$$\boldsymbol{P}_{\boldsymbol{k}} = (\boldsymbol{I} - \boldsymbol{K}_{\boldsymbol{k}}\boldsymbol{H})\boldsymbol{P}_{\boldsymbol{k}}^{*} \tag{4.28}$$

and the Kalman gain matrix, $\boldsymbol{K}_{\boldsymbol{k}}$, given by the following:

$$\boldsymbol{K}_{\boldsymbol{k}} = \boldsymbol{P}_{\boldsymbol{k}}^{*}\boldsymbol{H}^{\boldsymbol{T}}\left(\boldsymbol{H}\boldsymbol{P}_{\boldsymbol{k}}^{*}\boldsymbol{H}^{\boldsymbol{T}} + \boldsymbol{R}\right)^{-\mathbf{1}} \tag{4.29}$$

Before continuing onto further descriptions of other possible fusion filters, the underlying restriction of the Kalman filter must be discussed: the observations of the system's state must be linear. Unfortunately for the system being designed around this assumption is not true, in fact, it is rare for a real-world system to have this sort of linearity. Furthermore, this assumption plays an important part in the flow of the Kalman filter algorithm, as by applying a linear transformation to a gaussian random variable results in a new gaussian random variable which is essential for the updating stage of the filter. Figure 23 shows the results of applying both a linear and non-linear function to a gaussian random variable, with the bottom right of each graph being the initial variable, the top right being the function and the top left being the resulting variable. In the case of Figure 23a the resulting variable has a gaussian distribution with a mean $a\mu + b$ and a variance $a^2\sigma^2$, as represented by the grey area of $p(y)$. To determine and present the distribution of the resulting variable after undergoing a non-linear transformation is a far more complicated process as there are no identities to follow. Instead an estimate was formed using 500,000 samples that have been passed through the function g, and then histogramming of the range of g. The gaussian of $p(y)$ (solid line seen in Figure 23b) has been approximated using the Monte-Carlo estimate which computes the mean and covariance of all 500,000 sample points; this approximation will be taken as the gold standard.

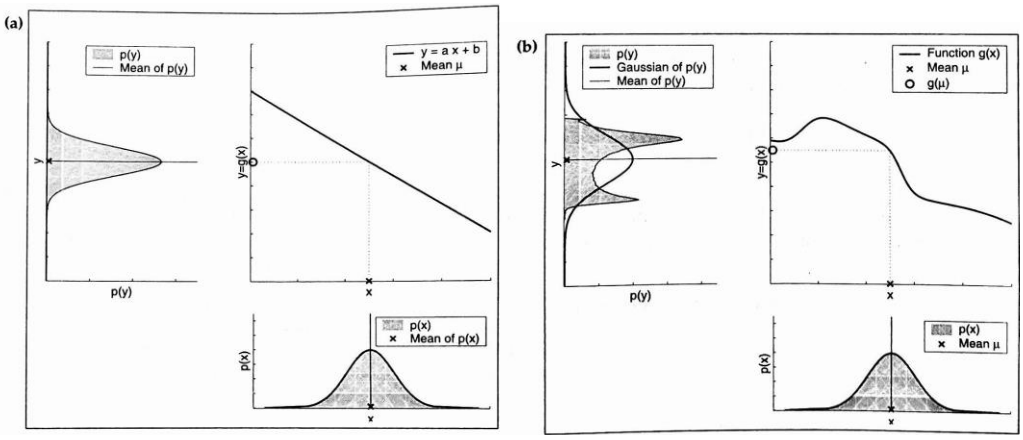


Figure 23 a) Linear and b) non-linear transformation of a gaussian random variable. Bottom right: mean and distribution of the argument for the system, p(x). Top right: the function representing the system. Top left: mean and distribution of the output of the system, p(y), bold line: the gold standard gaussian of the output. [68]

5.3 Extended Kalman Filter

The Extended Kalman Filter (EKF) as the name suggests is part of the Kalman filter family, and works very similarly to the original Kalman filer, however, does not require the system to be linear. The key difference between the EKF and the Kalman filter is that the EKF has an additional step, locally linearizing the system before applying basic Kalman filter steps. There are several methods for local linearization with the most commonly used being the through the Taylor expansion. As with the case of the Kalman filter the control vector will be dropped as it is irrelevant in the system being developed.

The Taylor expansion forms a local linear approximation of non-linear functions through partial derivatives, as seen below:

$$a'(x_{k-1}) \coloneqq \frac{\delta a(x_{k-1})}{\delta x_{k-1}} \tag{4.30}$$

Where $a(x_{t-1})$ is a non-linear function representing the dynamics of the system, i.e. the state and control of the system. The value of a and its slope is dependent on the on the input of the system, which needs to be the most likely state at time $k-1$. In the case of a gaussian variable the most likely argument is the mean, i.e. $\hat{x}_{t-1}$. The resulting first order linear Taylor expansion approximation can be seen below:

$$a(x_{k-1}) \approx a(\hat{x}_{k-1}) + a'(\hat{x}_{k-1})(x_{k-1} - \hat{x}_{k-1}) \tag{4.31}$$

This approximation can be split into three parts:

1. $a(\hat{x}_{k-1})$, determine the result of the function at the known best estimate
2. $a'(\hat{x}_{k-1})$, find the first derivative at this point in the function
3. $(x_{k-1} - \hat{x}_{k-1})$, determine how far the state is from the best estimate

$a'(\hat{x}_{k-1})$ is often referred to as the Jacobian, $\boldsymbol{A_k}$, which is a matrix of $n \times n$, where n denotes the dimension of the state x. This same linearization technique is used on the measurement function h, this time, $\hat{x}_k^-$, represents the most likely state and is used as the argument for the Taylor expansion.

$$h(x_k) \approx h(\hat{x}_k^-) + h'(\hat{x}_k^-)(x_t - \hat{x}_k^-) \tag{4.32}$$

$h'(\hat{x}_k^-)$ is often referred to as the Jacobian, $\boldsymbol{H_k}$, which is a matrix of $n \times n$, where n denotes the dimension of the state x.

With the linearization process complete, only a few minor adjustments must be made to the original Kalman filter algorithm to get the following:

Prediction stage:

1. $\hat{x}_k^* = a(\hat{x}_{k-1})$ originally: $\hat{x}_k^- = \boldsymbol{A}\hat{x}_{k-1}$
2. $\boldsymbol{P_k^*} = \boldsymbol{A_k P_{k-1} A_k^T} + \boldsymbol{Q_k}$

Measurement correction stage:

3. $\boldsymbol{K_k} = \boldsymbol{P_k^* H_k^T}\left(\boldsymbol{H_k P_k^* H_k^T} + \boldsymbol{R_k}\right)^{-1}$
4. $\hat{x} = \hat{x}_k^* + \boldsymbol{K_k}\left(z_k - h(\hat{x}_k^*)\right)$ originally: $\hat{x}_k = \hat{x}_k^- + \boldsymbol{K_k}(z_k - \boldsymbol{H\hat{x}_k^-})$
5. $\boldsymbol{P_k} = (\boldsymbol{I} - \boldsymbol{K_k H_k})\boldsymbol{P_k^*}$

Further adjustments are also made to the definitions of $\boldsymbol{A_k}$ and $\boldsymbol{H_k}$ which now are the Jacobians mentioned above. Completing the entire EKF algorithm.

Before applying the EKF to real-world systems a couple limitations must be considered, the linearity and the uncertainty of the function. Firstly, in a practical non-linear system the resulting function will have parts that act more linear and parts that act more non-linear. This is important to remember as it means that sometimes the linearization by Taylor expansion will have varying rates of success. This can be seen in below, when the focus is centred around a particularly non-linear part of the function (Figure 24a) the mismatch of the EKF gaussian and the gold standard gaussian is far greater than when centred around a more linear part of the function (Figure 24b). The uncertainty of the argument also plays a big part in the effectiveness of the EKF; due to a greater range of the non-linear function holding a significant weighting. This effect can be seen in Figure 25, when the uncertainty of the argument is low, the non-linearity of the function has less effect on both the output distribution and the effectiveness of the EKF.

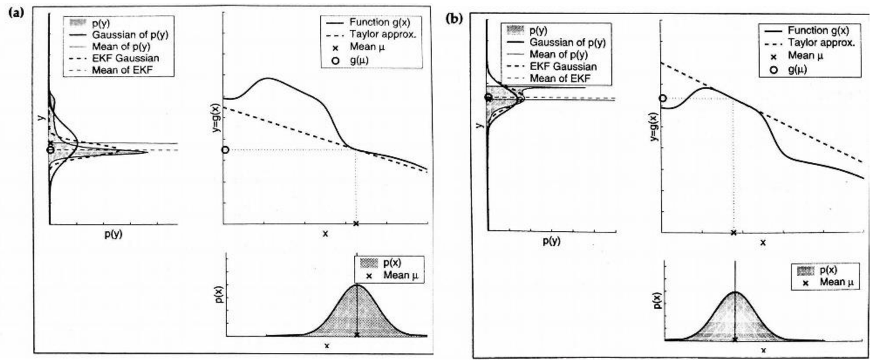


Figure 24 a) High local non-linearity and b) low local non-linearity of non-linear function representing the system. Bottom right: mean and distribution of the argument for the system, p(x). Top right: the function representing the system, dashed line is the linear approximation generated by the EKF. Top left: mean and distribution of the output of the system, p(y), bold line: the gold standard gaussian of the output, dashed line: the mean generated by the EKF, bold dashed line: the gaussian approximation generated by the EKF. [68]

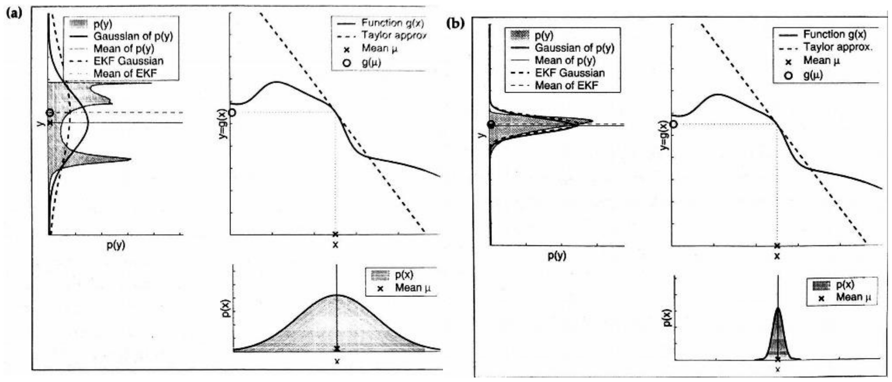


Figure 25 a) High uncertainty and b) low uncertainty of argument being passed through the system. Bottom right: mean and distribution of the argument for the system, p(x). Top right: the function representing the system, dashed line is the linear approximation generated by the EKF. Top left: mean and distribution of the output of the system, p(y), bold line: the gold standard gaussian of the output, dashed line: the mean generated by the EKF, bold dashed line: the gaussian approximation generated by the EKF. [68]

5.4 Unscented Kalman Filter

As mentioned previously the Taylor expansion is not the only way of locally linearizing a non-linear system, another method that is computationally cheap enough to be used in real time filtering is the unscented transform. The foundation of this transform is determining an optimal linear estimate using sample points. There other several techniques that use sample points to characterise a function, what makes the unscented transform unique is the deterministic procedure which extracts sigma points from the argument. The location of these sigma points generally gives more overlap when considering the resulting gaussian and the gold standard gaussian, compared to randomly chosen sample points.

The following describes the procedure for setting up the unscented transform algorithm:

- The sigma points are located at the mean and then in pairs symmetrically along the main axes of the covariance. The number of points chosen is equal to the system's dimension (n) plus one i.e. $2n + 1$.

$$\mathcal{X}^{[i]} = \mu + \left(\sqrt{(n+\lambda)\Sigma}\right)_i, \qquad \text{for } i = 1, \ldots, n. \tag{4.33}$$

$$\mathcal{X}^{[0]} = \mu \tag{4.34}$$

$$\mathcal{X}^{[i]} = \mu - \left(\sqrt{(n+\lambda)\Sigma}\right)_i, \qquad \text{for } i = 1, \ldots, 2n. \tag{4.36}$$

Where $\mathcal{X}$ is the sigma point and $\lambda = \alpha^2(n+\kappa) - n,$ where α and κ are scaling parameters that determine the spread of the sigma points.

- Once the locations of the sigma points are determined a weighting is applied to the points to recover the mean and covariance the output gaussian. A separate weighting is associated with the mean, ω_m, and the covariance, ω_c.

$$\omega_m^{[0]} = \frac{\lambda}{n+\lambda} \tag{4.37}$$

$$\omega_c^{[0]} = \frac{\lambda}{n+\lambda} + (1 - \alpha^2 + \beta) \tag{4.38}$$

$$\omega_m^{[i]} = \omega_c^{[i]} = \frac{1}{2(n+\lambda)}, \qquad \text{for } i = 1, \ldots, 2n. \tag{4.39}$$

Where β is an additional parameter that can be used to give the algorithm some further knowledge on the distribution characteristics of the Gaussian representation. For an argument which has exactly a gaussian distribution β will be equal to 2.

- Before applying these weightings, the sigma points must first be passed through the system function, g, to determine how the nonlinear system will affect the gaussian shape.

$$\mathcal{Y}^{[i]} = g(\mathcal{X}^{[i]}) \tag{4.40}$$

- To get the updated mean and covariance the following expression are applied using the sigma points and the weightings:

$$\mu_y = \sum_{i=0}^{2n} \omega_m^{[i]} \mathcal{Y}^{[i]} \tag{4.41}$$

$$\Sigma_y = \sum_{i=0}^{2n} \omega_c^{[i]} (\mathcal{Y}^{[i]} - \mu_y)(\mathcal{Y}^{[i]} - \mu_y)^T \tag{4.42}$$

As mentioned before, the UKF uses the unscented transform to linearize the system before continuing to apply the familiar original Kalman filter "predict and update" algorithm. The inclusion of the sigma points makes adding the unscented transform to the Kalman filter algorithm less straightforward than when doing the same with the Taylor expansion and the EKF.

To begin with the sigma points are chosen following the same method as described previously in the unscented transform algorithm.

$$\mathcal{X}_{k-1} = \left(\mu_{k-1} \qquad \mu_{k-1} + \sqrt{(n+\lambda)\Sigma_{k-1}} \qquad \mu_{k-1} - \sqrt{(n+\lambda)\Sigma_{k-1}}\right) \tag{4.43}$$

These sigma points are then applied to the system as arguments of the non-linear function g:

$$\mathcal{X}_k^* = g(\mathcal{X}_{k-1}) \tag{4.44}$$

The results of passing the sigma points through the system are then weighted using the weights determined using the method described previously. This weight result is then used to calculate the mean and covariance of the prediction made.

$$\mu_k^* = \sum_{i=0}^{2n} \omega_m^{[i]} \mathcal{X}_k^{*[i]} \tag{4.45}$$

$$\Sigma_k^* = \sum_{i=0}^{2n} \omega_c^{[i]} \left(\mathcal{X}_k^{*[i]} - \mu_k^*\right)\left(\mathcal{X}_k^{*[i]} - \mu_k^*\right)^T + Q_k \tag{4.46}$$

Using the predicted mean and covariance calculated above a new set of updated sigma points are computed.

$$\mathcal{X}_k^{**} = \left(\mu_k^* \qquad \mu_k^{**} + \sqrt{(n+\lambda)\Sigma_k^{**}} \qquad \mu_k^{**} - \sqrt{(n+\lambda)\Sigma_k^{**}}\right) \tag{4.47}$$

Next these updated sigma points are passed through the function, h, which represents the observable system, to determine the predicted observation.

$$\mathcal{Z}_k = h(\mathcal{X}_k^{**}) \tag{4.48}$$

The resulting observation sigma points, $\mathcal{Z}_k$, are used to calculate the predict observation z_k^* and its uncertainty, S_k.

$$z_k^* = \sum_{i=0}^{2n} \omega_m^{[i]} \mathcal{Z}_k^{[i]} \tag{4.49}$$

$$S_k = \sum_{i=0}^{2n} \omega_c^{[i]} \left(\mathcal{Z}_k^{[i]} - z_k^*\right)\left(\mathcal{Z}_k^{[i]} - z_k^*\right)^T + R_k \tag{4.50}$$

Next the cross-covariance is determined by comparing the similarity of the predicted state and the observation of the system (the measurement). This is then used to calculate the Kalman gain, K_k.

$$\Sigma_k^{x,z} = \sum_{i=0}^{2n} \omega_c^{[i]} \left(\mathcal{X}_k^{**[i]} - \mu_k^*\right)\left(\mathcal{Z}_k^{[i]} - z_k^*\right)^T \tag{4.51}$$

$$K_k = \Sigma_k^{x,z} S_k^{-1} \tag{4.52}$$

Having calculated the Kalman gain the last couple of step to the UKF algorithm do not need adjustments from the original Kalman filter algorithm. Therefore the "complete" mean and covariance can be determined using the following expressions:

$$\mu_k = \mu_k + K_k(z_k - z_k^*) \tag{4.53}$$

$$\Sigma_k = \Sigma_k^* - K_k S_k K_k^T \tag{4.54}$$

Just like the case of the EKF, the efficiency of the UKF is dependable on the system it is being applied to. Once again, the two main factors to consider is the linearity of the system and the uncertainty of the arguments within the system. Figure 26 shows the effect local non-linearity within the system can influence the UKF. As before, the output gaussian has a larger mismatch with gold standard gaussian when the mean of the argument is near a particularly non-linear part of the system. However, it is also clear from the figure that when comparing to the EKF this mismatch is far less. This can also be seen in Figure 27. When the uncertainty of the argument increases so does the mismatch between the UKF output gaussian and the gold standard gaussian. However once again, when compared to the EKF this mismatch increase is far less.

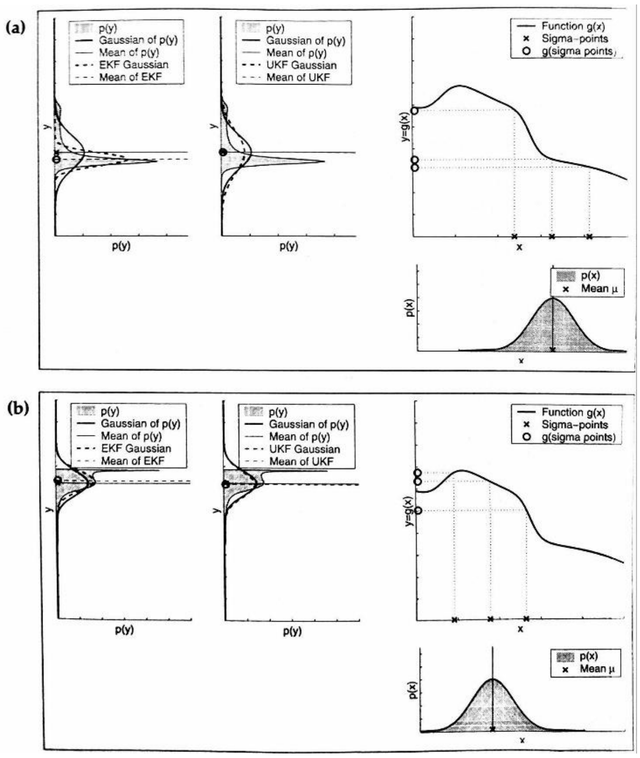


Figure 26 a) High local non-linearity and b) low local non-linearity of non-linear function representing the system. Bottom right: mean and distribution of the argument for the system, p(x). Top right: the function representing the system, dashed line is the linear approximation generated by the EKF. Top centre: mean and distribution of the output of the system, p(y), bold line: the gold standard gaussian of the output, dashed line: the mean generated by the UKF, bold dashed line: the gaussian approximation generated by the UKF. Top left: mean and distribution of the output of the system, p(y), bold line: the gold standard gaussian of the output, dashed line: the mean generated by the EKF, bold dashed line: the gaussian approximation generated by the EKF. [68]

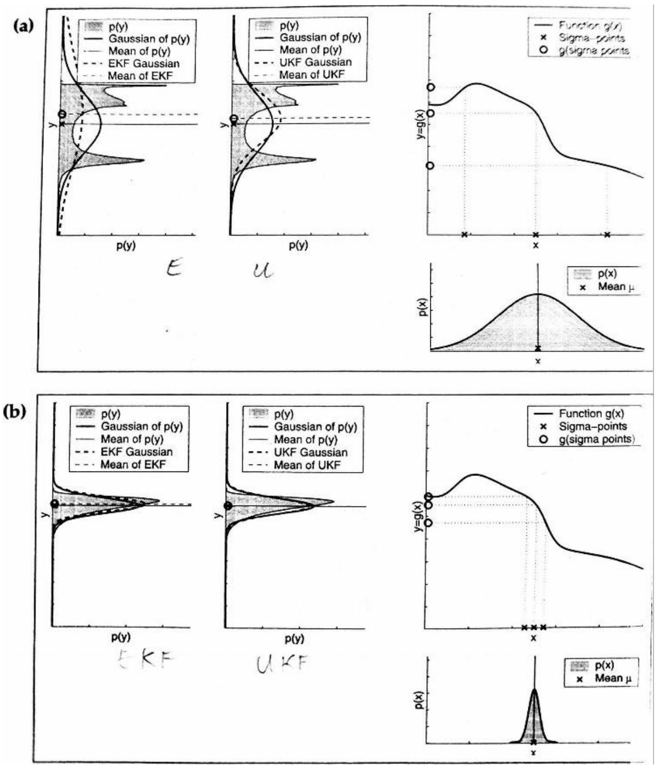


Figure 27 a) High uncertainty and b) low uncertainty of argument being passed through the system. Bottom right: mean and distribution of the argument for the system, p(x). Top centre: mean and distribution of the output of the system, p(y), *bold line: the gold standard gaussian of the output, dashed line: the mean generated by the UKF, bold dashed line: the gaussian approximation generated by the UKF. Top right: the function representing the system, dashed line is the linear approximation generated by the EKF. Top left: mean and distribution of the output of the system, p(y), bold line: the gold standard gaussian of the output, dashed line: the mean generated by the EKF, bold dashed line: the gaussian approximation generated by the EKF. [68]*

5.5 NON-PARAMETER ALGORITHMS

So far, all the fusion filters being considered assume that the argument of the system (the state) is distributed in a gaussian manner. This assumption is currently being considered a valid one, however without a proper test procedure it cannot be known for sure. Once a test procedure is complete the distribution of the state can be determined by taking several samples; if it turns out that the assumed gaussian distribution is incorrect a new set of filters will need to be considered. This new set of filters is often referred to as nonparametric filters and do not make any assumptions on the distribution of the argument. These filters can proceed without making any assumptions by approximating the argument with a finite number of values, each corresponding to a different region in the state space.

The two nonparametric filters that seem promising, however further research will be required, are the complementary filter and the Particle filter. Each has its own approach to approximating the argument over the continuous state space.

5.5.1 Complementary Filter

The complementary filter is a simple filter to understand and implement, as the name would suggest the filter is design to complement multiple filters. Considering Chapter 3 it can be summarised that accelerometers perform well over time however its error ratio is higher with short measurements, while gyroscopes perform poorly over time however can be considered reliable with short measurements as gyroscope drift has little impact. The complementary filter relies on this behaviour to combine a low pass filter on the accelerometer data and a high pass filter on the gyroscope data [69] [70]. Figure 28 shows the basic block diagram for a complementary filter.

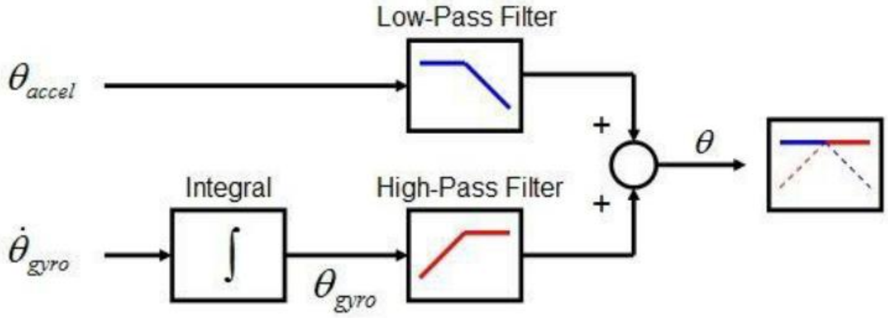


Figure 28 Basic block diagram of a complementary filter.

Let's consider the components of a complementary filter. The transfer function of a low pass filter is $\frac{1}{1+T_S}$ while the transfer function of a high pass filter is $\frac{T_S}{1+T_S}$. Therefore, the complementary filter's transfer function can be represented as:

$$Output = \frac{\theta_{accel}}{1+T_S} + \left[\frac{T_S}{S(1+T_S)} + \frac{1}{S}\right] \times \theta_{gyro} \qquad (4.55)$$

Next let's consider how the complementary filter will handle noise in the system. The total gain of the two filters is 1, therefore the following can be said:

$$Low\ pass\ gain = U_S \tag{4.56}$$

$$High\ pass\ gain = 1 - U_S \tag{4.57}$$

Let the final output signal be represented by W_S, the noiseless input by V_S, the noise associated with the accelerometer data by N_a, and the noise associated with the gyroscope data by N_g. Therefore, it can be proven that the gain of the filter only affects the noise components of the data:

$$W_S = (1 - U_S) \times (V_S + N_a) + U_S \times \left(V_S + N_g\right) \tag{4.58}$$

$$= V_S + (1 - U_S) \times N_a + U_S N_g \tag{4.59}$$

Note that the above is only true because the data is complementary and therefore each input can be considered by one variable.

Literature reviews have suggested the optimal gain balance for the complementary filter is 98% gyroscope data and 2% accelerometer data [71].

The one disadvantage of using the complementary filter is that in its basic form it cannot adapt to the sensor system, i.e. the gain values are set. This would be a concern for highly volatile environments where the noise present at each sensor (accelerometer and gyroscope) changes throughout the measurements.

5.5.2 Particle filter

The particle filter represents the argument by a finite number of random samples. The particle filter can be adjusted to act as a resource adaptive filter which adapts the number of samples used so that the entire available computation resources is used. Figure 29 shows the particle filter when being applied to the non-linear function previously present to the Kalman filter, EKF and UKF.

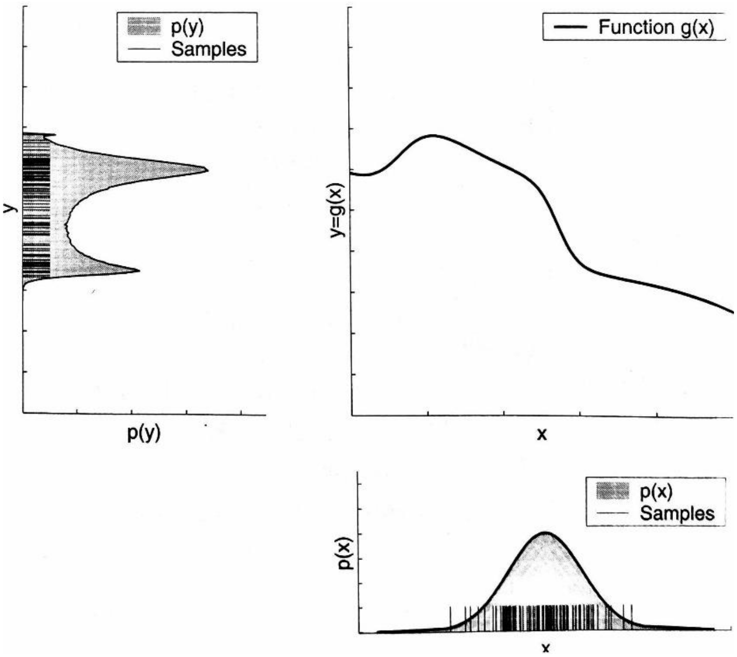


Figure 29 Particle representation of a gaussian distributed variable. Bottom right: mean and distribution of the argument for the system, p(x), the particle samples of this distribution are overlaid. Top right: the function representing the system. Top left: mean and distribution of the output of the system, p(y), the particle samples of this distribution are overlaid. [68]

The particle filter algorithm can be split into three main parts: initialisation, importance analysis and resampling.

5.5.2.1 Initialisation

To begin the initialisation-process the initial location for the particles must be chosen. This can either be done by stating a location and bounds of which the particles must adhere to, or by stating a location mean and covariance. The first method is often used for system with unknown starting states allowing the used to guess within the systems limits, for example between -180 degrees and 180 degrees for orientation. The second method is used for systems where the initial state is approximately known, for example if the sensor starts off stationary. Let the initial set of particles be known as X_t set which will contain M particles.

Next a state transition function must be given to the particle filter describing how the system responses to an input. When considering an IMU system the following state transition function has been proven to be a good representation of the system [72]:

$$f_k(x_k, u_k) = \begin{bmatrix} I_3 & -\Delta t[C_3 \times]_k \\ 0_{3\times3} & I_3 \end{bmatrix} x_k + \begin{bmatrix} \Delta t[C_3 \times]_k \\ 0_{3\times3} \end{bmatrix} u_k \tag{4.60}$$

Where $[C_3 \times]$ is the rotation operator of the sensors, x_k is the state vector and u_k is the gyroscope measurements used as a control input.

With the initial particles step up and the approximated state transition function given, the particle filter can predict the next state. This prediction then goes through the importance analysis to determine its value.

5.5.2.2 *Importance Analysis*

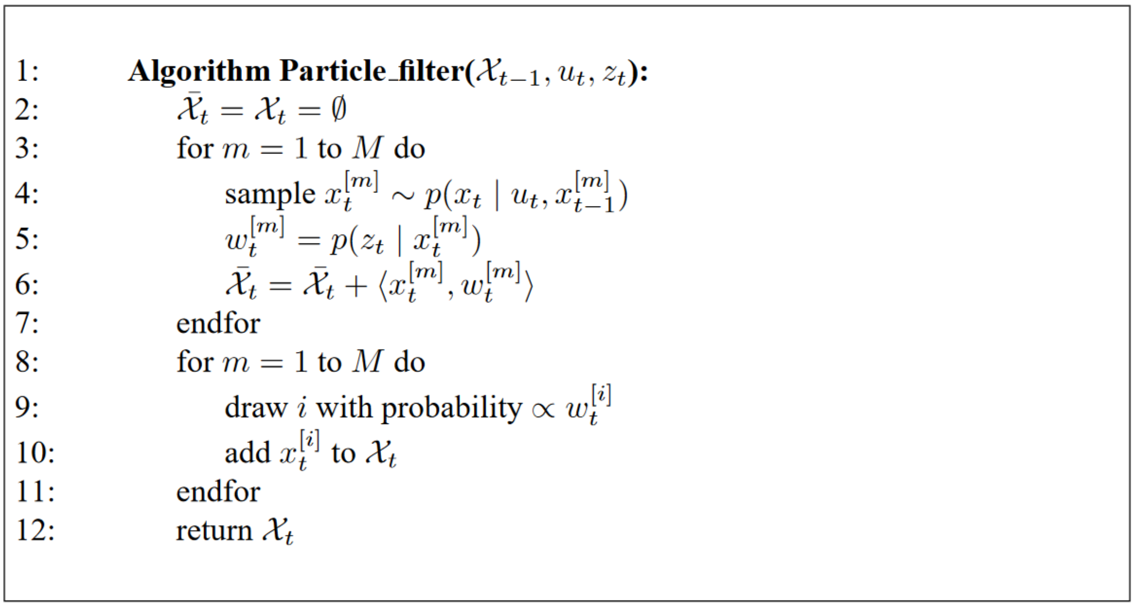
Once a prediction is made it is compared to the sensors' measurements where it is given a weight indicating its likelihood. This weighting technique is often referred to as importance factor or likelihood function. Once all the particles have been given an importance weighting, they are processed through to the resampling section of the particle filter.

5.5.2.3 *Resampling*

The algorithm draws with replacement M particles from the temporary set X_t. The probability of drawing each particle is given by its importance weight. Resampling transforms a particle set of M particles into another particle set of the same size.

5.5.2.4 *Algorithm overview*

Figure 30 gives an overview of the particle filter algorithm. The main benefit of the particle filter is that it does not assume any sort of distribution. However, it requires high computational power to handle the large amount of particle present.



```
Algorithm Particle_filter($\mathcal{X}_{t-1}, u_t, z_t$):
    $\bar{\mathcal{X}}_t = \mathcal{X}_t = \emptyset$
    for $m = 1$ to $M$ do
        sample $x_t^{[m]} \sim p(x_t \mid u_t, x_{t-1}^{[m]})$
        $w_t^{[m]} = p(z_t \mid x_t^{[m]})$
        $\bar{\mathcal{X}}_t = \bar{\mathcal{X}}_t + \langle x_t^{[m]}, w_t^{[m]} \rangle$
    endfor
    for $m = 1$ to $M$ do
        draw $i$ with probability $\propto w_t^{[i]}$
        add $x_t^{[i]}$ to $\mathcal{X}_t$
    endfor
    return $\mathcal{X}_t$
```

Figure 30 Overview of the particle filter algorithm

5.6 TESTING THE DISCUSSED FUSION FILTERS

Considering the test procedures described in Chapter 4, a test using precise actuators was setup to evaluate the fusion filters discussed in this chapter. Unfortunately, a CMM could not be acquired, instead the L-611 Precision Rotation Stage was used [73]. The MMR was mounted to the stage three times, each time with a different axis aligned to the axis of rotation. The stage was then set to rotate for 20 minutes at a constant rate. The resulting data was adjusted for any misalignment using the method described in Chapter 3, and then applied to the following filters: EKF, UKF, Complementary filter, the Particle filter with 100 particles and the Particle filter with 10000 particles. The particle filter was setup using 100 and 10000 particles to compare its performance to itself and the other filters. 100 particles were chosen as it resulted in approximately the same processing time as the other fusion filters used. 10000 particles were used to indicate the full potential of the particle filter if the system had no computational restrictions. The results for the yaw rotation will be analysed in this section as they exhibit the greatest differences between the filters.

5.6.1 Time domain

Figure 31 shows the response each filter has to the continual rotation around the yaw axis compared to the actual orientation of the MMR. None of the filters line up with the actual orientation on the time domain but are instead offset by varying degrees. As it has been determined in the aims of this project it is unnecessary for the system to work live and therefore this time shift does not affect the performance of the system. However, it is also clear from the figure that this offset for some of the filters is not constant which means the resulting RPM (Rotation per Minute) would be incorrect. This is particularly visible with the particle filter with 100 particles, just within this window of 25 seconds it shifts from leading the Complementary filter to synchronising it.

In terms of shape all except the particle filter with 100 particles have the correct shape. The particle filter with 100 particles has a slight curve where it has had to correct itself as it moves away from the measurements. As the L-611 precision rotary stage does not have any means to synchronise with the MMR device a novel approach to validate the fusion filters in the frequency domain was taken. By evaluating the filters in the frequency domain the lag between the L-611 and the filter would become irrelevant.

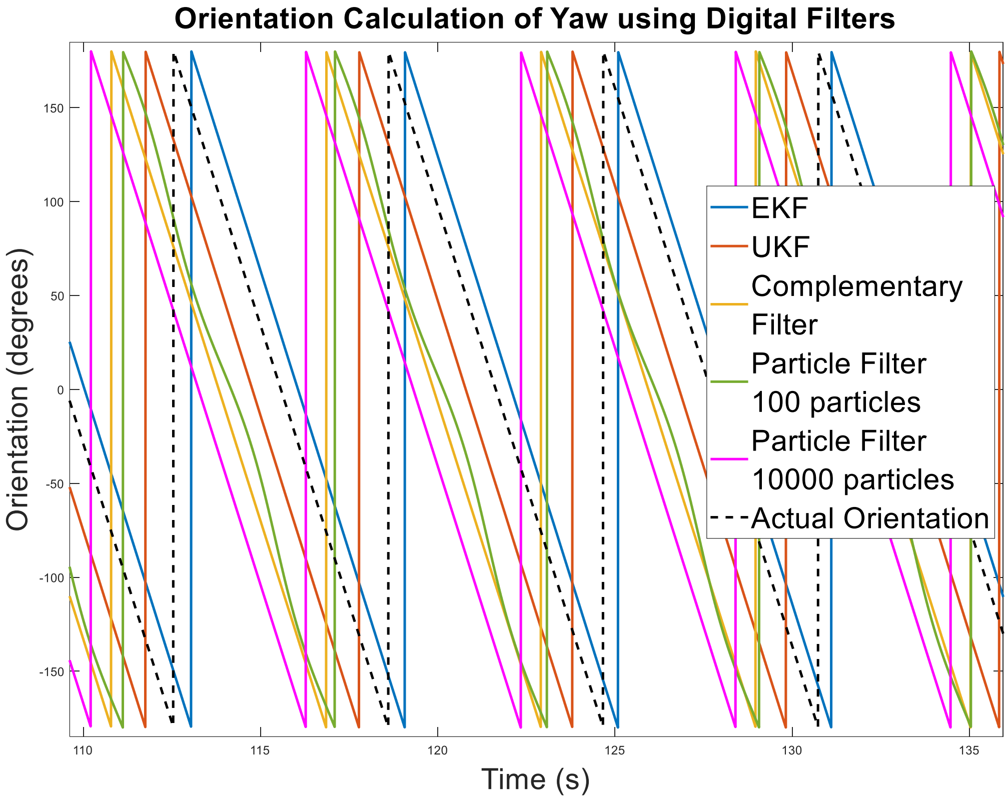


Figure 31 Time domain response of the following filters to 20 minutes of rotation around the Yaw axis: EKF, UKF, Complementary filter, Particle filter with 100 particles and Particle filter with 10000 particles.

5.6.2 Frequency domain

Figure 32 shows the frequency response each filter has to continual rotation around the yaw axis, including the frequency response of the L-611. All five filters seem to perform well at match the systems low harmonic frequencies however at the higher harmonic frequencies it becomes clear that some of the filters are responding on a different harmonic. All the filters successfully remove the majority of noise between the harmonics.

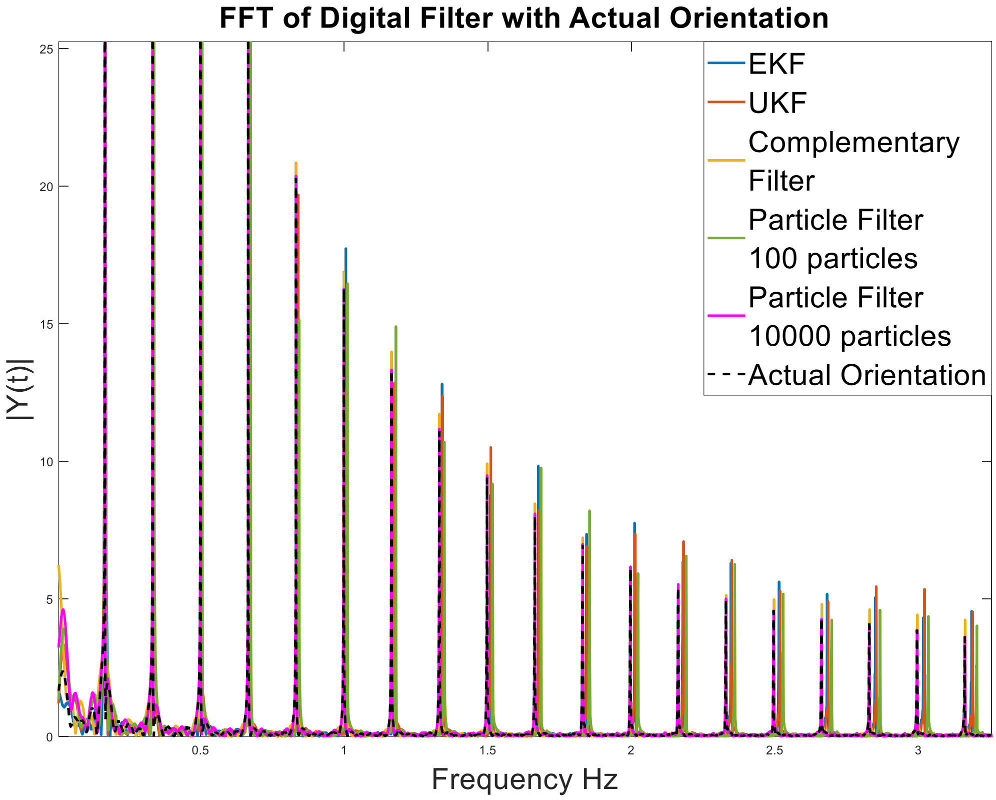


Figure 32 Frequency response of the following filters to 20 minutes of rotation around the Yaw axis: EKF, UKF, Complementary filter, Particle filter with 100 particles and Particle filter with 10000 particles.

The actual rotation of the L-611 is 9.983 RPM, whereas by analysing Figure 33 the following RPM values were calculated from the filters' frequency responses:

Fusion Filter	RPM
Particle Filter with 100 particles	10.115
Extended Kalman Filter	10.069
Unscented Kalman Filter	10.058
Complementary Filter	9.987
Particle Filter with 10000 particles	9.983

From these results it has been decided to consider both the Particle filter and the Complementary filter moving forward in this project. However, the time taken to computer these responses should also be consider, as seen below. The actual timing for the process will depend on the processing power of the computer used however the particle filter's processing time is concerning as this would scale up with the number of sensors used. This could result in a processing time of several hours for a full body analysis excluding other processing required of the system.

Fusion Filter	Computational time compared to the UKF (actual time)
Particle Filter with 100 particles	0.89 (7 minutes 40 seconds)
Extended Kalman Filter	0.95 (8 minutes 12 seconds)
Unscented Kalman Filter	1 (8 minutes 36 seconds)516
Complementary Filter	0.91 (7 minutes 47 seconds)
Particle Filter with 10000 particles	24.64 (3 hours 31 minutes 56 seconds)

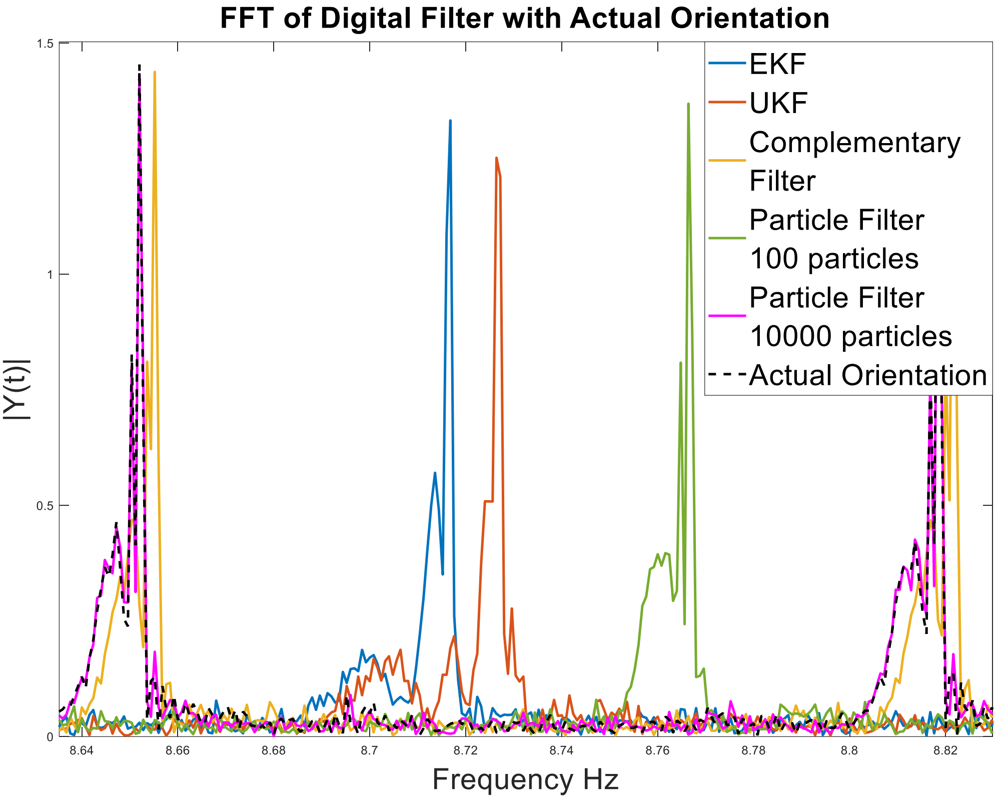


Figure 33 Close up of the 52nd harmonic of the system's frequency response showing the five filters 52nd harmonic frequency response

Figure 34 shows a close up of the 52nd harmonic focusing on the Complementary filter and the Particle filter with 10000 particles. In the figure it is seen that the particle filter does a great job at estimating the actual frequency response; not only having the correct dominant frequency but also the same shape as the actual response. At this scale the difference between the actual frequency response and the Complementary filter's frequency response is clear, however the Complementary filter still does a good job at estimating the correct shape and dominant frequency of the actual rotation. The difference in the RPM values of the Complementary filter and the actual rotation would correspond to an inaccuracy of 1.44°/minute which satisfies the specification of the project made in Chapter 1.

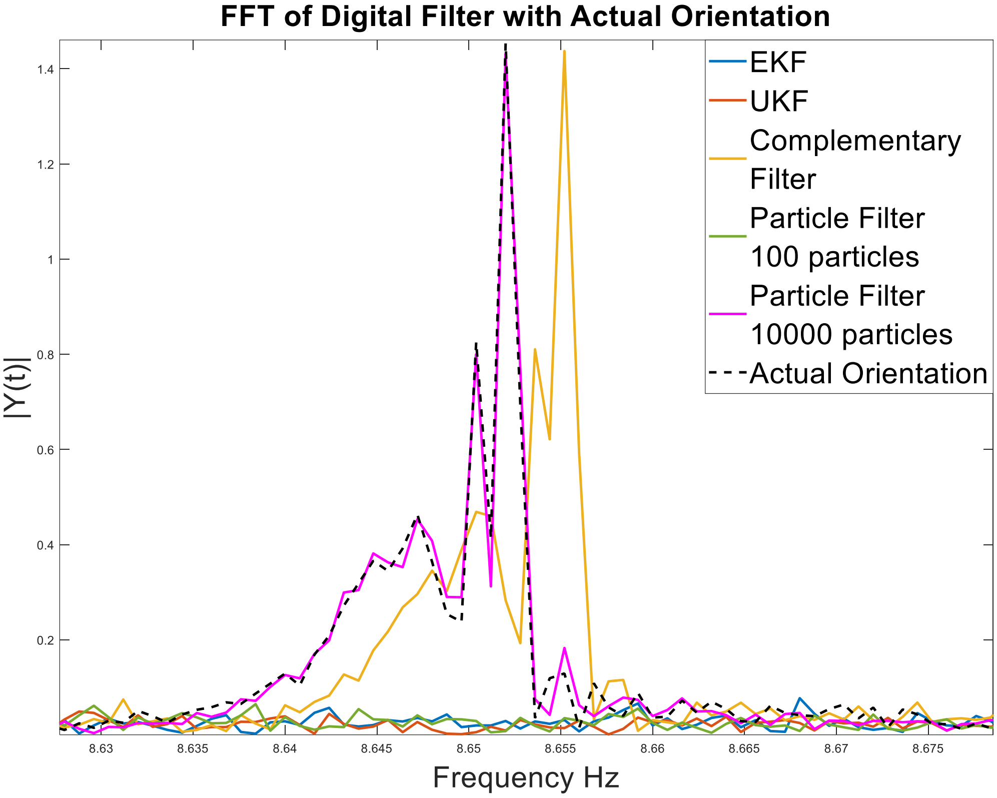


Figure 34 Close up of the 52nd harmonic focusing on the Complementary filter and the Particle filter with 10000 particles

5.7 CHAPTER SUMMARY

In this chapter the theory behind fusion filtering and sensor fusion is outlined, giving the reader an understanding of orthogonal estimates and how they can be used to remove noise in the system. It is concluded that the Kalman filter would not be suitable for this project due to its assumption of linear distribution. The EKF, UKF, Complementary Filter and Particle filter are compared both in the time domain and the frequency domain. It is concluded that the Complementary filter and Particle filter will be used as the project moves forward with the understanding that the Complementary filter is less accurate than the Particle filter but requires considerably less computational power.

6 EXPERIMENTAL SETUP

This chapter will go through the process used to collect both the Shaolin Kung Fu and Parkinson's datasets. This will include all steps required from setting up the IMU sensors to digitally filtering noise out of the measurements. Note the Shaolin Kung Fu practitioners participating in this study have all given their consent for their motion data to be used. Additionally, the participant in Figure 1, Figure 2, Figure 3, Figure 4, Figure 5, Figure 6, and Figure 38 has given their consent for these images to be used in this thesis.

Before going into the specifics of the experimental setup and the procedure used, a summary of each dataset's experimental setup has been included below.

The Shaolin Kung Fu dataset is comprised of 8 skilled practitioners, 3 females and 5 males, each of whom repeated Wu Bu Quan 40+ times, resulting in a dataset with 5 classes each of which have 350+ samples. Additionally, the practitioners have also stretched and warmed up while wearing the IMU bodysuit resulting in a 6th "other" class, comprising of 1600+ samples. In total the dataset comprises of 6 classes and 3573 samples. The dataset was labelled from video recordings by experts from the Shaolin Kung Fu school, 6 label were used: 0 = other, 1 = Gong Bu, 2 = Ma Bu, 3 = Xie Bu, 4 = Pu Bu, and 5 = Xu Bu. The raw data was post processed using a Complementary filter.

The movements done for the modified UPDRS tapping datasets were self-emulated after being coached on the movements by a researcher in neuropsychological assessment. The modified UPDRS tapping test was repeated 100 times for each of the 5 severity levels with a IMU device attached to both the finger and the wrist. The tests were done using only the dominant hand. The datasets were then labelled by a researcher in neuropsychological assessment and validated by their colleagues. Some of the recordings were not validated and therefore removed from the dataset resulting in the following distribution of classes: 0 = 97, 1 = 99, 2 = 99, 3 = 99, 4 = 99. In total the dataset comprises of 5 classes and 493 samples. The raw data was post processed using a low pass filter.

6.1 SENSOR LOCATION

In this section the location of the sensors for each dataset will be discussed, indicating why the locations were chosen and how the sensors were attached.

6.1.1 Shaolin Kung Fu dataset

Some work has previously been done determining the optimal number of IMU devices required for full body motion capture. *Marcard et al.* have shown that using just 6 IMU devices accompanied with their novel Sparse Inertial Poser (SIP) method full body motion tracking in possible [74]. However, the SIP method is unable to accurately track wrist or ankle joint parameters due to the required IMU placements. Referring to section 2.1 while wrist position and orientation are never mentioned in the movement sets, ankle position and orientation are crucial throughout the movement set. Therefore, it was decided not to use the SIP method. *Geissinger et al.* have created a large full body inertial motion capture dataset of natural motions using the Xsens full body configuration [75] [19]. They explored Seq2Seq and Transformer models for inferring upper-body and full-body kinematics using different configurations of sparse sensors. However, their models produce mean angular errors of 10–15 degrees for both the upper body and full body, and worst-case errors of less than 30 degrees. While at this point in the project it is unclear whether these levels of errors are acceptable for the given movement sets, their results have shown that several human motions can be inferred effectively with a reduced amount of IMU devices. This reduction in sensor dimensionality will be considered in when discussing the results and in the future works of this project.

For the this project the number of sensors and locations used have been chosen following the guidelines set by Xsens [20], however it was decided instead of placing a device on each hand, a device would be place behind each hip. This adjustment would result in a loss of wrist joint parameters and an increase in hip joint parameters. This was decided due to the lack of interest the movement set has on the wrists and the importance hip orientation has on the positions being captured. To aid in the Shaolin Kung Fu dataset, three bodysuits including 14 pockets for the MMR sensors were made. The pockets are located between each major articulating joint: upper arms, forearms, thighs, calves, lower back and upper back. Additional sensor pockets are located on the following minor articulating joints: scapula and above each side of the pelvic bone. A final pocket is located on the centre of the back however it was decided this would not be used. Additional to the sensor pockets, a headband is used to secure a sensor to the user's head and two further sensors are placed in the laces of the user's shoes, giving a total of 17 sensor locations.

Figure 35 shows the bodysuit used to capture the data for the Shaolin Kung Fu dataset. The bodysuit is made from Polyester and Spandex allowing it to both be suitable for physical activity (it is the same material often used in gymnasts' leotards) and remain tight on the body. It is important that the bodysuit is tight on the body as it reduces independent motion of the sensors.

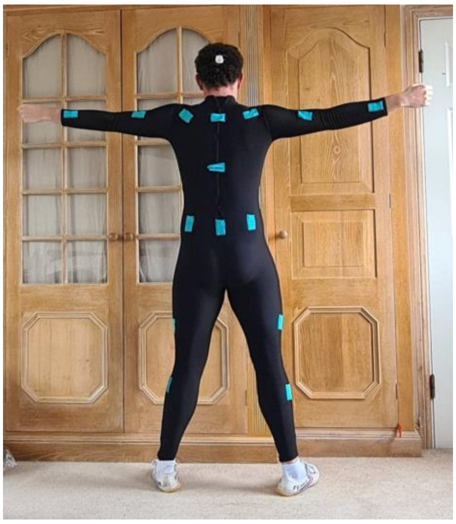


Figure 35 Full bodysuit used to capture data for Shaolin Kung Fu dataset. The light blue shapes on the bodysuit are small pockets for the MMR sensor.

6.1.2 Modified UPDRS datasets

Two possibly locations were explored in the Parkinson's dataset: on the wrist and on the index finger, as seen in Figure 36. Having the sensor on the wrist is ideal as it does not hinder the motion of the finger at all, however, the influence tapping has on the wrist is limited and may not be enough to analyse the movement. The sensor is small enough to fit between the first and second knuckles of the index finger meaning it is unlikely to affect the motion of the finger, however, considering the users would have some severity of Parkinson's even the slightest interference could cause hinderance.

The sensors are secured to the locations using cohesive tape; a tape often used in sports science and in the clinic due its ability to itself without sticking to other materials, for example: skin or clothing. Like with the Shaolin Kung Fu dataset, the sensors will be setup first before attaching them to the user. Even though setup will be a lot quicker for the Parkinson's dataset, considering the users, it may be an issue to keep the sensors steady during the setup process.

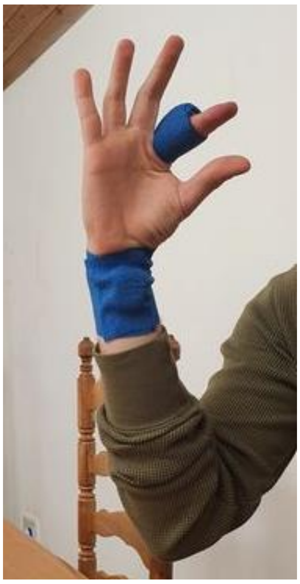


Figure 36 Photo of the two setups used for the Parkinson's dataset. An MMR device is strapped to the wrist and around the base of the index finger using cohesive tape.

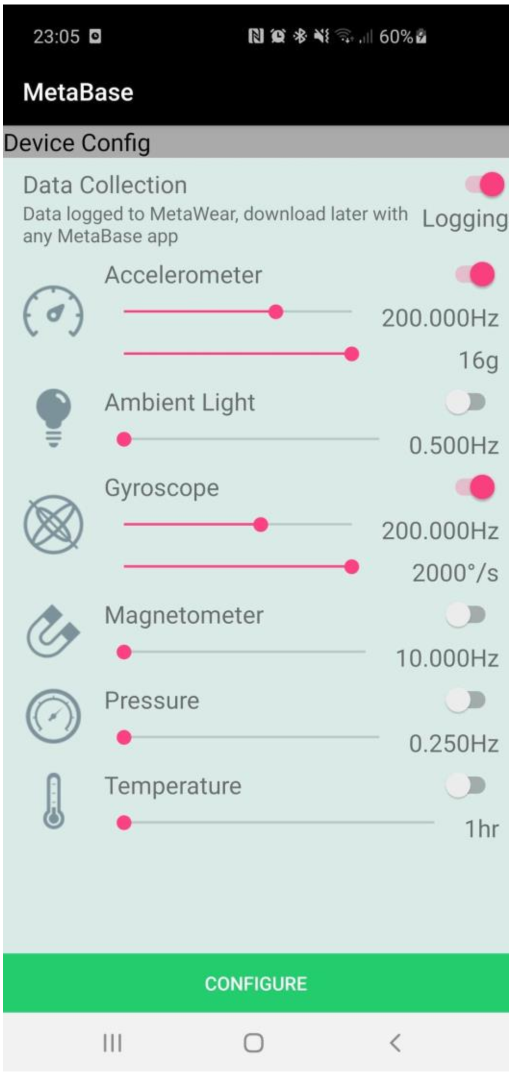
6.2 Sensor setup

This section will simply go through the process of configuring the MMR sensor. As mentioned in Chapter 3, the MMR sensor has sensor fusion options allowing the user to select either Euler angles or quaternions as an output format. It should be noted that if one of these fusion methods are selected, the other method as well as the raw data cannot be outputted. Despite these being the eventual intended formats for this project the MMR will be configured to output the raw data of the accelerometer and gyroscope sensors. The raw data is preferred as it allows the most flexibility for data manipulation, meaning it can then be processed to both orientation formats and compared later.

Figure 37 shows the graphic user interface for Metabase, the application which comes with the MMR sensor [76]. Metabase allows the user to select which sensors are being used and whether the sensor is logging or streaming data. Additionally, it allows the user to select the sampling frequency and measuring range for each sensor individually. As stated in the aims of this project, live analysis is not required in this project and therefore the sensors will be set to logging mode.

The sampling frequency was chosen to be 200 Hz, which has been proven to be the lowest effective frequency for IMU human movement capturing systems [40]. Even though the MMR sensor can have a higher sampling frequency its memory becomes a problem. Sampling at 200 Hz with both the accelerometer and gyroscope fills up the MMR's memory in 22 minutes; this is already a small window for the user to record their measurements without considering that each sensor must be configured separately meaning by the end of configuring all 17 sensors (for the Shaolin Kung Fu dataset) the user will have approximately 15 minutes of recording time.

The measurable range was chosen after doing some trial runs of the selected motions. For both the Shaolin Kung Fu and Parkinson's datasets there were some measurements exceeding the 8g and 1000º/s thresholds. It was therefore decided to use the full range of the accelerometer, 16g, and gyroscope, 2000º/s, sensors.



MetaBase

Device Config

Data Collection
Data logged to MetaWear, download later with any MetaBase app
Logging

Accelerometer
200.000Hz
16g

Ambient Light
0.500Hz

Gyroscope
200.000Hz
2000°/s

Magnetometer
10.000Hz

Pressure
0.250Hz

Temperature
1hr

CONFIGURE

Figure 37 Metabase GUI for sensor configuration. [76]

6.3 The Movement's setup

This section will focus on any additional steps the user must make on top of their desired movements. A description of the desired movements can be found in Chapter 2.

6.3.1 Shaolin Kung Fu dataset

As the features that will be used from this dataset will require orientation calculations, the user has been asked to start in the "I" position, refer to Figure 38. This moment in time is then recorded manually and will be used to synchronise the IMU devices as well calibrate the system to this known orientation. For the training dataset the user has been asked to return to this "I" position between each repetition of the Wu Bu Quan form to maximise the accuracy of the dataset.

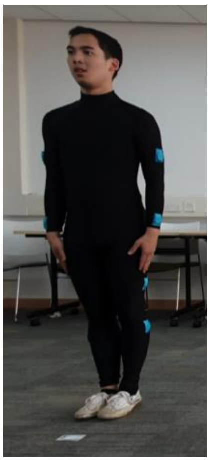


Figure 38 Shaolin Kung Fu practitioner standing in the "I" position.

6.3.2 Modified UPDRS datasets

Although orientation is not one of the features used in this dataset the user is still encouraged to start with their hand on the table and place it back on the table once they are finished the exercise. This process will not affect the accuracy of the dataset however will allow the system to automatically select the relevant window in time. If the user is unable to follow this process the data will have to be manually windowed.

As the dataset was only an emulation of the Parkinson's MDS-UPDRS tapping exercise the users were all able to start and finish by placing their hand on a table.

6.4 CATEGORISING THE DATA

Once the desired movements are completed the next step in the system is categorising the data. This step is both vital and tedious. Due to the nature of this project the only way of categorising the data is using experts for each respective field to manually categorise the data.

6.4.1 Shaolin Kung Fu dataset

All the forms measured using the MMR sensors were also recorded. For the case of the Shaolin Kung Fu dataset, experts from the Shaolin Kung Fu school went through recordings of their students doing Wu Bu Quan indicating which forms were successful. The successful forms were then further analysed going through them frame by frame and labelling the exact movement each stance starts and ends. The labelling process is very precise where minor difference makes a difference, an example of this can be seen in Figure 39. To help categorise the 350+ recorded forms, the head of the school taught a select few high-level practitioners to identify when the stances began and ended. From the videos not categorised by the experts, 50 were randomly selected and assessed by the head of the school; all 50 were correctly labelled.

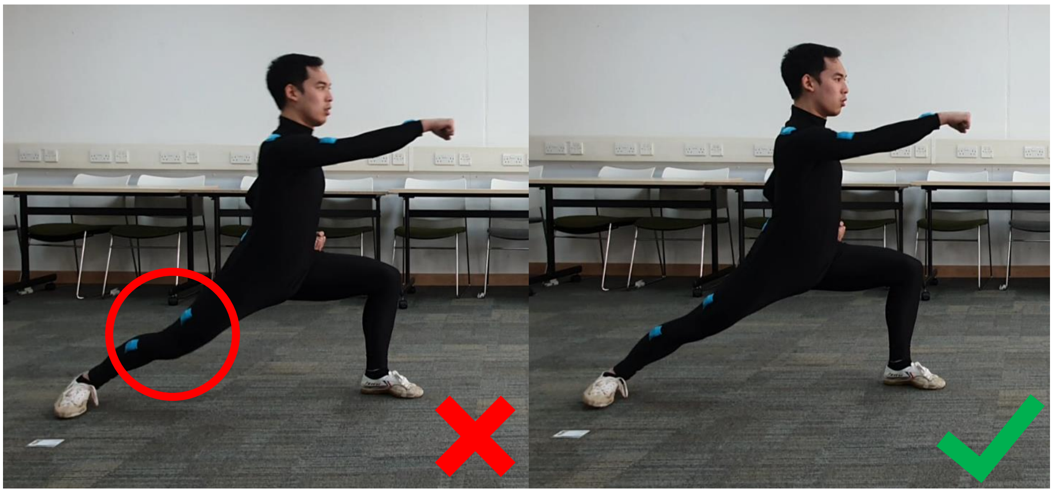


Figure 39 Practitioner performing the Gong Bu stance. On the left the stance is not satisfactory due to the slight bend in the right leg. The stance on the right is labelled a successful Gong Bu.

6.4.2 Modified UPDRS datasets

A researcher in neuropsychological assessment, Sara Pisani, who specialised in using the MDS-UPDRS assessment procedure has given a tutorial on how to emulate the 4 different level of severity classified by the assessment procedure.

The dataset was then emulated by the author and labelled by Sara Pisani and validated by her colleagues. The emulation focussed on getting a variety of cases for each severity description given in Figure 7. In total the dataset compromised on 100 examples for each severity level, with the sensor being replaced on the finger/wrist at slightly different positions to account for human subjectivity.

6.5 Post processing

Before applying the measured data for both movement sets some post processing is required. Due to most of the processing being done within the feature extraction only three steps are required at this point.

Step 1: Transfer the measured data from the Metabase application to a personal cloud storage, using CSV formatting.

Step 2: Synchronise the data using MATLAB's xcorr method which can be used to synchronise two signals by determining the lag which results in the greatest correlation. This is only required for the Shaolin Kung Fu dataset as it uses multiple sensors.

Step 3: Shaolin Kung Fu: import labelled timestamps into a response variable matrix. Modified UPDRS taps: import severity labels into a response variable matrix. These matrices will be used as the ground truth for the models presented in the next chapter.

6.6 FEATURES

This section will describe the features used and the process of extracting these features from the processed measurements. However, before going into the details of the features used a summary will be given for ease of reference.

For the Shaolin Kung Fu dataset, a piecewise joint-based feature set has been extracted from the processed data. This feature set is made up of the mean of windowed and quantised orientation data as well as the standard deviation of windowed and quantised angular velocity data. Therefore, considering that each dimension of the 4 dimensions for the orientation data (quaternion representation) and 3 dimensions for the angular velocity data is an individual feature, and that there are 17 sensor locations, the total number of features is 119. However, during the initial evaluations of the model it became clear the sensors located on the forearms, upper arms, and head were irrelevant to the performance of the model and should be excluded from the study. Therefore, excluding the features associated to these locations, a total of 70 features were evaluated.

For the modified UPDRS dataset, the following 8 features were chosen: duration, number of peaks, average peak height, number of detected taps, standard deviation of tap height, standard deviation of tap gap, standard deviation of points to plane, and index of Durbin Watson analysis.

6.6.1 Shaolin Kung Fu dataset

6.6.1.1 Baseline models

In this section will brief outline the work done on full body motion capture, focussing on skeletal feature extraction.

Existing skeleton-based human action recognition approaches can be broadly grouped into two main categories: joint-based approaches and body part-based approaches. Joint-based approaches consider human skeleton as a set of points, whereas body part-based approaches consider human skeleton as a connected set of rigid segments. Approaches that use joint angles can be classified as part-based approaches since joint angles measure the geometry between (directly) connected pairs of body parts.

Joint-based approaches:

Human skeletons were represented in [77] using the 3D joint locations. To encode the relationship between joint movement and time, they deploy multiple covariance matrices over sub-sequences in a hierarchical fashion. The descriptor has a fixed length that is independent from the length of the described sequence, making it a viable method for the Kung Fu dataset as the form and transition times vary from subject to subject. A similar method was used by [78], this time using a Hidden Markov Model (HMM) to learn the features of each dynamic action class separately. The observation probability is computed in each HMM and a weak classifier for that feature is formed based on those probabilities. Splitting the dataset into separate action classes allows the system to be scalable which will be necessary when new forms/other applications are considered.

A set of 13 joint trajectories in a 4D XYZT space was used in [79] to represent a human action, and their affine projections were compared using a subspace angles-based view-invariant similarity measure. They identify three important sources of variability: (1) viewpoint, (2) execution rate, and

(3) anthropometry of the subject. In the case of the Kung Fu dataset, viewpoint should be considered as the difference between the local and global axes.

In [80], a human skeleton was represented using pairwise relative positions of the joints, and the temporal evolutions of this representation were modelled using a hierarchy of Fourier coefficients. Furthermore, an action let-based approach was used, in which discriminative joint combinations were selected using a multiple kernel learning approach. The feature detector proposed is robust to noise, invariant to translational and temporal misalignments, all useful aspects when considering the adapted spherical coordinates system used whilst processing the dataset.

In [81], a human skeleton was represented using relative joint positions, temporal displacement of joints and offset of the joints with respect to the initial frame. Action classification was performed using the Naive-Bayes nearest neighbour rule in a lower dimensional space constructed using principal component analysis (PCA). A similar skeletal representation was used with random forests in [82].

A view invariant representation of human skeleton was obtained in [83] by quantizing the 3D joint locations into histograms based on their orientations with respect to a coordinate system fixed at the hip centre. The temporal evolutions of this view-invariant representation were modelled using HMMs.

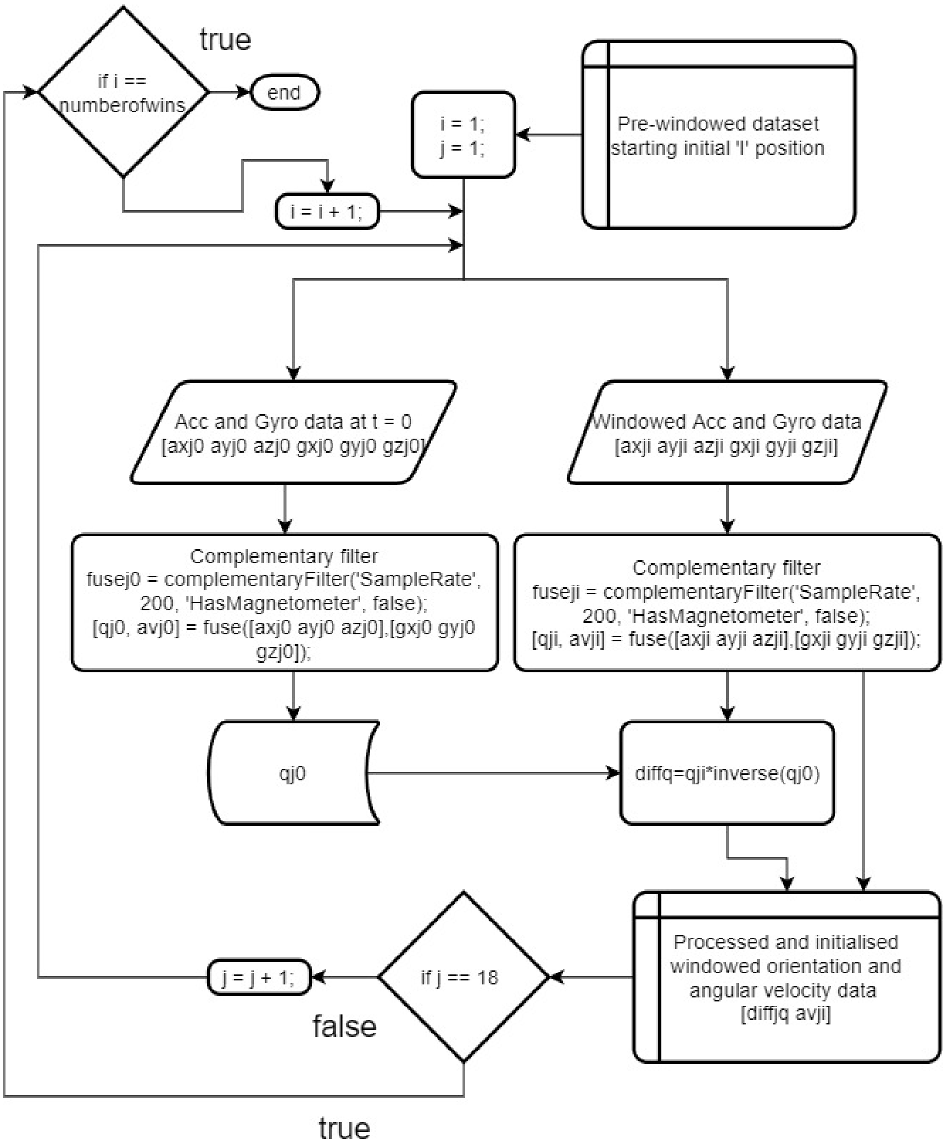
Part-based approaches:

The human body was divided into five different parts in [84], and actions were represented using the motion parameters of individual body parts like horizontal and vertical translations, in-plane rotations, etc. Principal component analysis was used to represent an action as a linear combination of a set of action basis, and classification was performed by comparing the PCA coefficients. In [85], a human skeleton was hierarchically divided into smaller parts and each part was represented using certain bio-inspired shape features. The temporal evolutions of these bio-inspired features were modelled using linear dynamical systems. Human skeleton was represented using 3D joint angles in [86], and the temporal evolutions of these angles were compared using dynamic time warping (DTW). In [87], few informative skeletal joints were automatically selected at each time in-stance based on highly interpretable measures such as mean or variance of the joint angles, maximum angular velocity of the joints, etc. Human actions were then represented as sequences of these informative joints, which were compared using the Levenshtein distance. Skeletal sequences were represented in [88] using pairwise affinities between joint angle trajectories, and then classified using linear SVM.

Note: Despite promising results regarding the use of PCA features in related work it has been decided that this project would not use PCA features. By using PCA features the affect each sensor's location has on the model becomes unknown therefore removing the potential of reducing the number of sensors required for full body analysis for the given movement set.

6.6.1.2 Features

As the stances are concerned with the body's orientation as well as dynamic to static transitions a complementary filter has been used to convert the accelerometer and gyroscope data into quaternions and angular velocity measurements. This process is outlined by the flowchart in Figure 40. The initial "I" position was used to determine the change in orientation from this initial starting point to the stance. This allowed the deterministic noise to be reset to zero reducing the influence of gyroscope drift as gravitational acceleration will have a majority influence on roll and pitch orientations during this static initialisation position. By considering difference in orientation rather than absolute orientation and the assumption that the main articulating joints are perfectly rigid, the system can account for different body types as well as the sensors being placed on different sections of the joint or in different orientations.



if i == numberofwins
true
end
i = 1;
j = 1;
Pre-windowed dataset starting initial 'I' position
i = i + 1;
Acc and Gyro data at t = 0
[axj0 ayj0 azj0 gxj0 gyj0 gzj0]
Windowed Acc and Gyro data
[axji ayji azji gxji gyji gzji]
Complementary filter
fusej0 = complementaryFilter('SampleRate', 200, 'HasMagnetometer', false);
[qj0, avj0] = fuse([axj0 ayj0 azj0],[gxj0 gyj0 gzj0]);
Complementary filter
fuseji = complementaryFilter('SampleRate', 200, 'HasMagnetometer', false);
[qji, avji] = fuse([axji ayji azji],[gxji gyji gzji]);
qj0
diffq=qji*inverse(qj0)
Processed and initialised windowed orientation and angular velocity data
[diffjq avji]
if j == 18
j = j + 1;
false
true

Figure 40 Accelerometer and gyroscope to quaternions and angular velocity flowchart

Once these difference orientations are processed the data can then have the joint limitations applied to determine if any of the measurements are outside the range of motion possible. To apply the joint limitations the difference between each joint and its parent joint is calculated (e.g., calf and thigh). This difference is then compared to the joint's limits outlined in Chapter 2.3. If the joint rotates beyond this limit, it is instead kept at the limit, giving a warning to the system, allowing the next rotation to occur on this updated orientation. If the joint continues to rotate beyond this limit, i.e., having sequential measurements resulting in warnings, the window will be given an error flag and be invalid for the dataset. The threshold chosen until the system stops giving warning and flags an error was 10, which when considering the sample rate of 200Hz would be 0.05 seconds of lost data. Due to the length of measurements between re-initialisation being less than 20 seconds, very few warnings were given, all of which were on the Yaw axis and none of which resulted in an error flag.

The limited data was then quantised using the logic described by the flowchart in Figure 41. It was decided to quantise the data following the description of the movements, which can be seen in Chapter 2. As described the stances each have a clear orientation associated to them and should be held statically for a split second. Going through the dataset, this split second was at its shortest 0.4 seconds for any given stance. Therefore, a quantising window of 40 samples (0.2 seconds) where each frame calculated the mean orientation and standard deviation of the angular velocity. This will reduce the 240 elements of data points per joint per frame to just 6 elements of data per joint per frame. The standard deviation of the angular velocity should be able to indicate if the measurements are static while the mean of the orientation will indicate the position the body was in during the quantised window.

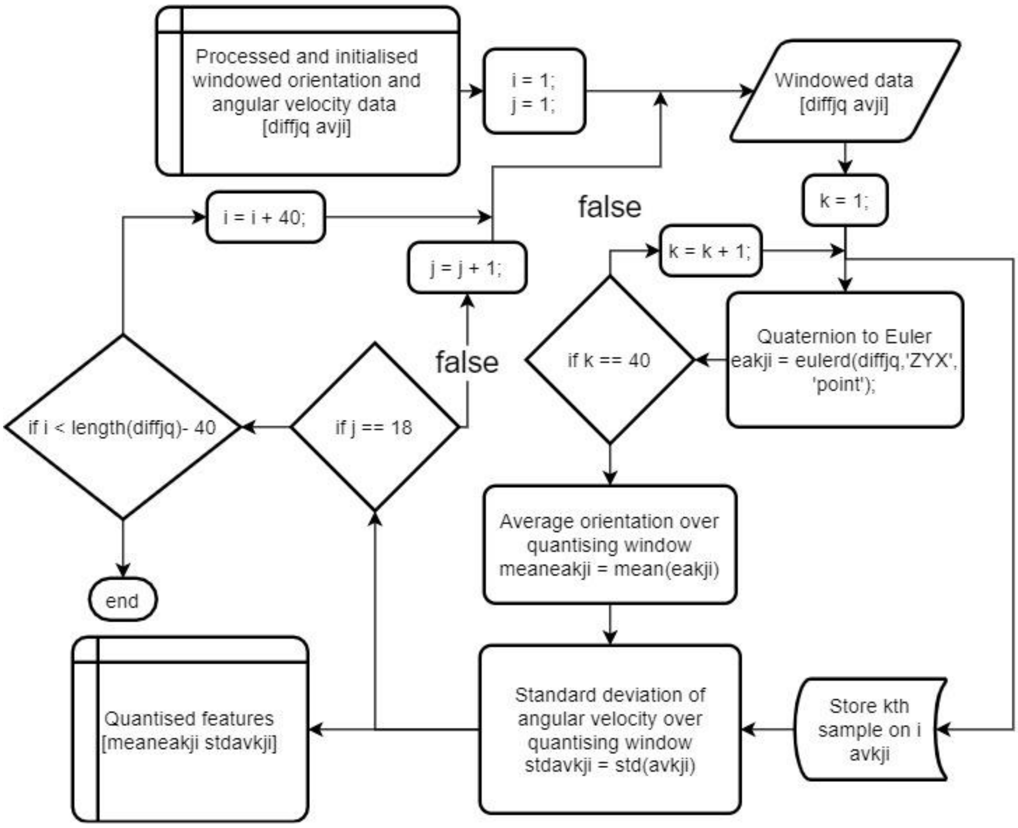

Processed and initialised windowed orientation and angular velocity data [diffjq avji]
i = 1;
j = 1;
Windowed data [diffjq avji]
k = 1;
i = i + 40;
false
j = j + 1;
k = k + 1;
Quaternion to Euler eakji = eulerd(diffjq,'ZYX', 'point');
if k == 40
false
if i < length(diffjq)- 40
if j == 18
Average orientation over quantising window meaneakji = mean(eakji)
end
Quantised features [meaneakji stdavkji]
Standard deviation of angular velocity over quantising window stdavkji = std(avkji)
Store kth sample on i avkji

Figure 41 Flowchart which describes the process used to quantise the Shaolin Kung Fu movement set

Before moving on to the machine learning model, the data was analysed to determine variance between the categorised measurements. This was done for all the relevant joints (i.e., excluding the arms and head) across the whole dataset. This process was done taking the range of only the categorised data from the output of Figure 40. It was also done manually on some samples to get a better understanding of the data. Figure 42 and Figure 43 are two separate recordings of Wu Bu Quan done by two practitioners; the points highlighted are within the areas previously categorised. Table 7 compares these points, giving the difference for each yaw, pitch and roll before and after considering the initial "I" position. Figure 44 and

Figure 45 were used for the results presented in Table 8 for the right calf orientations. It is clear from these results that the roll and pitch measurements have little difference across these two samples while the yaw measurement has much greater difference. Considering the practitioners did all their stances along a straight line to reduce variables in the system, this yaw range is put down to either less strict categorisation when considering yaw or inaccurate yaw calculations.

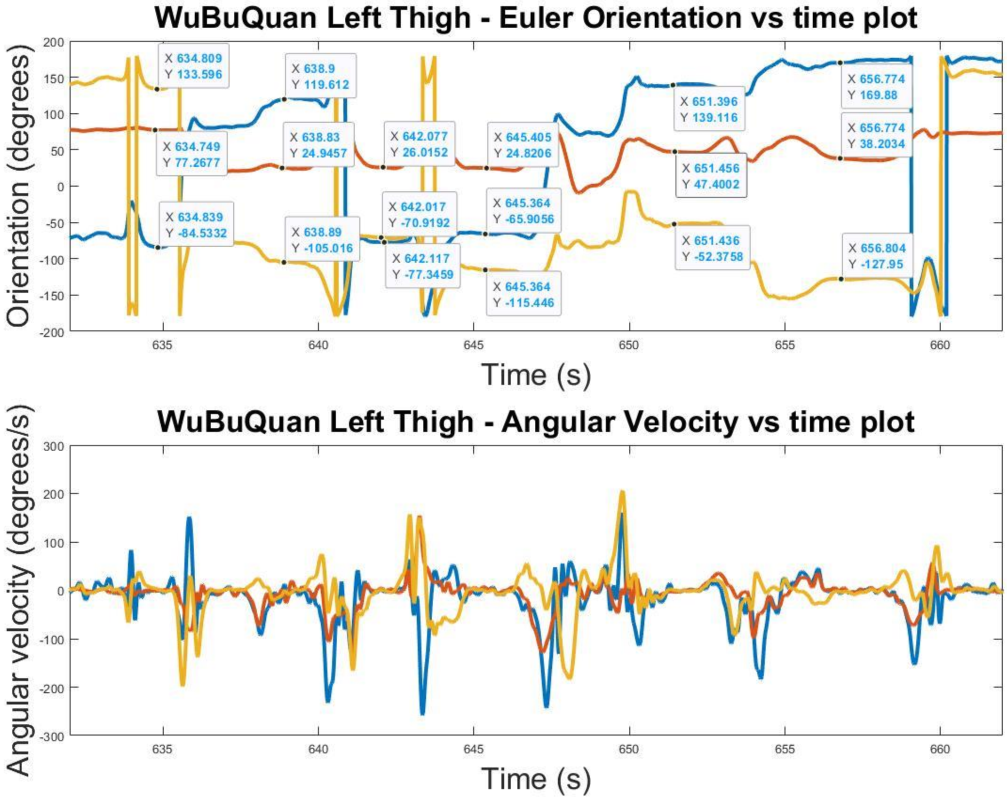


Figure 42 Euler orientation (degrees) measurements against time (seconds) and angular velocity (degrees/second) against time (seconds) for a windowed sample (sample set 1) of the left thigh indicating time stamps of each stance during the form Wu Bu Quan.

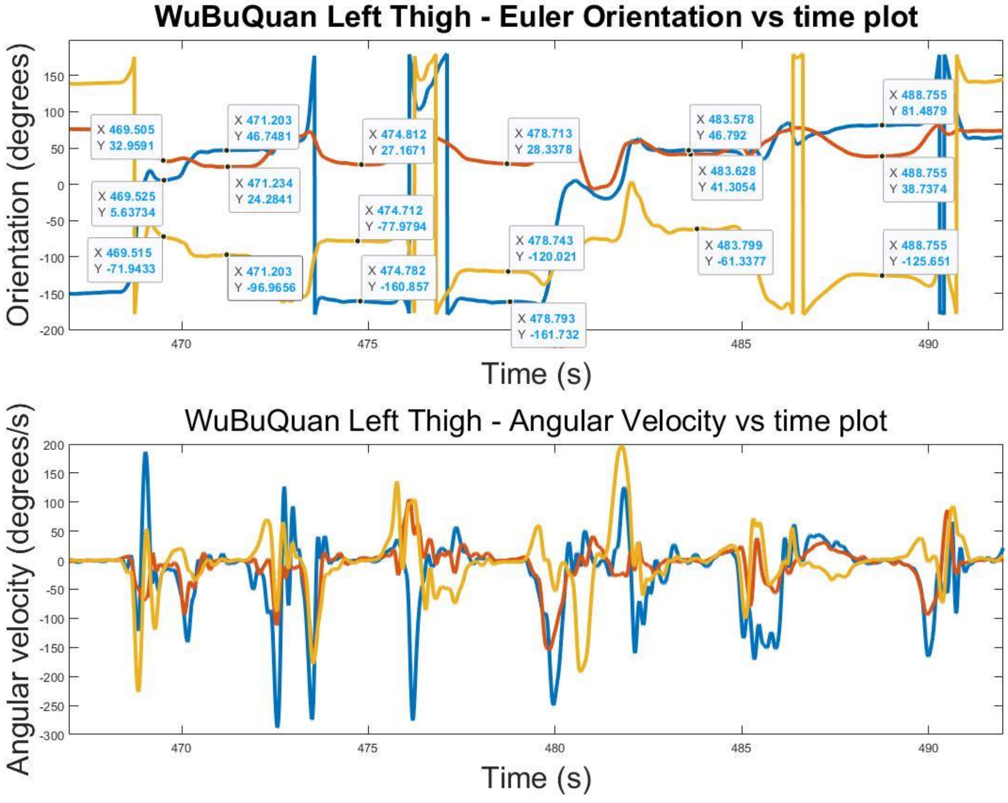


Figure 43 Euler orientation (degrees) measurements against time (seconds) and angular velocity (degrees/second) against time (seconds) for a windowed sample (sample set 2) of the left thigh indicating time stamps of each stance during the form Wu Bu Quan.

	Left thigh window 1 (degrees)	**Left thigh window 2 (degrees)**	**Absolute difference (degrees)**	**Absolute difference including initial offset (degrees)**
Gong Bu				
Yaw	46.8	119.6	72.8	7.4
Pitch	24.3	24.9	0.6	1.1
Roll	-97.0	-105.0	8.0	2.5
Ma Bu				
Yaw	-160.9	-77.3	83.6	18.2
Pitch	27.2	26.0	1.2	0.5
Roll	-78.0	-70.9	7.1	1.6
Xie Bu				
Yaw	-161.7	-65.9	95.8	30.4
Pitch	28.3	24.8	3.5	1.8
Roll	-120.0	-115.4	4.4	1.1
Pu Bu				
Yaw	46.8	139.1	92.3	26.9
Pitch	41.3	47.4	6.1	4.4
Roll	-61.3	-52.4	8.9	3.4
Xu Bu				
Yaw	81.5	169.9	88.4	23
Pitch	38.7	38.2	0.5	1.2
Roll	-125.7	-128.0	2.3	3.2

Table 7 Yaw, pitch and roll comparisons of two randomly selected sample sets of the left thigh; indicating the difference between the sets as well as the difference including initial offset compensations.

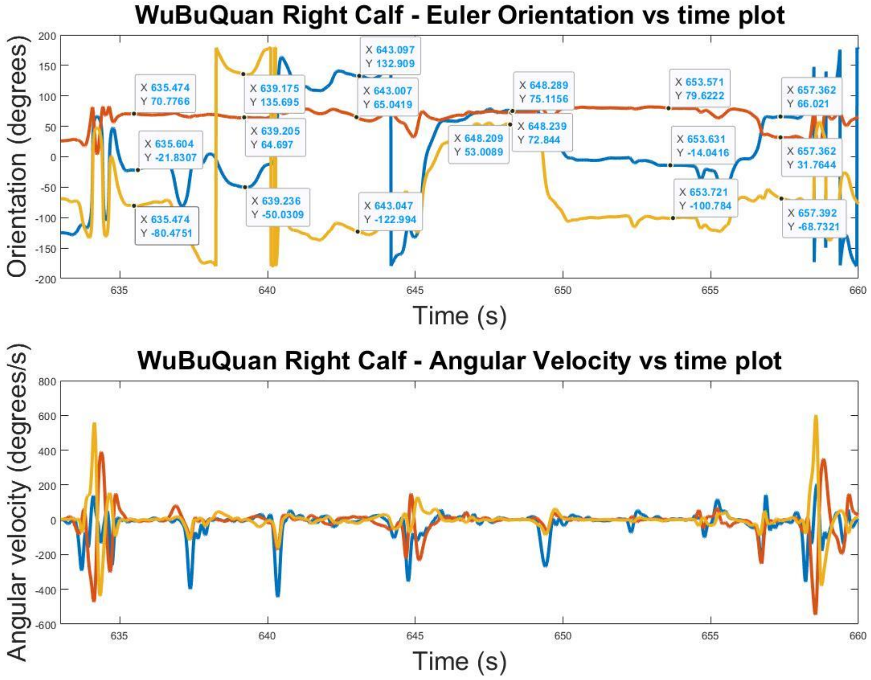


Figure 44 Euler orientation (degrees) measurements against time (seconds) and angular velocity (degrees/second) against time (seconds) for a windowed sample (sample set 1) of the right calf indicating time stamps of each stance during the form Wu Bu Quan.

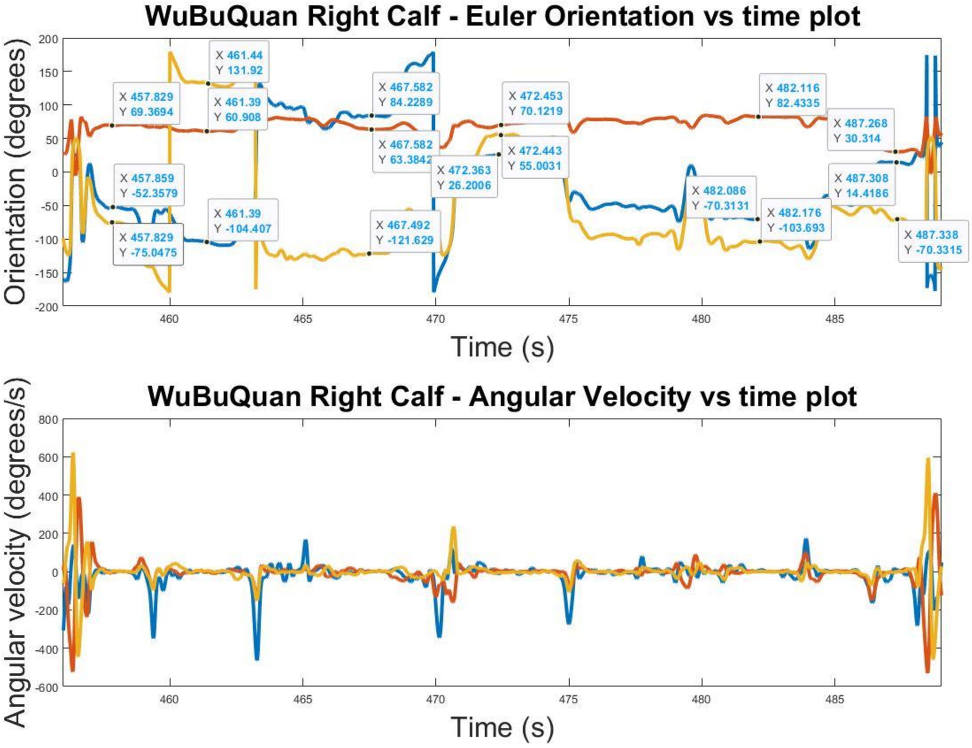


Figure 45 Euler orientation (degrees) measurements against time (seconds) and angular velocity (degrees/second) against time (seconds) for a windowed sample (sample set 2) of the right calf indicating time stamps of each stance during the form Wu Bu Quan.

	Right calf window 1 (degrees)	**Right calf window 2 (degrees)**	**Absolute difference (degrees)**	**Absolute difference including initial offset (degrees)**
Gong Bu				
Yaw	-50.0	-104.4	54.4	23.8
Pitch	64.7	60.9	3.8	2.4
Roll	135.7	131.9	3.8	1.7
Ma Bu				
Yaw	132.9	84.2	48.7	18.1
Pitch	65.0	63.4	1.6	0.2
Roll	-123.0	-121.6	1.4	4.1
Xie Bu				
Yaw	75.1	26.2	48.9	18.3
Pitch	72.8	70.1	2.7	1.3
Roll	53.0	55.0	2	3.5
Pu Bu				
Yaw	-14.0	-70.3	56.3	25.7
Pitch	79.6	82.4	2.8	1.4
Roll	-100.8	-103.7	2.9	2.6
Xu Bu				
Yaw	66.0	14.4	51.6	21
Pitch	31.8	30.3	1.5	0.1
Roll	-68.7	-70.3	1.6	3.9

Table 8 Yaw, pitch and roll comparisons of two randomly selected sample sets of the right calf; indicating the difference between the sets as well as the difference including initial offset compensations.

As mentioned before, this comparison was done with all the relevant joints using only the data selected in the categorising process. Table 9 shows the results of this process. The yaw results clearly have a larger range than the pitch/roll results, as mentioned with the manual assessment this either indicates that the system has low accuracy yaw measurements or that the categorisation procedure for yaw is less strict.

In fact, it may be both factors limiting the system when looking at the procedure used to categorise the data and the yaw data itself. Figure 42 has 4 seconds from second 650 to second 654 where the practitioner remains mostly static. During this time, pitch has a range of 1 degree, roll a range of 3 degrees, while yaw has a range of 16 degrees. This would suggest that the system has not fully removed the gyroscope drift when considering the yaw axis. However, another consideration to be made is that the stances were categorised by video, which has a single side on viewing point which would mean changes in yaw would be difficult to identify.

Joint Class	Range in degrees (Yaw)(Pitch)(Roll)	Joint Class	Range in degrees (Yaw)(Pitch)(Roll)
LC Gong Bu	(29.4)(4.4)(4.8)	RC Gong Bu	(21.1)(4.7)(2.4)
LC Ma Bu	(27.7)(3.8)(3.8)	RC Ma Bu	(28.6)(3.0)(5.4)
LC Xie Bu	(23.3)(4.0)(3.8)	RC Xie Bu	(23.7)(2.8)(4.0)
LC Pu Bu	(30.1)(4.2)(3.9)	RC Pu Bu	(26.6)(2.9)(2.9)
LC Xu Bu	(27.3)(4.6)(4.9)	RC Xu Bu	(26.2)(1.8)(5.9)
LF Gong Bu	(35.5)(4.3)(2.7)	RF Gong Bu	(31.3)(2.0)(1.8)

LF Ma Bu	(28.1)(4.3)(2.1)	RF Ma Bu	(26.1)(2.3)(1.9)
LF Xie Bu	(24.4)(2.9)(2.2)	RF Xie Bu	(28.9)(4.6)(2.2)
LF Pu Bu	(22.1)(5.8)(2.1)	RF Pu Bu	(31.2)(4.1)(3.5)
LF Xu Bu	(24.3)(3.7)(2.1)	RF Xu Bu	(25.5)(3.8)(5.1)
LT Gong Bu	(24.1)(3.4)(2.5)	RT Gong Bu	(23.3)(3.0)(2.2)
LT Ma Bu	(26.6)(2.8)(5.1)	RT Ma Bu	(31.1)(3.0)(3.6)
LT Xie Bu	(22.4)(3.3)(4.3)	RT Xie Bu	(32.0)(5.6)(3.6)
LT Pu Bu	(28.8)(3.2)(3.0)	RT Pu Bu	(36.4)(4.5)(2.8)
LT Xu Bu	(23.9)(2.0)(4.5)	RT Xu Bu	(30.0)(3.6)(4.0)
LB Gong Bu	(26.6)(2.5)(4.8)	UB Gong Bu	(22.2)(3.1)(3.4)
LB Ma Bu	(26.9)(2.4)(5.1)	UB Ma Bu	(26.9)(2.6)(5.1)
LB Xie Bu	(23.8)(2.8)(4.1)	UB Xie Bu	(24.4)(3.3)(5.1)
LB Pu Bu	(24.5)(2.9)(9.8)	UB Pu Bu	(23.8)(3.0)(11.1)
LB Xu Bu	(24.5)(2.4)(6.7)	UB Xu Bu	(25.0)(2.8)(7.0)

Table 9 Range of differences between the sets in degrees. LC = left calf; LF = left foot; LT = left thigh; LB = lower back; RC = right calf; RF = right foot; RT= right thigh; UB = upper back.

6.6.2 Modified UPDRS datasets

6.6.2.1 Baseline models

This section will focus on possible features that could be used to classify the Parkinson's dataset. This will be done by taking the descriptions given in Figure 7 and translating them into quantifiable features.

The descriptions given in Figure 7 can be broken into 3 groups: speed/duration, amplitude, and changes throughout the measurement (change in amplitude, change in frequency, and hesitations/halts). The following feature descriptions will be made with the assumption the data has been windowed only to include the exercise.

Standard statistical features will be able to represent the first two groups, such features have previously been used in similar projects that use a single sensor location [28], [30] and [31]. Analysing changes within a measurement will require more advanced techniques.

A signal that is consistent throughout can be considered to have a high autocorrelation. Autocorrelation is a characteristic of data which shows the degree of similarity between the values of the same variables over successive time intervals.

The description of the movement set, seen in chapter 2, suggests that patients with low severity PD would have a higher autocorrelation measurement than patients with high severity. Therefore, it can be assumed that the modified UPDRS score should be inversely proportional to the autocorrelation measurement.

There are many ways to measure autocorrelation and many more to score the measurement. This thesis proposes that the Durbin Watson analysis as a feature would be able to distinguish the different levels of severity. The Durbin Watson analysis has yet to be used in motion capture data however has previously been used to detect temporal autocorrelation in Vehicular Crash data [89].

The Durbin Watson analysis uses the following calculation:

$$d = \frac{\sum_{i=2}^{n}(e_i - e_{i-1})^2}{\sum_{i=1}^{n} {e_i}^2}$$

Where $e_i = y_i - \hat{y_i}$ are the residuals (with y_i being the observed value and $\hat{y_i}$ being the predicted value) and n the number of elements in the sample.

The resulting figure of the Durbin Watson analysis signifies the possibility that the sample has autocorrelation. The closer the result is to zero the higher the probability that the sample does not have autocorrelation. The units of the analysis depend on the residuals used, however as the analysis is mainly used for comparisons stake, the units are arbitrary.

6.6.2.2 Features

The following 8 features have been implemented using the methods described in the subsequent subsections. In this section the resulting features will be discussed, highlighting which severity levels are isolated by the feature and whether a feature would be enhanced when combined with another feature. Note that of the 100 sets completed for each level of severity, there was some unvalidated samples resulting in 97 level 0, 99 level 1, 99 level 2, 99 level 3 and 99 level 4. Additionally note that all the tests discussed in this section have been done on both accelerometer and gyroscope data separately, however as the accuracy and selected features were consistent across both sensors only the accelerometer results have been highlighted.

Duration

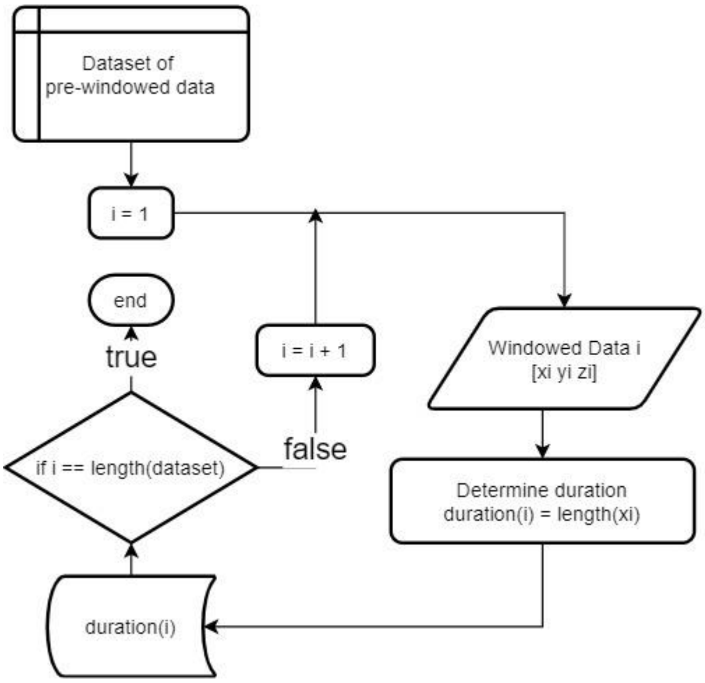


Figure 46 Flowchart of the logic used to establish the movement set's duration

This feature is simply the number of samples each test consists of, meaning that both the wrist and the finger sensor should be close to identical if the cropping process has been done successfully.

Figure 47 shows a clear correlation between the levels and the duration of the exercise. This is particularly so in levels 0 and 1, however there are some sets when levels 2, 3 and 4 behave against this correlation. This feature is good at identifying the lower levels.

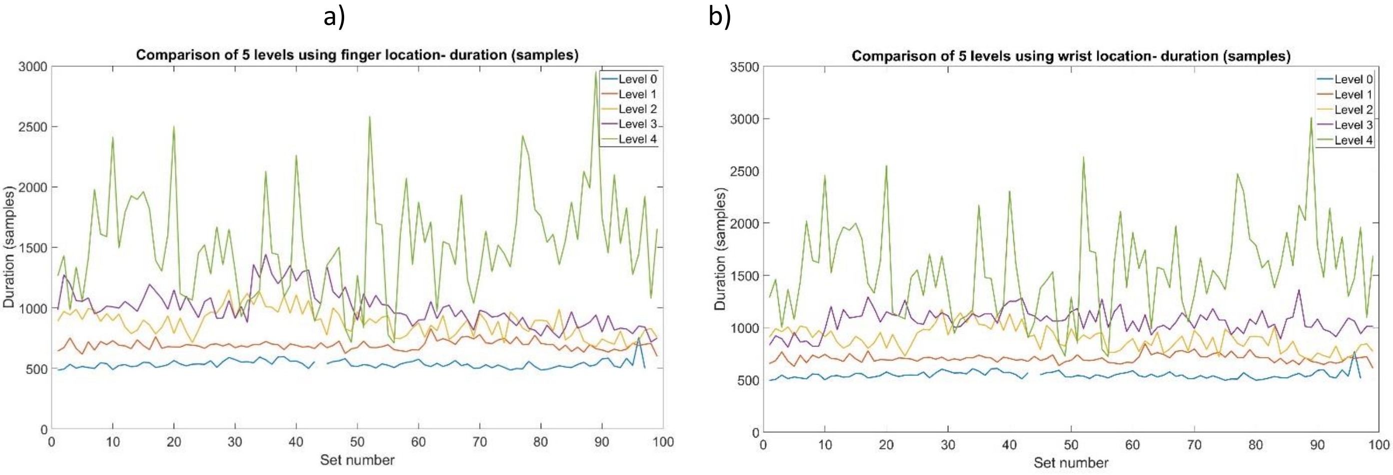


Figure 47 The duration feature as a number of samples for each set of severity.

Number of peaks

This feature set counts every peak in the data without any processing, with the logic that the noise is independent of the movement recorded thus consisted throughout the levels. Therefore, the difference in peaks is dominantly because of the movement during the exercise. There is a correlation between the number of peaks and the levels however there is a lot of cross over. Considering the finger sensor levels 0 and 1 predominantly follow this correlation whereas level 1 in the wrist dataset is less distinguishable from the upper levels. Overall, the number of peaks is less in the wrist dataset leading to the conclusion that some aspects of the movements made during the exercise were loss on this sensor.

This feature would pair well with the duration feature as the number of peaks has a strong dependency on the duration. By pairing the two features it would be possible to determine if the number of peaks are high because of the movement or because the exercise took longer.

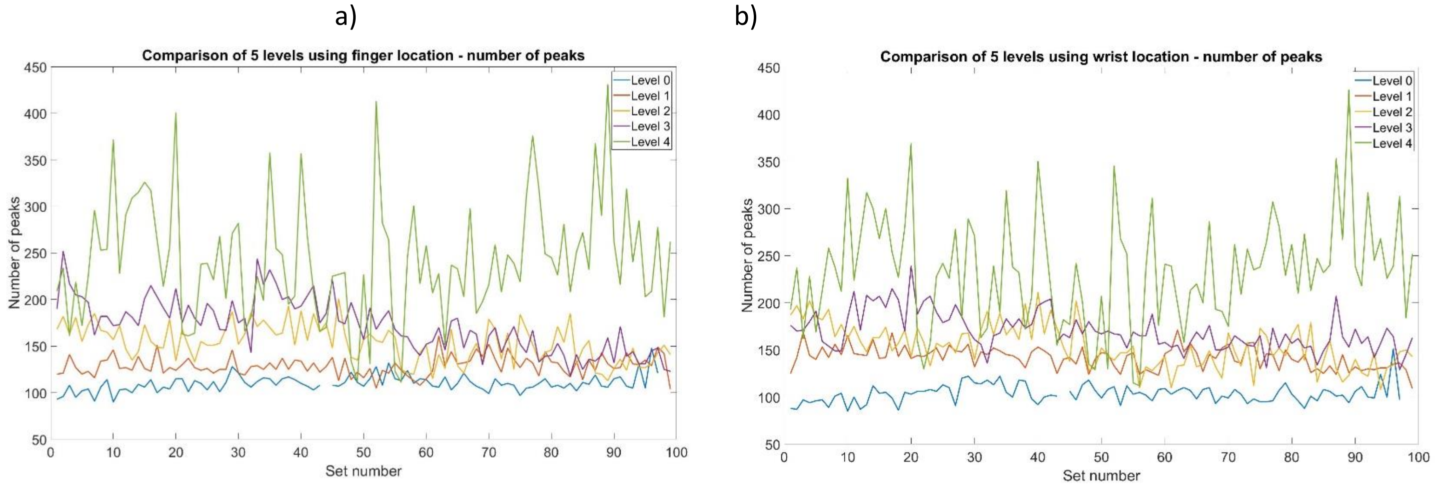


Figure 48 Number of peaks feature plotted against movement set for each severity level.

Average peak height

This feature gives a general approximation of the amplitude of the movements done during each exercise. The assumption made is that as the severity increases the amplitude decreases, however it turns out the hesitations in the exercise can also result in high accelerations, this is particularly visible in level 2 severity. Looking at the finger location, Figure 49a), the levels 0 and 1 have a slight distinction from the other levels however levels 2, 3 and 4 completely overlap. This indicates that with the lower accelerations the signal to noise ratio becomes too high to distinguish between the levels. This is even more apparent where the acceleration is much lower on the wrist location; here no levels are distinct. By coupling this feature with the points to plane features it would be possible to determine which peaks are relevant and which can be considered noise.

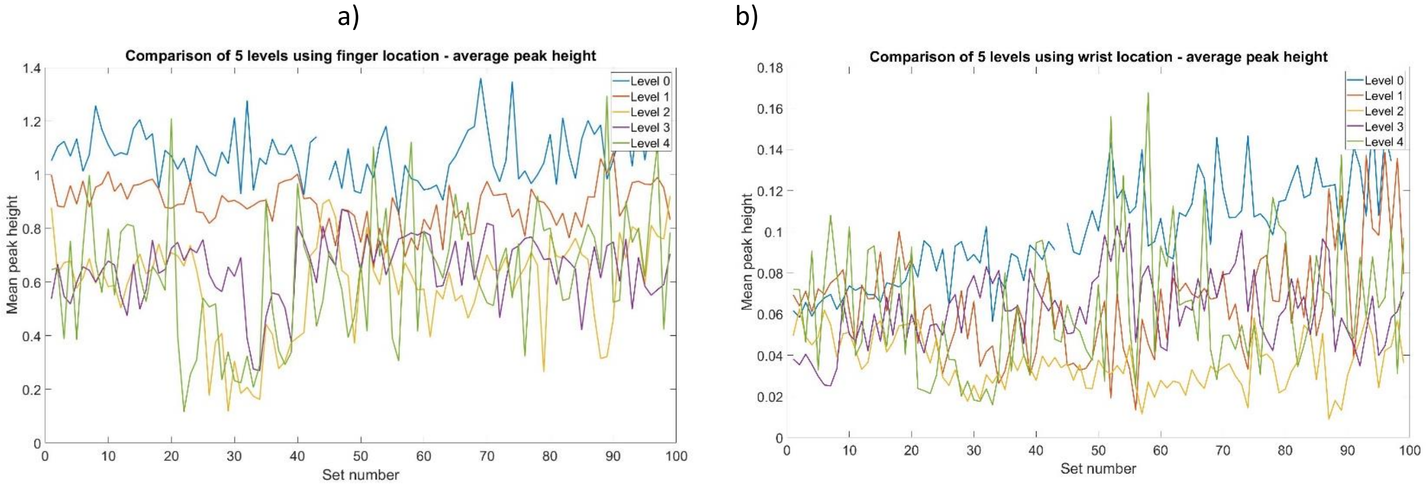


Figure 49 Average peak height feature plotted against movment set for each severity level. The units are arbitrary for this plot.

Number of detected taps

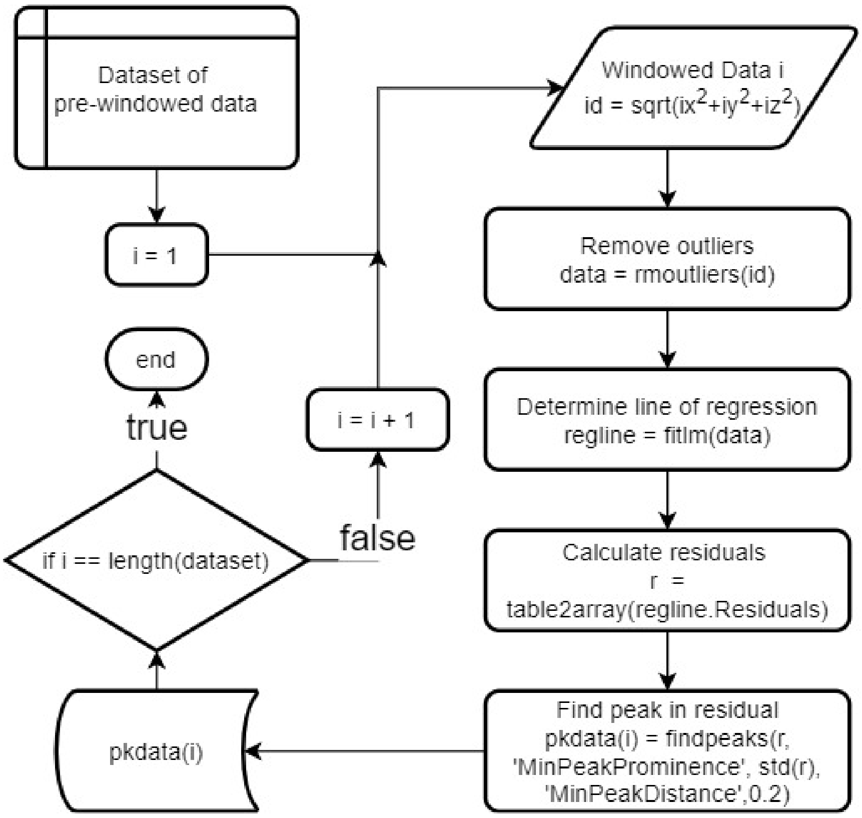


Figure 50 Logic flowchart for the feature extraction of the number of taps

This feature has been designed to distinguish the 4th level of severity from the rest, as this level should be the only one not achieving 10 taps. However, it is clear from the Figure 51 that the process of detecting taps is flawed as levels 3 and 4 mostly have more than 10 taps. This is because at these levels the tremor becomes dominant making its undisguisable from the tap accelerations resulting in the system counting tremors as taps as well. Even though this was not the intention of the feature it is able to distinguish the higher levels because they have many more detected taps.

This is less evident in the wrist position as the tremors have a higher impact on the feature resulting in levels 2, 3 and 4 having higher tap counts.

There is still some distinction between these levels meaning that when paired with another feature it may be possible to further this distinction. A viable pairing would be duration as the number of tremors would be proportional to the duration of the exercise.

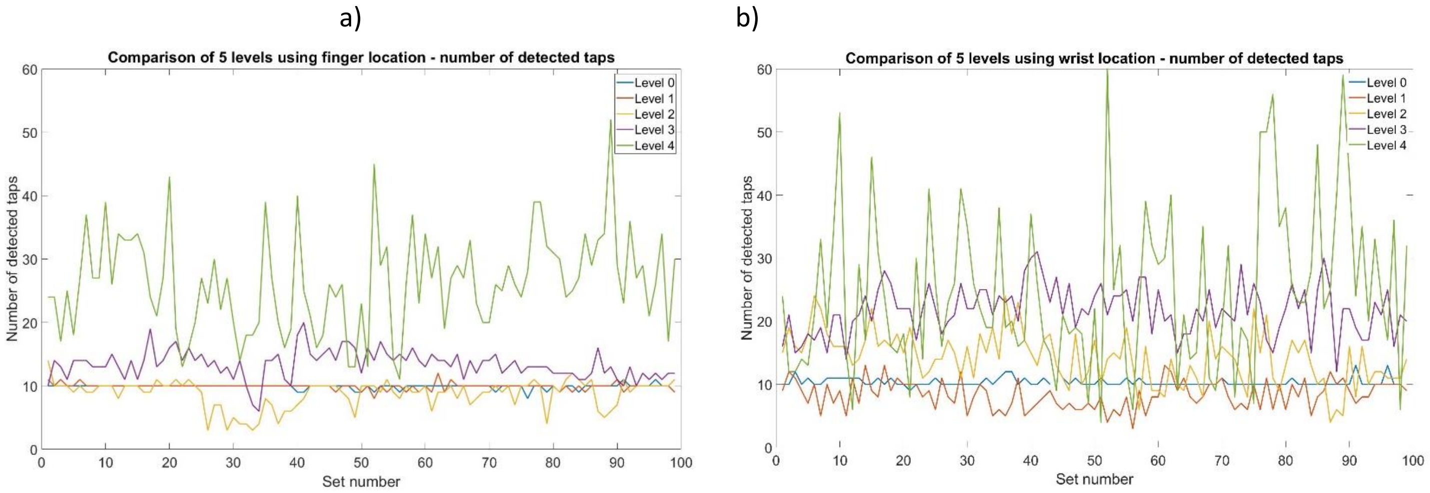


Figure 51 Number of detected taps feature plotted against movment set for each severity level. The units are arbitrary for this plot.

Standard deviation of tap height

This feature takes the detected taps feature and applies a standard deviation to determine how much the amplitude of the taps vary. However as previously discussed the tap feature also counts the tremors meaning this feature will be determining the standard deviation of the tremors as well as the taps. As can be seen in the figures below for both the finger location and the wrist location this information does not distinguish any severity level as there is just too many variables to consider as an isolated feature. Assuming the tremors are proportional to the duration of the exercise this feature would benefit in being coupled with the duration feature.

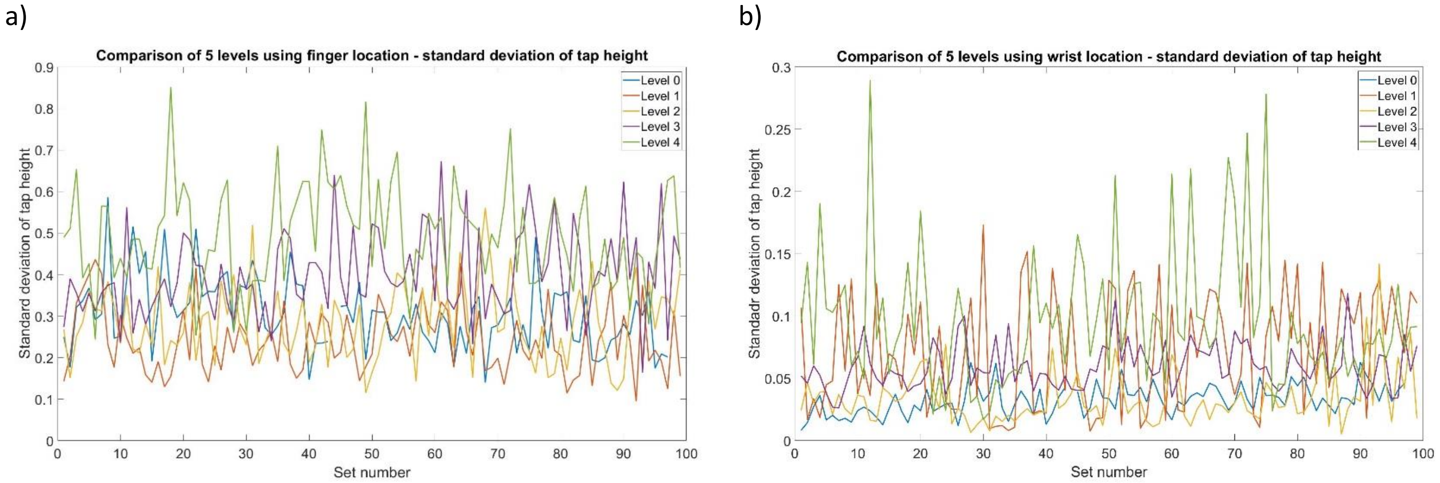


Figure 52 Standard deviation of tap height feature plotted against movment set for each severity level. The units are arbitrary for this plot.

Standard deviation of tap gap

This feature also uses the detected taps feature and applies a standard deviation of the time between detected taps. However, once again this feature would also be including the tremors that have been passed off as taps in the detected tap feature.

This feature will have a strong negative correlation with the duration feature because as the exercise goes on the tremor to tap ratio would increase reducing the deviation between the tremors/taps. This is assuming that the tremors are near constant throughout the exercise, otherwise there should still be a negative correlation but a weaker tone.

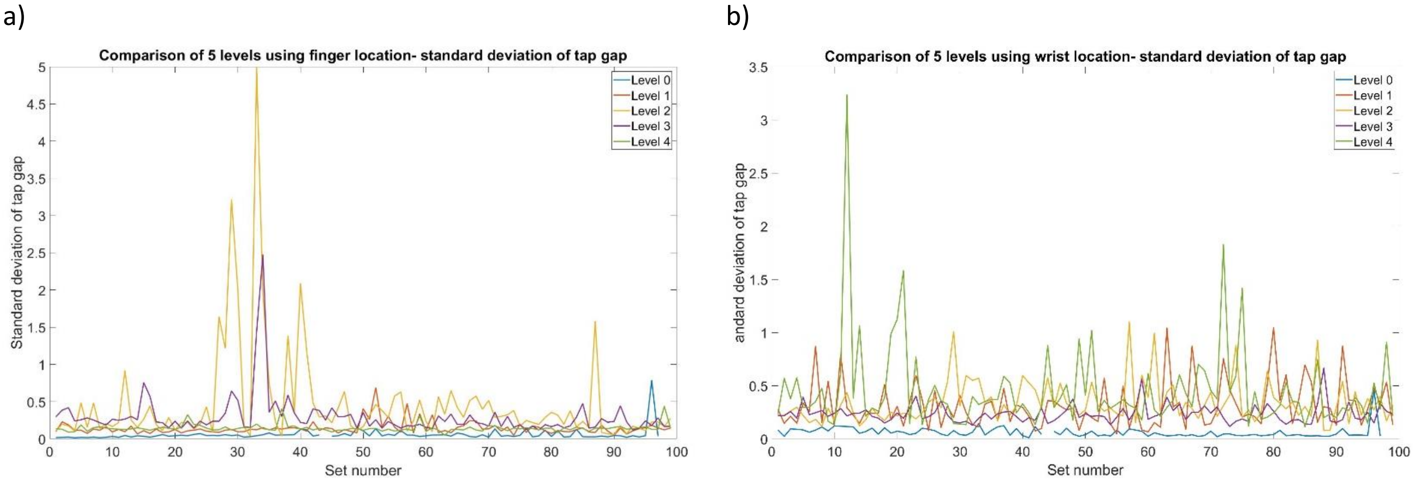
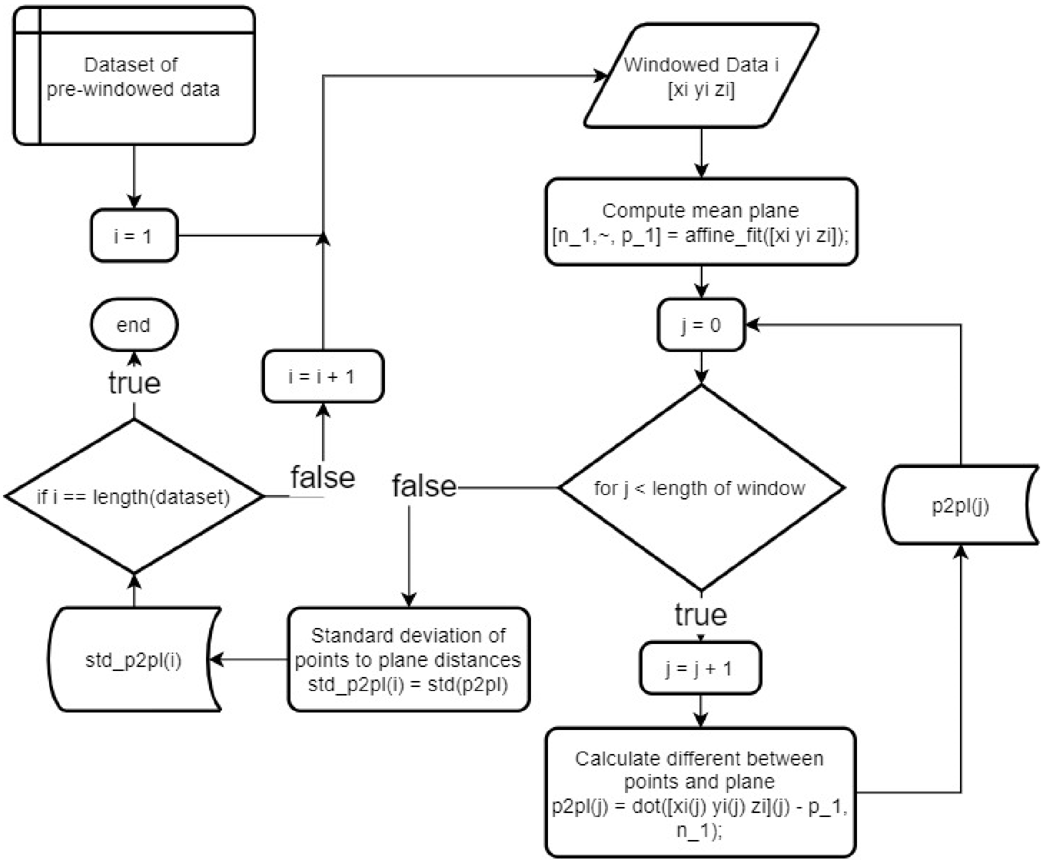


Figure 53 Standard deviation of tap gap feature plotted against movment set for each severity level. The units are arbitrary for this plot.

Standard deviation of points to plane analysis



Dataset of pre-windowed data
i = 1
end
true
i = i + 1
if i == length(dataset)
false
std_p2pl(i)
Standard deviation of points to plane distances std_p2pl(i) = std(p2pl)
false
Windowed Data i [xi yi zi]
Compute mean plane [n_1,~, p_1] = affine_fit([xi yi zi]);
j = 0
for j < length of window
true
j = j + 1
Calculate different between points and plane p2pl(j) = dot([xi(j) yi(j) zi](j) - p_1, n_1);
p2pl(j)

Figure 54 Flowchart describing the process to determine the standard deviation of the points to plane feature.

This feature has been designed to distinguish movement off the desire downward tap direction. Such movement would be the larger tremors seen in levels 3 and 4. Looking at Figure 55 this feature is mostly able to detect these larger tremors, seen by the higher variation off the desired plane of movement in levels 3 and 4. However considering not all patients experience these large tremors this feature alone cannot be relied on to differentiate the higher levels from the lower.

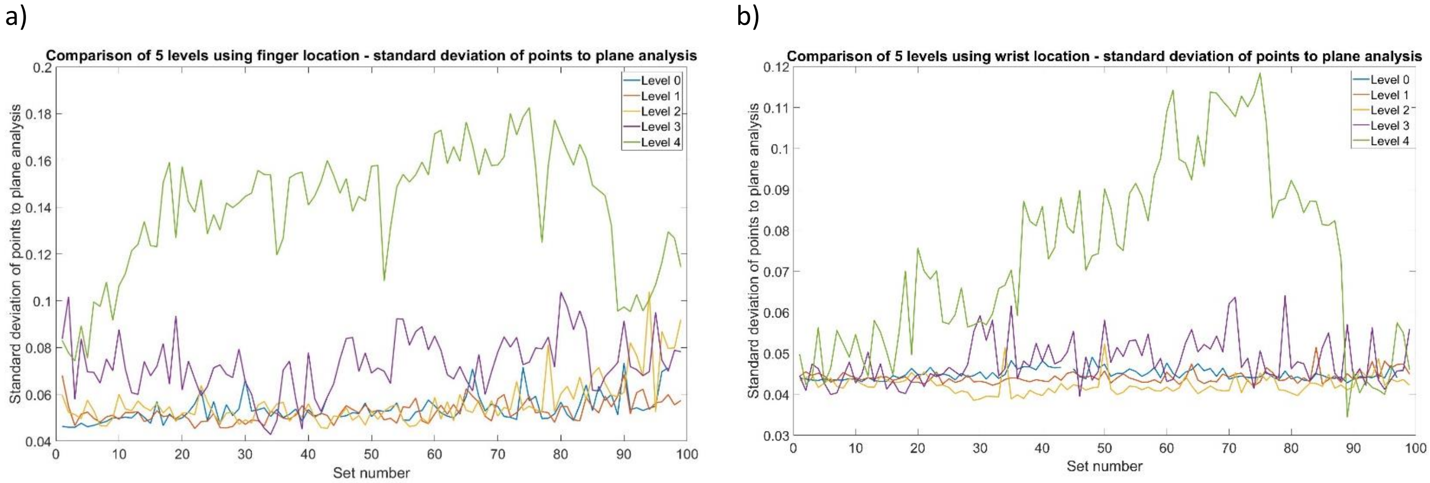


Figure 55 Standard deviation pf points to plane analysis feature plotted against movment set for each severity level. The units are arbitrary for this plot.

Durbin Watson

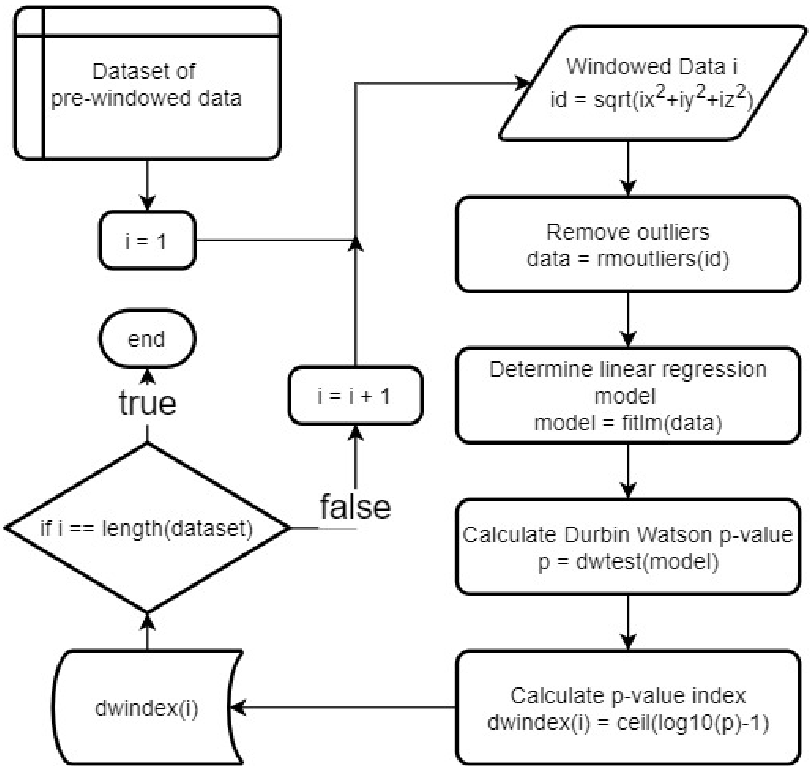
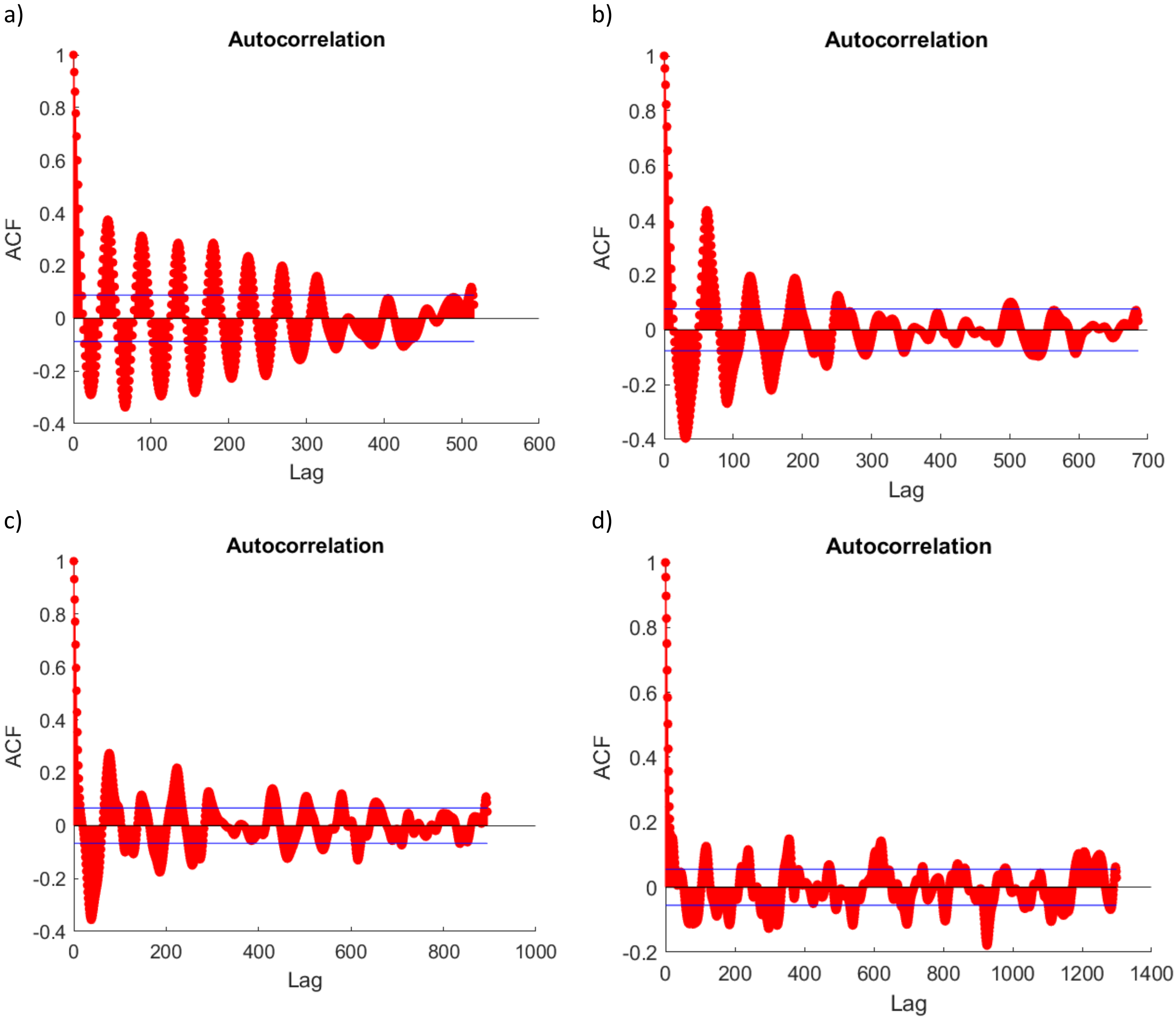


Figure 56 Flowchart describing the process to determine the Durbin Watson analysis index using Matlab methods.

Although this does not use the autocorrelation plot, they are useful in understanding the score given, hence Figure 57. This feature has taken the resulting Durbin Watson score for each dataset and removed its coefficient leaving only the index value. Note that level 4 has been given a null value meaning it below the threshold decided to indicate any level of autocorrelation. These null values have been converted to 0 for the machine learning model to handle. Figure 58 shows the index value of the Durbin Watson analysis plotted against the sample (set number). From the figure it is clear the Durbin Watson analysis is able to distinguish the higher severity levels for both locations, however struggles to make any separation in the lower severity levels.



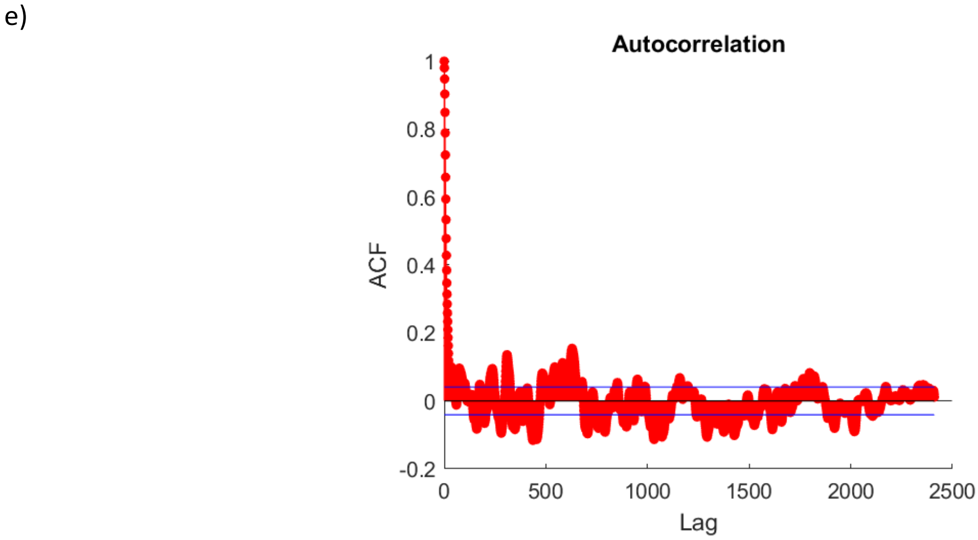


Figure 57 Autocorrelation plots of each severity level of a random sample set. a) level 0; b) level 1; c) level 2; d) level 3; e) level 4. The units are arbitrary for this plot.

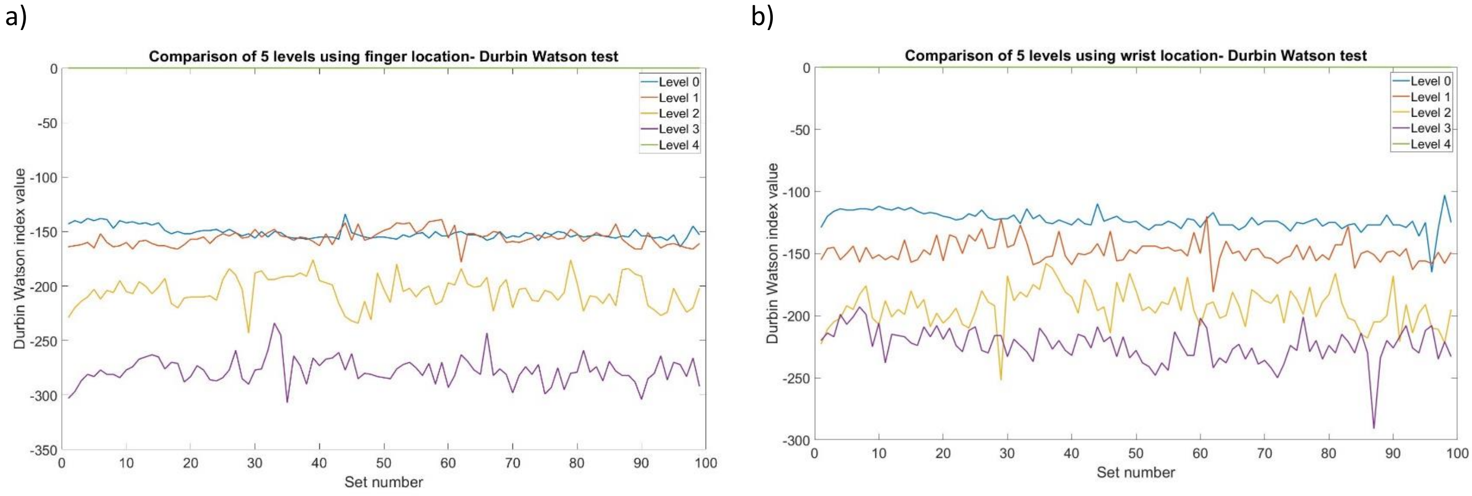


Figure 58 Durbin Watson test index feature plotted against movement set for each severity level. The units are arbitrary for this plot.

7 EXPERIMENTS AND EVALUATION

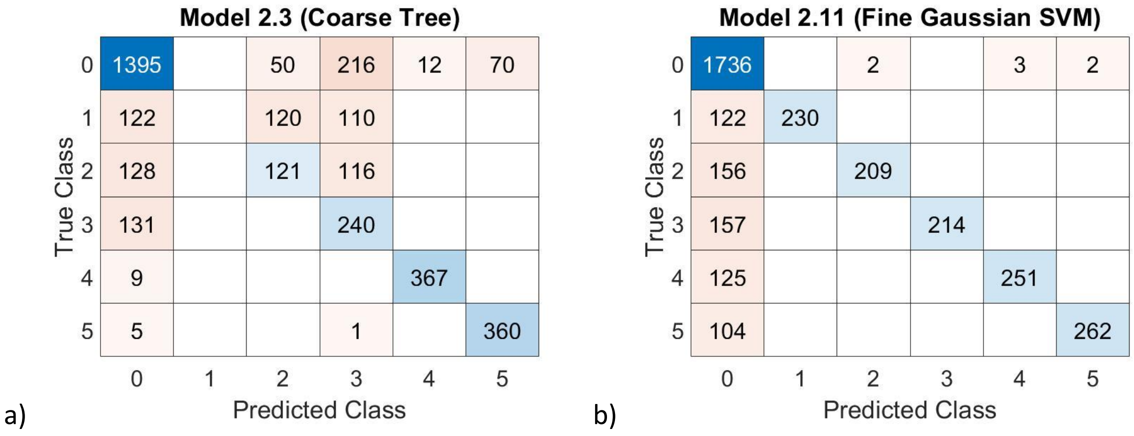
In this chapter the project's results will be displayed, allowing direct comparisons between the different machine learning models used. Each model's confusion matrix will be discussed to gain a better understanding of the accuracy of the system. A sequential feature selection analysis will be done on the features allowing a better understanding of which features perform better at distinguishing the classes. In the case of the Shaolin Kung Fu dataset this analysis will also give an understanding of which sensor locations are required to analyse the chosen movement set. Finally, each section will be concluded with a comparison against published models.

7.1 SHAOLIN KUNG FU

This section will go through the results of using the MMR bodysuit on Shaolin Kung Fu Practitioners while they transition through the 5 basic stances. The dataset described in this section is made from the movements recorded from 8 practitioners each with over 10 years of experience in the Shaolin Kung Fu. The section will begin with a description of the features used, followed by the machine learning model and conclude with an evaluation of the model used.

7.1.1 Models and evaluation

The features created in the last section were applied to several models using the Classification learner application on MATLAB [32]. There was a lot of variation in the models' accuracies, from 69.5% to 98.9%. This is due to some models, such as coarse decision tree, having too few comparisons resulting in some features being completely ignored, see Figure 59 a); while other models, such as fine gaussian SVM, use the wrong type of kernels for the features resulting in underfitting to the training data, see Figure 59 b).



a) Model 2.3 (Coarse Tree)

True Class \ Predicted Class	0	1	2	3	4	5
0	1395		50	216	12	70
1	122		120	110		
2	128		121	116		
3	131			240		
4	9				367	
5	5			1		360

b) Model 2.11 (Fine Gaussian SVM)

True Class \ Predicted Class	0	1	2	3	4	5
0	1736		2		3	2
1	122	230				
2	156		209			
3	157			214		
4	125				251	
5	104					262

Figure 59 Models with unsuited parameters results in inaccurate predictions a) Coarse decision tree; b) fine gaussian kernel SVM.

The most accurate model, see Figure 60, was the fine decision tree that had been customised to increase its number of branches. From this confusion matrix the following can be calculated: accuracy = 98.9%, precision = 96.9%, recall = 99.1% and f1-score = 98.0%

Decision trees are known to overfit to a dataset, therefore it is important to take steps to reduce the probability of overfitting. As with the modified UPDRS dataset this dataset has been validated using k-fold cross validation. In the case of this dataset however this has been combined with hold-out validation, this process is called out-of-fold predictions. In this case the hold out data is entirely from one of the practitioners, meaning the final test set will be a combination of data from practitioners used to train the model and data from this held out practitioner. Large number of features is another cause for overfitting, therefore as with the modified UPDRS dataset a sequential feature selection analysis has been done to reduce the number of features. On top of this the features associated to the arms and head have also been removed as these were determined from the descriptions given in chapter 6 to be less relevant than the other locations. This was confirmed using a sequential feature selection where it was found the features associated to the arms and head never positively impacted the model's accuracy.

After eliminating the arm and head sensors, the optimal number of splits was approximately 300, however this varied a lot, from 200 to 400. This variance was mainly due to the 0 class which consisted of transitions in the Wu Bu Quan form as well practitioners doing other stances/stretches. This class has been used to create potential false positives in the model. During the 0 class measurements the practitioners performed stretches that were like the stances but were never actually the stances; this as well as the transition points in the Wu Bu Quan form gave many opportunities for false positive measurements. Although false positive measurements are not desired, having data that could possibly be considered a false positive reduces the window of fitting and minimises the chance the model underfits.

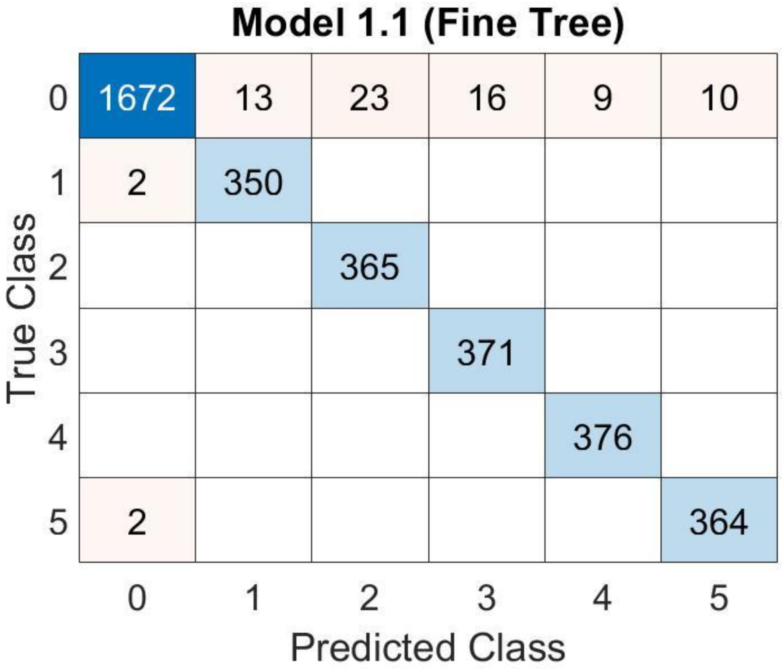


Figure 60 Confusion matrix of continually initialised data model

Table 10 shows the results of using the forward sequential feature selection analysis on the fine decision tree model. The corresponding feature labels are in Table 11. There are two important aspects of Table 10 that should be highlighted: the features associated with yaw and the feature grouping. Note that the forward sequential feature analysis was completed 10 times, each time using a different random seed 10-fold cross validation, to determine if the result of the feature analysis was biased towards any training set. The following observation were present on all 10 cases with the specific features/feature group changing from cases to case.

The features associated to yaw have been "*" in Table 10, 3 of these features have no effect on the model's accuracy while 5 decreases the accuracy. This is expected when looking back at the variation of the yaw measurements in Table 9. In the case of the yaw features that have no effect on the model, the high variation in the yaw measurements have resulted in large truth windows that are made inconsequential due to other features already narrowing these truths. In the case of the 5 features which have resulted in the reduction of the model's accuracy, the truth windows established from the training data do not encompass the entire dataset's yaw measurements. However, there is only a slight reduction in the model's accuracy meaning it can be assumed that mostly these yaw features are also inconsequential with only some impact on the truth windows established by the rest of the model's features. As mentioned previously the specific yaw features changed from case to case with no clear pattern emerging during the 10 cases; however, in all 10 case the yaw features were either inconsequential or caused inaccuracies in the model.

To fully understand which features are useful to the models and which are causing generalisation errors, the sequential feature selection analysis has been coded to include all features in a forward fashion. Therefore, all 48 features will be included starting with the most important and finishing with the least important. The criterion value at each step is the cumulative mean squared error considering the last added feature and all features before.

The highlighted colours in Table 10 correspond to features that belong to the same sensor that have been selected sequentially by the forward sequential feature selection analysis. There are two interesting observations to be made when considering these groups: the sensor group corresponding locations and the progression of the criterion value moving from one sensor group to the next. It can be seen in table 10 the first four coloured groups increase the system's accuracy greatly, while the next two groups have minimal effect, and every group after a negative effect. It can therefore be concluded that only 4 device locations are required to accurately distinguish the movement classes, while 6 device are optimal.

In all 10 cases features associated to the thighs have been considered the most important for the model's accuracy, with 7 cases having a group of features associated to the left thigh first and 3 cases the right thigh. The third most important group in all 10 cases either go to the features associated with the upper back or the lower back: 5 cases upper back, 5 cases lower back. If the upper back features are considered highly important the lower back features end up having no effect on the model's accuracy and vice versa. This would suggest that the features associated to the upper back and lower back are highly correlated. This is expected considering the importance a straight back holds in Shaolin Kung Fu, however if this model was to be used on non-practitioner the correlation of these locations would likely reduce dramatically. This would suggest the false positive attempts in class 0 did not give the model sufficient measurements across these locations. The next most important groups appear to be random requiring more cases to definitively rank the location in terms of effect on accuracy of the model.

Another interesting observation to be made from Table 10 can be seen at steps 8, 12 and 15. The criterion value at these steps indicate the model's accuracy is either unchanged or decreases when including these features. This would suggest that after a point including individual features from new sensor locations does not benefit the model, instead features starting a new location must be paired with another feature from that location to increase the model's accuracy. The first feature of each group is either an orientation calculation or angular velocity calculation with the paired feature being the other type of calculation. All 3 axes of angular velocity appear in these groupings while only pitch and roll calculations appear. There is no apparent correlation of which orientation calculation is paired with which angular velocity calculation.

Step	Feature	Criterion value	Step	Feature	Criterion Value
1	16	0.000514144	25	35	0.000021700
2	15	0.000123672	26	34	0.000021700
3	14	0.000061087	27	32	0.000021700
4	41	0.000044589	28	21	0.000021700
5	39	0.000039914	29	43*	0.000021700
6	40	0.000036011	30	46	0.000021700
7	38	0.000033697	31	10	0.000021700
8	47	0.000033697	32	18	0.000021700
9	44	0.000026653	33	42	0.000021700
10	45	0.000025867	34	13*	0.000021700
11	48	0.000024867	35	20	0.000021700
12	11	0.000024973	36	17	0.000021700
13	9	0.000023163	37	31*	0.000021700
14	12	0.000023013	38	33	0.000021700
15	8	0.000022911	39	24	0.000021700
16	2	0.000022913	40	6	0.000021700
17	4	0.000022211	41	5	0.000023170
18	3	0.000022208	42	19*	0.000025867
19	27	0.000022178	43	36	0.000028225
20	29	0.000021777	44	7*	0.000029017
21	28	0.000021745	45	1*	0.000030577
22	30	0.000021740	46	22	0.000044670
23	26	0.000021700	47	37*	0.000045467
24	23	0.000021700	48	25*	0.000046483

Table 10 Sequential forward feature selection of continually initialised data of the Shaolin Kung Fu movement set. The green highlighted steps indicate features that when added improve the model's accuracy; the orange highlighted steps indicate features that when added have no effect on the model's accuracy; the red highlighted steps indicate features that when added decreases the model's accuracy.

	Mean Yaw	Mean Pitch	Mean Roll	STD Z Angular Velocity	STD Y Angular Velocity	STD X Angular Velocity
Left Calf	1	2	3	4	5	6
Left Foot	7	8	9	10	11	12
Left Thigh	13	14	15	16	17	18
Lower Back	19	20	21	22	23	24
Right Calf	25	26	27	28	29	30
Right Foot	31	32	33	34	35	36
Right Thigh	37	38	39	40	41	42
Upper Back	43	44	45	46	47	48

Table 11 Feature labels

Figure 61 shows a Wu Bu Quan form with the quantised categories indicated in green, positive positives in blue and false negatives in red on the upper half of the graph. Note the amplitudes of these plots are arbitrary allowing the results to be more disguisable. This window of data has been chosen as it was the worst performing form with the most errors associated to it in the dataset. As can be seen in the figure, most of the false positive results are within one quantised step from the actual positive classes. When considering the quantising step size of 0.2s, it is possible that that the measurements have been miscategorised. Further work could include having the experts recategorising the data blindly (without seeing the model's results) or having the false classifications double checked by the experts.

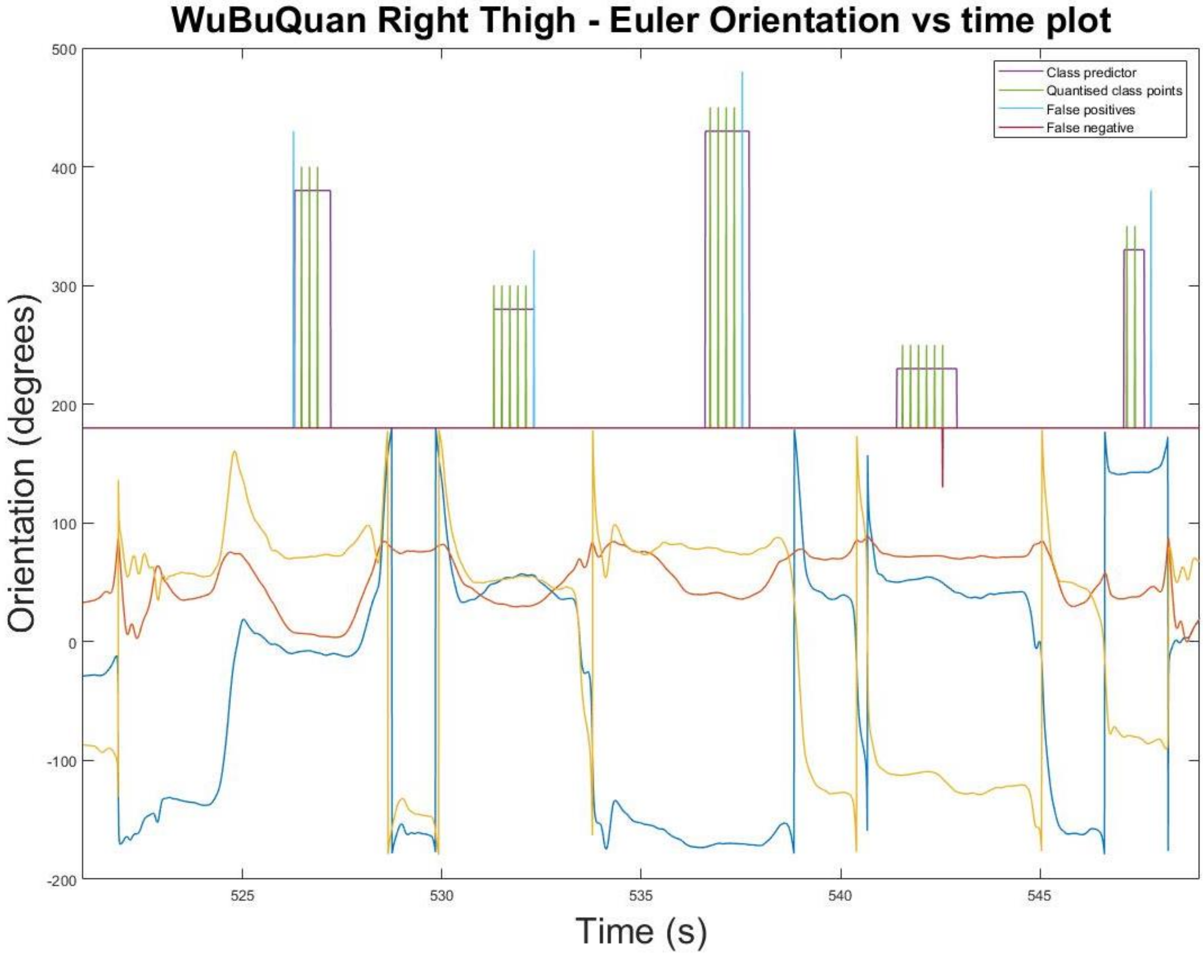


Figure 61 False positive and false negative plot of least successful sample set

7.1.2 Conclusion of Shaolin Kung Fu results

The initial results in the project show that sensors located between non-deterministic joints are considered unimportant by all models tested. The irrelevant sensors have only increased in the systems computational cost without effecting its accuracy. Therefore, it was decided to remove these sensors from the models. In hindsight the irrelevance of these location could have been predicted from the movements' predefined rule, and in future if this reference standard were to be used on a different set of rules it may be possible to reduce the number of sensors required; thus, reducing the systems price and computational cost.

The joint limitations discussed in Chapter 2 have mostly been unnecessary for the system with only a few cases resulting in warnings. This is due to the system being reinitialised before each set of movements. In future datasets where reinitialisation is not possible or the movements take longer to complete these joint limitations will play a more vital role in the models.

The features associated with yaw have also proven to be mostly redundant with a few cases where they cause inaccuracies within the model. This has been put down to two reasons: gyroscope drift is too prominent when using IMU module alone even with the use of the complimentary filter, and the movements were classified from a video recording using a single side view.

Several models were tested using the MATLAB Classification App [32], of which a few performed particularly poorly. This bad performance is due to the number of features in the model being too high for some of the low computational models. While computational cost is an important aspect of this project, it's the cost of using the model that is of interest rather than the cost of creating the model, therefore high computational cost algorithms can be used to create the model. If an application required the creation of the model to have a low computational cost, the feature dimensions could possibly be reduced using PCA, which Hachaj has proven to be a viable approach to reduce computational cost while maintaining a high accuracy [29]. Another possible approach could be excluding some sensor locations or certain features from the model, as Table 10 has shown that less than half the features are ever required for the model to reach maximum accuracy. Which features to exclude may only be possible through hindsight as determining which features would be redundant prior to seeing the full model's results would be a difficult task, however, may be possible with a thorough understanding of the predefined rule of movement.

The most accurate model was the fine decision tree customised to include 400 splits, with an accuracy of 98.9%. As seen from Figure 60, the model has high recall ability (100% for 3 classes, and >99% for 2 classes) while falling slightly on precision with an average precision of 96.2%. The model's errors were almost always (14 cases not) on the extremities of the categorisation windows which could mean these measurements may be miscategorised. Future works should include going through the errors with the experts to determine if the movements were correctly categorised to begin with.

Table 12 allows the comparison of this project's model to three similar projects' models. This project's model has achieved high accuracy levels comparable to both Hachaj's model and Bianco's model [29] [27]. Unlike these models, the model presented in this project does not require any specifications of the environment, whereas Hachaj's model requires no ferromagnetic material nearby due to the use of magnetometers and Bianco's model requires the movements to be done adjacent to the Microsoft Kinect [90]. Samiullah et al. have also used IMU technology, however only a single sensor was placed on the practitioner's sternum resulting in a much lower accuracy and a much higher variance. Due to the nature of this project being creating a reference standard it is difficult to have a direct comparison, however, looking at the accuracy of the models used for these similar predefined rules the preliminary results for this project's model is promising.

Project	Type of motion capture system	Number of modules	Features	Model	Accuracy
This project	IMU	17	Joint based	Decision Tree	98.9%
Tomasz Hachaj et al. [29]	IMU + EMS (magnetometer)	17	Joint based using PCA	SVM	94%
Christopher Samiullah et al. [28]	IMU	1	Standard statistical features	Random forest + HMM	72 ± 20%
Simone Bianco et al. [27]	Microsoft Kinect [90]	15	Part based	DTW	96.25%

Table 12 Comparison of models created to evaluate full body martial arts

7.2 MODIFIED UPDRS DATASET

7.2.1 Models

MATLAB's Classification app [32] was used to apply the 8 features to all the standardised models available in the application. This allowed a quick and reproducible way to test the features as a proof of concept. The classification app has 24 models to choose from, including parameter variants such as coarse/medium/ fine decision tree and linear/quadratic/cubic SVM. The top model of each variant has been selected in this section to aid with the project's proof of concept. These models were: Gaussian Naïve Bayes, Fine/Medium Tree, Medium Gaussian/Linear SVM and Fine/Medium KNN.

It was decided to use the k-fold cross validation method over the holdout method to minimise the possibilities of overfitting; 30 folds were chosen. The higher number of folds will lower the bias towards estimating generalisation error, however it will also increase both the variance and the computational cost. Although the computational cost is an important aspect to consider, this project is more concerned with the cost to use the model rather than to train it. With the current size of the dataset the computational cost will still be relatively low. The variance would increase as the overlap between the folds increases; this variance will be considered in the following results.

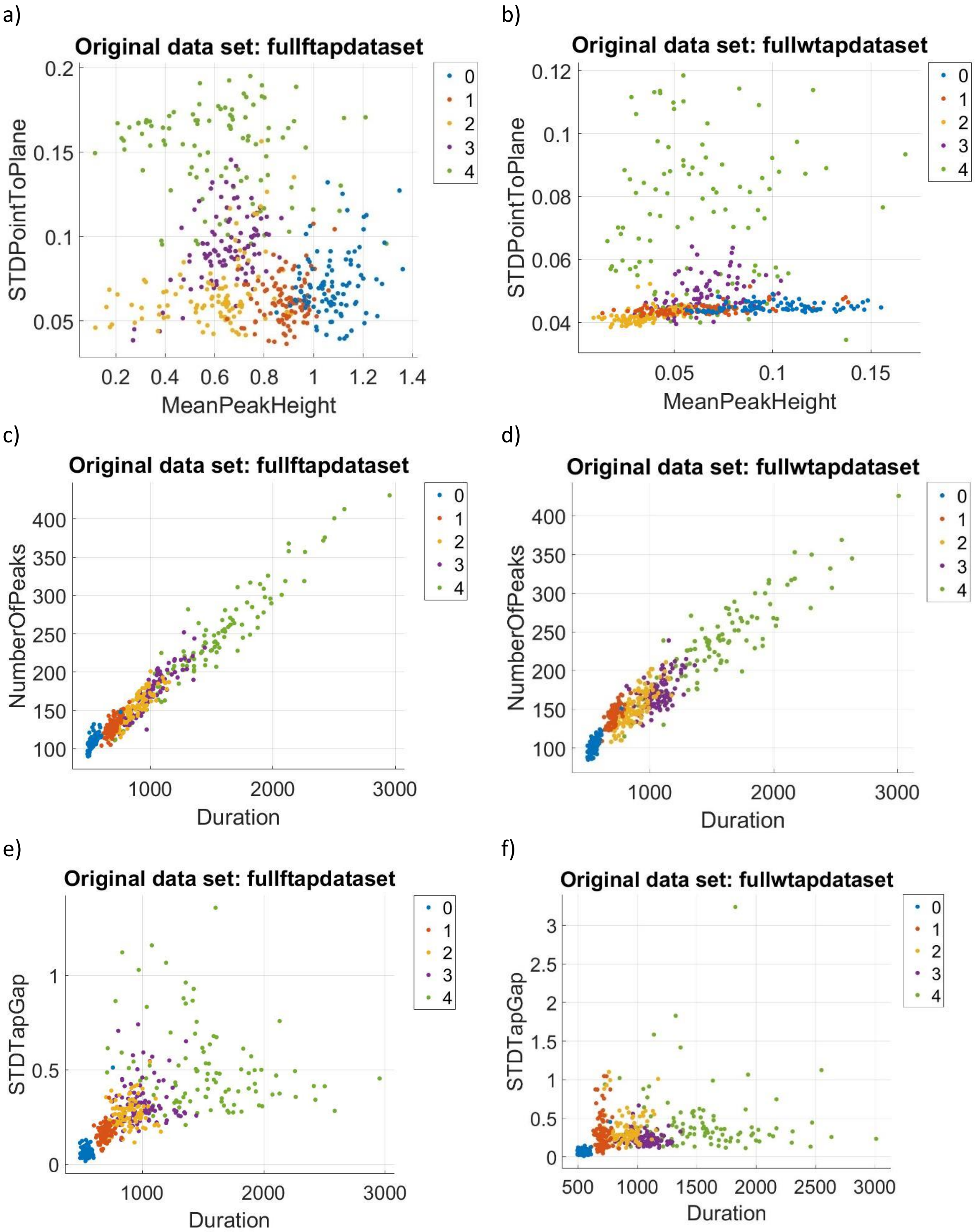
Figure 62 shows the feature combinations suggested in the previous section for the finger location and the wrist location datasets.

Figure 62 a) and b) shows the plot of the standard deviation of points to plane feature against the mean peak height feature. Although there is no correlation between these features this plot is useful in showing the difference in effectiveness the features have on the two locations. Both plots have a lot of interlaced classes, however the finger location plot Figure 62 a) has clearer edges between the classes.

Figure 62 c) and d) shows the plot of the number of peaks against the duration of the exercise. As expected, there is a strong positive correlation between these features. For both locations the classes have begun to separate with only some interlacing. It should be noted that the correlation is less strong in the wrist location, this is likely caused by the signal to noise ratio being greater in this location. While some models benefit from positive correlation, it is important not to have features that are too highly correlated. Highly correlated features can result in harmful bias if the features suffer from the same errors. Highly correlated features can also cause multicollinearity, a phenomenon where the model can become unstable and vary greatly between samples [91]. This will be considered when evaluating the models.

Figure 62 e) and f) shows the plot of the standard deviation of the tap gap against the duration of the exercise. When describing these features there was an assumed negative correlation between them, however overall, there seems to be a positive correlation with a possible weak negative correlation within each class.

Figure 62 g) and h) shows the plot of the Durbin Watson analysis index against the duration of the exercise. By combining these two features, the severity levels become more distinguished compared to using the Durbin Watson analysis index alone. Severity level 1 is more distinguished on the wrist location than the finger location although neither have overlap, meaning using just these two features the machine learning model should have a high accuracy. However due to the distance between severity level 0 and 1 on the finger location a large holdout or k fold cross-validation should be used.


a)
Original data set: fullftapdataset
STDPointToPlane
MeanPeakHeight
0.2
0.15
0.1
0.05
0.2 0.4 0.6 0.8 1 1.2 1.4
0
1
2
3
4
b)
Original data set: fullwtapdataset
STDPointToPlane
MeanPeakHeight
0.12
0.1
0.08
0.06
0.04
0.05 0.1 0.15
0
1
2
3
4
c)
Original data set: fullftapdataset
NumberOfPeaks
Duration
400
350
300
250
200
150
100
1000 2000 3000
0
1
2
3
4
d)
Original data set: fullwtapdataset
NumberOfPeaks
Duration
400
350
300
250
200
150
100
1000 2000 3000
0
1
2
3
4
e)
Original data set: fullftapdataset
STDTapGap
Duration
1
0.5
0
1000 2000 3000
0
1
2
3
4
f)
Original data set: fullwtapdataset
STDTapGap
Duration
3
2.5
2
1.5
1
0.5
0
500 1000 1500 2000 2500 3000
0
1
2
3
4

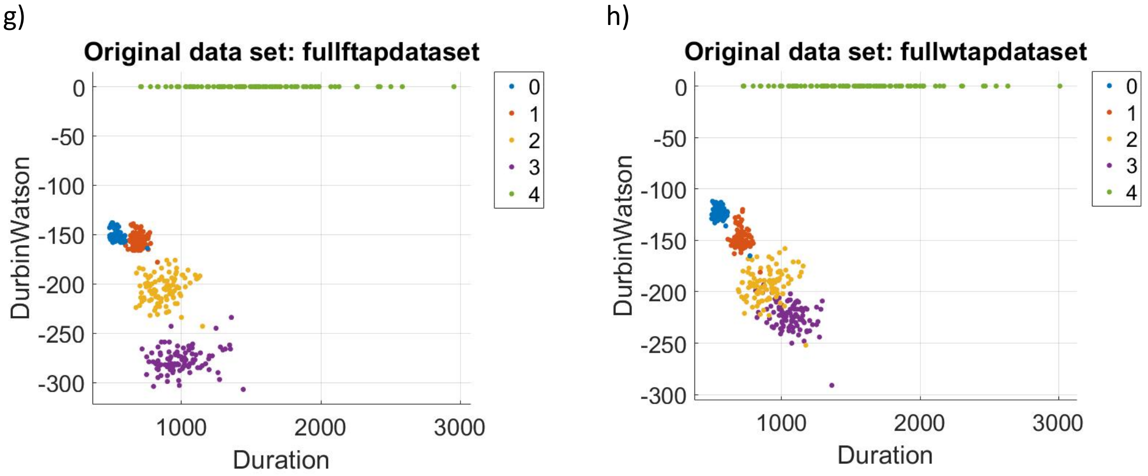


Figure 62 Each plot shows the correlation between two features for both sensor locations with the different severity levels indicated by the colours on the legends; a) the standard deviation points to plane feature plotted against the mean peak height feature for the finger location; b) the standard deviation points to plane feature plotted against the mean peak height feature for the wrist location; c) the number of peaks feature plotted against the duration feature for the finger location; d) the number of peaks feature plotted against the duration feature for the wrist location; e) the standard deviation of the tap gap feature against the duration feature for the finger location; f) the standard deviation of the tap gap feature against the duration feature for the wrist location; g) the Durbin Watson analysis index feature against the duration feature for the finger location; h) the Durbin Watson analysis index feature against the duration feature for the finger location. The units are arbitrary for this plot.

Table 13 shows the model parameters used to maximise the accuracy of each model for both the finger and wrist location. The table also includes the models' variance in accuracy using the same parameter over 20 trainings. The variance is due to which data is chosen to train the model and which is chosen to test the model due the cross-validation steps. Another important property included in the table is the prediction speed for each model considering the parameters given.

Overall, the SVM models have performed with the greatest accuracy with both finger and wrist location getting an accuracy of 99.2%. However, all the models have achieved accuracies above 97% which is an excellent benchmark for a golden standard in the MDS-UPDRS finger tapping exercise.

On top of the accuracies achieved by the models another important statistic to look at is the variance in the system. Due to the high number of folds used in the cross validation it would be expected for the variance to be high, however the highest variance seen in our models is the Fine KNN used for the wrist location with less than 1% variance. Having a low variance while using high number of folds could suggest two things about the system; either the features are well designed to consider variance in the classifications or there is not enough variation in the dataset. Considering the size of the dataset and the fact that the movements were emulated there is a possibility that it is the latter of these statements, however, as previously mentioned, when emulating the exercises an emphasis was made in doing a variety of movements that still fall under each level's classification. Moreover, looking back at the plots of the features in the previous section there is large variations throughout the dataset and even the Durbin Watson feature which has the least variation can only achieve an accuracy of 78% as a standalone feature.

Model Parameters
Kernel Naïve Bayes ***(Used for finger and wrist location)***
Kernel type: Gaussian Support: Unbound **Results** Accuracy: 98.4% (±0.70) (finger) 97.0% (±0.66) (wrist)

Total misclassification cost: 9 (finger) 15 (wrist) Prediction speed: ~1700obs/sec Training time: 3.9466 sec	
Medium Tree *(Used for finger and wrist location)*	
Maximum number of splits: 20 Split criterion: Gini's diversity index Surrogate decision splits: Off **Results** Accuracy: 98.2% (±0.2) (finger) 95.1% (±0.18) (wrist) Total misclassification cost: 9 (finger) 25 (wrist) Prediction speed: ~14000obs/sec Training time: 0,43113 sec	
Quadratic SVM *finger*	**Linear SVM *wrist***
Kernel function: Quadratic Kernel scale: Automatic Box constraint level: 1 Multiclass method: one vs one **Results** Accuracy: 99.2% (±0.69) Total misclassification cost: 4 Prediction speed: ~3900obs/sec Training time: 2.2127 sec	Kernel function: Linear Kernel scale: Automatic Box constraint level: 1 Multiclass method: one vs one **Results** Accuracy: 99.2% (±0.71) Total misclassification cost: 4 Prediction speed: ~4000obs/sec Training time: 2.3425
Medium KNN *finger*	**Fine KNN *wrist***
Number of neighbours: 10 Distance metric: Euclidean Distance weight: Equal **Results** Accuracy: 98.0% (±1.10) Total misclassification cost: 10 Prediction speed: ~8600obs/sec Training time: 0.54909 sec	Number of neighbours: 1 Distance metric: Euclidean Distance weight: Equal **Results** Accuracy: 97.3% (±0.90) Total misclassification cost: 13 Prediction speed: ~9000obs/sec Training time: 1.0151 sec

Table 13 Parameters used to structure the machine learning models including each model's results for both the finger and wrist locations.

Table 14 shows the results of applying the sequential feature selection analysis on each of the models. This analysis is used to reduce the feature dimensional space by automatically selecting a subset of features that are most relevant to the solution. It is used to reduce computational cost in the model and reduce generalisation error associated to highly correlated features. There are two main versions of the algorithm: forwards and backwards. Forward starts with no features and adds in features starting with the most important first; it continues this until a threshold is met, adding further features would cause errors, or a specified number of features has been included. Backward sequential feature selection starts with all the features included and removes features starting with the least important; it continues this until a threshold is met, removing further features would cause errors, or a specified number of features have been met. The algorithm determines importance by a criterion value, in this case the criterion is the mean squared error.

To fully understand which features are useful to the models and which are causing generalisation errors, the sequential feature selection analysis has been coded to include all features in a forward fashion. Therefore, all 8 features will be included starting with the most important and finishing with the least important. The criterion value at each step is the cumulative mean squared error considering the last added feature and all features before.

For all 8 models the Durbin Watson feature has been selected as the most important feature. Despite the finger location having a higher overall accuracy in all but the SVM models, the initial criterion value for each model is lower for the wrist location. This means the Durbin Watson feature has performed better with the wrist location dataset. This is expected looking back to Figure 58, it's clear that the feature does a better job at distinguishing the 5 levels on the wrist location while on the finger location levels 0 and 1 are not distinguishable.

The second most important feature is usually the Duration feature, however during the 20 tests there were a couple of occasions the Number of Peaks feature was considered the second most important feature. Due to their high correlation when one of these features were chosen the other feature either resulted in no change in the model's accuracy or caused generalisation errors.

Another important detail to note from Table 14, only 2-4 features were ever needed to bring the models to their highest point of accuracy. Further added features often caused generalisation errors.

Kernel Naïve Bayes finger Step 1, added column 8, criterion value 0.00305127 Step 2, added column 2, criterion value 0.000293936 Step 3, added column 3, criterion value 0.000253185 Step 4, added column 5, criterion value 0.000253185 Step 5, added column 6, criterion value 0.000253185 Step 6, added column 7, criterion value 0.000253185 Step 7, added column 4, criterion value 0.000253185 Step 8, added column 1, criterion value 0.000293936	**Kernel Naïve Bayes wrist** Step 1, added column 8, criterion value 0.00160772 Step 2, added column 2, criterion value 0.000995369 Step 3, added column 4, criterion value 0.000832731 Step 4, added column 7, criterion value 0.000954709 Step 5, added column 5, criterion value 0.000837665 Step 6, added column 3, criterion value 0.000837665 Step 7, added column 6, criterion value 0.000837665 Step 8, added column 1, criterion value 0.000878416
Medium Tree finger Step 1, added column 8, criterion value 0.00322244 Step 2, added column 2, criterion value 0.000334494 Step 3, added column 1, criterion value 0.000334494 Step 4, added column 4, criterion value 0.000334494 Step 5, added column 5, criterion value 0.000334494	**Medium Tree wrist** Step 1, added column 8, criterion value 0.00179669 Step 2, added column 2, criterion value 0.00110758 Step 3, added column 5, criterion value 0.000787056 Step 4, added column 7, criterion value 0.000787148 Step 5, added column 1, criterion value 0.000787148

Step 6, added column 6, criterion value 0.000334494 Step 7, added column 3, criterion value 0.000374239 Step 8, added column 7, criterion value 0.000414898	Step 6, added column 3, criterion value 0.000868467 Step 7, added column 6, criterion value 0.000868467 Step 8, added column 4, criterion value 0.000909126
Quadratic SVM finger Step 1, added column 8, criterion value 0.0029732 Step 2, added column 2, criterion value 0.000324453 Step 3, added column 1, criterion value 0.000202383 Step 4, added column 4, criterion value 0.000202383 Step 5, added column 5, criterion value 0.000202383 Step 6, added column 3, criterion value 0.000202383 Step 7, added column 6, criterion value 0.000202383 Step 8, added column 7, criterion value 0.000202383	**Linear SVM wrist** Step 1, added column 8, criterion value 0.00272062 Step 2, added column 2, criterion value 0.000370311 Step 3, added column 1, criterion value 0.00020749 Step 4, added column 3, criterion value 0.00020749 Step 5, added column 4, criterion value 0.00020749 Step 6, added column 5, criterion value 0.00020749 Step 7, added column 6, criterion value 0.00020749 Step 8, added column 7, criterion value 0.00020749
Medium KNN finger Step 1, added column 8, criterion value 0.00352171 Step 2, added column 2, criterion value 0.000369315 Step 3, added column 1, criterion value 0.000369315 Step 4, added column 4, criterion value 0.000369315 Step 5, added column 5, criterion value 0.000369315 Step 6, added column 6, criterion value 0.000369315 Step 7, added column 3, criterion value 0.000369315 Step 8, added column 7, criterion value 0.000415	**Fine KNN wrist** Step 1, added column 8, criterion value 0.00172751 Step 2, added column 5, criterion value 0.000949135 Step 3, added column 7, criterion value 0.000867725 Step 4, added column 4, criterion value 0.000827066 Step 5, added column 6, criterion value 0.000827066 Step 6, added column 3, criterion value 0.000908476 Step 7, added column 1, criterion value 0.000989795 Step 8, added column 2, criterion value 0.00135693

Table 14 Each element shows the response when using the sequential feature selection on the corresponding models. The green highlighted steps indicate features that when added improve the model's accuracy; the orange highlighted steps indicate features that when added have no effect on the model's accuracy; the red highlighted steps indicate features that when added decreases the model's accuracy.

7.2.2 Evaluation

This section will look at the confusion matrices of the models used, allowing a better understanding of where errors occur in the system and the models' precision and recall. However before looking at each model it should be stated that the following cost matrix, Table 15, was used to determine the classification cost of each model, as seen in Table 13. The weight values were chosen considering that the exercises have sequential levels, therefore, predicting an adjacent level should cost less to the model than predicting a further level.

X	1	2	3	4
1	X	1	2	3
2	1	X	1	2
3	2	1	X	1
4	3	2	1	X

Table 15 Cost matrix used for the modified UPDRS model evaluations

Figure 63 shows the confusion matrices for the four models (for both sensor locations) described previously in Table 13. Even in the lower performing models the misclassifications were mostly one class over. This is important to note as some variation between clinicians is possible however this would only result in one severity level difference [92].

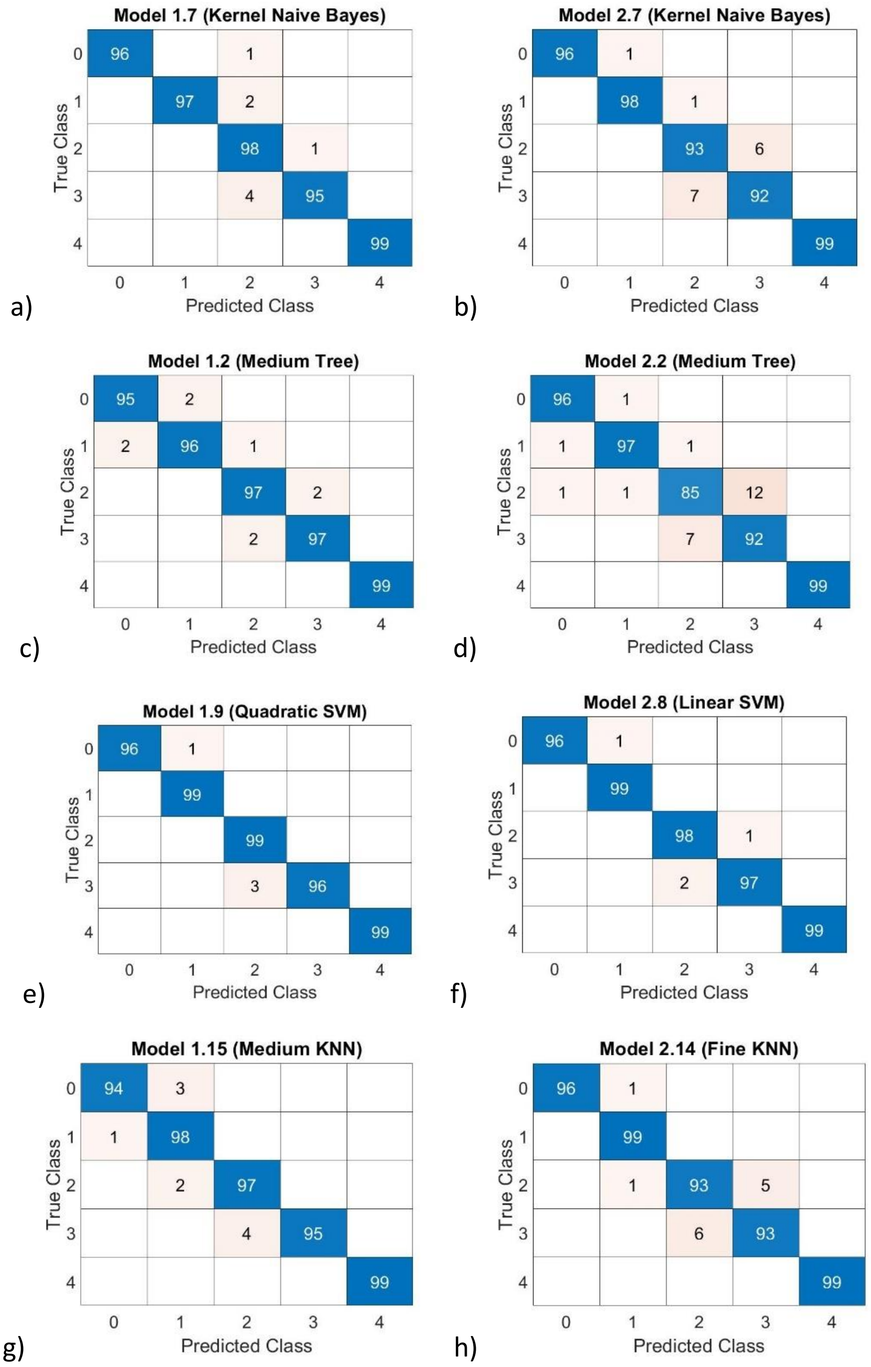


Figure 63 Confusion matrices for each model analysed a) Kernal naïve bayes for finger position; b) Kernal naïve bayes for wrist position; c) Medium decision tree for finger position; d) Medium decision tree for wrist position; e) Quadratic kernal SVM for finger position; f) Linear kernal SVM for wrist position; g) Medium KNN for finger position; and h) Fine KNN for wrist position.

7.2.3 Conclusion of Parkinson's results

To conclude the Parkinson's results, the Durbin Watson analysis has proven to be useful feature for distinguishing between the MDS-UPDRS 2nd, 3rd and 4th levels of severity, while standard statistical features such as number of peaks and duration were able to distinguish between the 0th and 1st level. The Durbin Watson analysis feature paired with the duration and number of peaks features whilst using a Gaussian Kernel SVM model achieved an accuracy (as well as recall, precision and f1-score) of 99.2% for both the finger location and the wrist location. Using other classification models result in similar accuracies ranging from 98.0% to 98.4% for the finger location and 95.1% to 97.3% for the wrist location. For most of the models evaluated during this project the features extracted from the finger location performed better at distinguishing the severity levels, however the difference was less than 3%.

These accuracies are considerably high for a reference standard when comparing to similar models such as *Printy et al.* model which achieved 94.5% accuracy using the gyroscope, accelerometer and touch capacitance of an iPhone to capture the motion of patients performing the old UPDRS finger tapping test [30]. Another study to compare this project results to is *Huo et al.* model which uses a combination of IMU sensors and mechanomyography sensors to measure tremor during the UPDRS tests has achieved an average accuracy of 89.9% [31].

It is hard to directly compare this project's model with previous models due to the data being emulated resulting possible inaccuracies in the representation of the movements and an imbalance in model sizes. However, preliminary results look promising for the use of a single IMU module in creating a reference standard for the modified UPDRS.

7.3 Conclusion of results

Using IMU modules has proven to be a successful reference standard for both datasets, having comparable accuracies as standards of similar published studies. As mentioned in the introduction of this thesis, these two movement sets were chosen as they allowed different aspects of an IMU based system to be evaluated. The results of the three datasets (two modified UPDRS datasets and the Shaolin Kung Fu dataset) have been summarised in Table 16.

The Parkinson's dataset has proven the use of features based on accelerometer and gyroscope measurements is a valid approach when categorising quality of movement on a repetitive motion. This is an important find as by using the accelerometer and gyroscope data directly the system has avoided any integration which in Chapter 3 has been proven to result in large errors in IMU systems. While this has only been done using a single sensor in this project. The results would likely be equally or more effective using multiple locations as it can be seen just from the two locations tested those certain locations may be more suitable at distinguishing some aspects of the movement while other locations may be better at distinguishing other aspects. Further work will have standard statistical feature based on accelerometer and gyroscope measurements applied to full body motions such as the Shaolin Kung Fu dataset created in this project or the modified UPDRS Gait test [6].

The results drawn from the Shaolin Kung Fu dataset show that orientation and angular velocity-base features can be used to create an accurate reference standard. However, the models have avoided using the yaw associated features due to their relatively high range. For this dataset it was still possible to categorise the data without the yaw calculations even with an attempt to give the system yaw specific false positive data. This success has been put down to the effect these false positive measurements have on the roll and pitch calculations making it possible to distinguish them without the need of the yaw calculations.

For both datasets the memory on the MMR modules was a limiting issue in the system. As previously mentioned, whilst using the 200 Hz sampling frequency the modules were only able to record accelerometer and gyroscope data for 22 minutes. This window was very small to setup all the devices separately and record the movements for the Shaolin Kung Fu dataset resulting in several sessions being necessary. On top of that the sessions had to be spaced out as uploading all the data via BLE would take over 12 hours. This limited space was similarly an issue for the Parkinson's dataset; however, this was due to the data being emulated thus having several exercises done sequentially. This 22-minute window should be enough time to record the tapping exercise however considering the goal is to eventually have all the modified UPDRS motor examinations recorded using this IMU system this window would once again become problematic. For both datasets this short recording window would be less of a problem if the data could be uploaded quickly between exercises. As it stands this is not possible with commercial systems only having a BLE upload option. Future work should include finding or building a system where multiple devices can be setup in parallel and data can be uploaded via a hard data line.

Dataset	**Number of sensors**	**Features**	**Model**	**Accuracy**
Parkinson's modified UPDRS	1 (wrist and finger location tested)	Autocorrelation and standard statistic features based on accelerometer and gyroscope data	Quadratic SVM (finger) Linear SVM (wrist)	99.2% (both locations)
Shaolin Kung Fu basic stances	17	Standard statistic features based on orientation and angular velocity data	Fine decision tree with 400 splits	98.9%

Table 16 Summary of datasets methods and results

8 CONCLUSION

To begin the final chapter of this thesis we will refer to the aims, objectives and requirements given in chapter 1 outlining how well the thesis have accomplished these. Following this a conclusion of the proposed system's results will be given. Next the usability and impact of the system described in this thesis will be discussed. Finally, the main contributions, the project's limitations and future works will be outlined.

The thesis started by identifying a problem that many clinical and sports science analyses do not have a gold standard due to limitations and drawbacks in the current motion capturing benchmarks. Aims were identified from these limitations and drawbacks and two movement sets were selected to test the solution of the project on a wider scope. An IMU based system was chosen to be the foundation of the project as its fundamental operations did not contradict the requirement analysis, see chapter 1, of the project. Having chosen a basis for a system, the research aim and 6 research objectives were set. These will be highlighted during the following thesis summary.

Many of the requirements of this project were given as delimitations to the project's direction moving forwards. The project has successful matched similar project's accuracies **[Requirement 6]** for both the Shaolin Kung Fu movement set and the Parkinson's movement set **[Requirement 7]** without requiring any expectations of the working environment **[Requirement 1]**. It has also managed to do this without requiring high computational processing **[Requirement 3]**, having been completed using an Intel i3 8100 processor and 4GB RAM, a setup which is considered on the lower end of commercial computers during the time of this project.

When creating the bodysuit for the Shaolin Kung Fu movement set the locations of the sensors were specifically chosen considering the movements to be done. The locations of the sensors were adjusted so that the experts were able to do the movements without the sensor restricting their movement **[Aim 4]**. This was possible due to the assumption of the joints being perfect rigid segments, meaning the sensors could be placed anywhere along the joint segment and result in the same orientation change. This assumption is inaccurate due to soft tissue around the bones however the errors associated with this assumption have proven to be low enough to still produce an accurate system.

Two sensor locations were tested for the Parkinson's movement set: the finger and the wrist. The finger location was thought to be more likely to produce an accurate model while being more invasive to the finger tapping movement. While it is true the finger location was invasive to the tapping movement, requiring the middle finger and index finger to be strapped together, the wrist location has proven to be as accurate as the finger location. The wrist location generally performed less accurately than the finger location however with the optimal feature it managed to match its accuracy without being invasive to the tapping movement **[Requirement 4]**.

By using commercial devices, it is possible to get an estimate of the cost effectiveness of the project. At the time of purchase the MMR modules were $75 each while the bodysuit was $20, meaning the system minus software would cost $1295, nearly 10% of the market's current gold standards' starting price (also excluding software). Although the post processing done during this project used MATLAB no specialised tools were used that do not have an equivalent of open-source alternatives such as Python. The project has therefore successfully achieved its aim of costing less than $2000 for full body motion capture **[Requirement 2]**.

Unfortunately, not all the aims of the project were fully achieved. While the sensors themselves were fully portable **[Requirement 5]** and required no external setup, the user application, Metabase [79], proved to be complicated requiring some engineering experience to follow. The application was not designed with the average consumer in mind, expecting some understanding of the IMU modules. Therefore the system was not fully portable as it was not accessible without the help of an expert. Future work should improve the user interface of the associated application, or even remove the application entirely and have the system begin recording without needing external initialisation.

The other requirement not fully achieved is designing a system that meets the requirements of the Health Insurance Portability and Accountability Act **[Requirement 8]** [20]. While neither movement set requires any personal information about the users, such as body measurements, meaning the data would not be considered PHI, this may not be the case with all moevement sets. The system designed in this project has avoided the use of any personal information, however considering **Requirment 7** the system should be flexible to a variety of movement sets. This could include sets where personal information would be included in the feature set. In such cases the data being transferred via BLE would be unsecure using the firmware present on the MMR modules which uses "Just Works" Secure Simple Pairing [93]. Therefore, future work should adjust the firmware to increase the security of the device requiring at least "Passkey Entry" Secure Simple Pairing and potentially further techniques to protect against "Man-in-the-middle" attacks [93].

The other aspect of security that has not been considered in this project is storing the data. As previously mentioned, the data collected during this project did not fall under the guideline of PHI meaning no storage security was necessary, however for future applications that may be PHI there are several safeguards that must be met, including administrative, physical and technical measures [94]. Whilst these considerations may be necessary for future applications, an immediate future task associated to this project should be labelling and exporting the datasets created in this project to an online platform to allow other research to have access to the data collected.

Once the aim, objectives and requirements of the system were given, the thesis continued to chapter 2 where the movement sets were described in detail. The chapter went on two explain the two possible orientations representations commonly used in motion capture, indicating that Euler angles were more comprehensive however suffered from Gimbal lock whereas Quaternion orientations were complicated to use/understand but did not suffer from Gimbal lock. The chapter continues outlining the human joint limitations and how these limitations can be used to constrain the system, therefore reducing the possibility for errors.

Chapter 3 began by identifying the available commercial IMU devices and subjugating them to a list of criteria based on conversations had with the associated experts in each movement set **[Objective 1]**. The chapter continue onto identifying the sources of the errors associated to IMU-devices and described methods to eliminate many of these sources via sensor calibration. However, some of the errors cannot be calibrated out and must be filtered during post processing. The calibration methods described in this chapter were then used on Mbientlab's MMR module. In chapter 4 a short literature review on possible motion capturing assessment techniques was completed. It was concluded that the best technique to assess the MMR modules was using precise actuators, the ideal device would be the CMM, [68], however as it was not available the L-611 Precision Rotation Stage would later be used [95]. Chapter 5 reviewed several fusion filters suitable for IMU systems, outlining each filter's theory. The chapter ends with each filter being assessed in the frequency domain using the L-611. It was concluded that the particle filter using 10,000 particles was the most accurate but too computationally costly, therefore the next most accurate the complementary filter was chosen moving forward in the project.

The experimental method for the project was described in chapter 6, starting with sensor placement and concluding with the extraction of features from the three sets of processed measurements: resulting in the formation of 3 new datasets **[Objective 2]**.

A moving quantised window of 0.2 seconds was used to reduce the Shaolin Kung Fu data size. For each step the mean of the orientation and the standard deviation of the angular velocity was calculated forming the feature set for the dataset **[Objective 3]**. Forward sequential feature selection indicated that the features associated with yaw were too varied to have any impact on the model, with some cases having a negative impact on the model's accuracy. The sensors located on the arms including the shoulders and head were also found to have no impact on the model. In hindsight this is expected from the movement set description. It was concluded that only 4 sensors were required to have an accuracy commensurate with similar published models **[Objective 5]**. The features were applied to all the models of the classification app. Some models such as the coarse decision tree performed very poorly with 69.5% accuracy. Of the 24 models tested the fine decision tree customised to have 400 splits performed best with accuracy of 98.9%. An out-of-fold cross validation method was used, with one of the practitioner's measurements entirely held off for the testing set, to reduce the possibility of overfitting. The model's accuracy successfully matched the standards set by models of similar movement sets seen in Table 12 **[Objective 4]**.

The following features were extracted for the modified UPDRS: duration, number of peaks, average peak height, number of detected taps, standard deviation of tap height, standard deviation of tap gap, standard deviation of points to plane analysis and index value of Durbin Watson analysis. Forward sequential feature selection analysis was completed to determine the importance of each feature. The feature associated to the Durbin Watson analysis was most effective at distinguishing severity level 2, 3 and 4, with number of peaks being the next most important, distinguishing levels 0 and 1 **[Objective 3]**. Most models tested resulted in high accuracies, with gaussian (finger location) SVM and linear (wrist location) SVM being the most accurate with 99.2%. The model's accuracy successfully matched the standards of the models used for the old UPDRS movement set **[Objective 4]**. The finger location's model mostly outperformed the wrist location's model, however only by a small 3% margin. Additionally, as stated before, using the optimal features and model both locations performed equally well **[Objective 6]**.

As both movement set have had machine learning models constructed with accuracies commensurate with similar published studies, it can be said the project has achieved its **aim** stated in chapter 1.

The system developed in this thesis could be used in several applications both in sports science and in the clinic. Many of these applications stem from the constructed system not requiring any external devices or information, allowing usage both indoors and outdoors. The three-dimensional data would allow a professional to precisely monitor the user's movement. Depending on the professional the following applications in sports science and in the clinic would be possible:

- Outpatients for physiotherapists
- Teaching in sports
- Injury monitoring in a sporting environment and quotidian environment

Currently the systems constructed in this thesis are not yet ready for real-world environments, however the results presented are promising. By designing a basic user interface this system could be distributed to researchers to increase the datasets and gain user feedback bring it one step closer to being implemented as a commercial syste.

Main contributions:

- A novel approach using a precise rotary stage, the L-611, to validate a fusion filter to reduce stochastic errors has been implemented in chapter 4. To the best of my knowledge the approach of comparing the fusion filters in the frequency domain against the ground truth given by the precise rotary stage has not been published before.
- Three datasets have been constructed during this project: the Shaolin Kung Fu dataset contain 6 classes and over 3500 well distributed samples from 8 participants; and two modified UPDRS tapping test datasets which includes self-emulated tapping data of approximately 100 samples for each of the 5 severity levels (total of approximately 500 samples) for two locations (the index finger and the wrist). These datasets will be accessible on the follow github profile: https://github.com/Thumus.
- An evaluation on the effect location has when using a single IMU device to capture motion has been completed with results indicating that devices placed near a joint compared to on the joint are still able to distinguish differences in the movements so long as well-designed features are used.
- In addition to using some standard statistical features, two novel features were extracted from the modified UPDRS dataset: the standard deviation of points to plane analysis and the index of the Durbin Watson analysis. The latter feature played an essential part to the final accuracies achieved by the proposed model.
- A novel technique using forward sequential feature selection to determine the minimum number of IMU devices required to fully distinguish the movement classes was described and implemented in section 7.1. It was concluded that the optimal number of devices for this Shaolin Kung Fu movement set was 4, however the accuracy continues to increase until 6 devices.
- An evaluation of 24 models has been done on all 3 datasets and the optimal model for each dataset has been determined.

Limitations:

- As with any project where a large amount of data must be collected, time was a big limiting factor in this project. In fact, the following limitations are all subject to this time limitation.
- The modified UPDRS datasets were emulated rather than collected from actual patients. Other than time another contributing factor to this limitation was lockdown due to COVID-19 resulting in self-emulated data being the only option available.
- Only a single item from the modified UPDRS test was tested in this project, limiting the generalisation of the system proposed.
- For the Shaolin Kung Fu dataset only 5 labelled classes were used. This number of classes is low, again limiting the generalisation of the system proposed.
- Only 8 participants were measured for the Shaolin Kung Fu dataset and all these participants were trained under the same master, again limiting the generalisation of the system proposed.
- For all three datasets the labelling process was done by watching a video recording of the experiments. This limits the viewing angle resulting in possible mislabelled data.

The limitations above will form the basis for the future work outlined below:

In the future of this project the Shaolin Kung Fu dataset should be extended to included participants from other schools. With more participants in the study further analysis can be done on the effect certain attributes have on the performance of the system, such as height, weight and gender. Additionally, by including more participants to the study the system's ability to generalise will be increase reducing the chances that overfitting has occurred. After more participants are included in the study the next set to improve the dataset would be including more classes, i.e., Shaolin Kung Fu stances.

The future work for the modified UPDRS datasets should be to include further datasets base of the items: 3.5 (hand), 3.6 (forearm), 3.7 (ankle) and 3.8 (hip) in the modified UPDRS instruction booklet [6]. These new items, like the tapping exercise, focus on a single part of the body and therefore should be straightforward to implement the system described in this project for these datasets. As with the tapping exercise the effect of the device location should be analysed.

Finally, the labelling process should be improved for future datasets, including more viewing angles from video recording and multiple experts from different training backgrounds to make the system more generalised.

9 BIBLIOGRAPHY

[1] N. Crowther, Sports in Ancient Times, University of Oklahoma Press, 2007.

[2] B. Elliott, "Biomechanics: an integral part of sport science and sport medicine," *Journal of Science and Medicine in Sport,* vol. 2, no. 4, pp. 299-310, 1999.

[3] G. Lee and F. E. Pollo, "Technology Overview: The Gait Analysis Laboratory," *Journal of Clinical Engineering,* vol. 26, no. 2, pp. 129-135, 2001.

[4] J. Feng, J. Wick, E. Bompiani and M. Aiona, "Applications of gait analysis in pediatric orthopaedics," *Current Orthopaedic Practice,* vol. 27, no. 4, pp. 455-464, 2016.

[5] A. W. S. Rutjes, J. B. Reitsma, A. Coomarasamy, K. S. Khan and P. M. M. Bossuyt, "Evaluation of diagnostic tests when there is no gold standard. A review of methods," *HEALTH TECHNOLOGY ASSESSMENT-SOUTHAMPTON-,* vol. 11, no. 50, 2007.

[6] C. G. Goetz, B. C. Tilley, S. R. Shaftman, G. T. Stebbins, S. Fahn, P. Martinez-Martin and e. al, "Movement Disorder Society-Sponsored Revision of the Unified Parkinson's Disease Rating Scale (MDS-UPDRS): Scale presentation and clinimetric testing results," *Movement Disorders,* vol. 23, no. 15, pp. 2129-2170, 2008.

[7] E. van der Kruk and M. M. Reijne, "Accuracy of human motion capture systems for sport applications; state-of-the-art review," *European journal of sport science,* vol. 18, no. 6, pp. 806-819, 2018.

[8] M. Begon, F. Colloud, V. Fohanno, P. Bahuaud and T. Monnet, "Computation of the 3D kinematics in a global frame over a 40 m-long pathway using a rolling motion analysis system," *Journal of biomechanics,* vol. 42, no. 16, pp. 2649-2653, 2009.

[9] J. Spörri, C. Schiefermüller and E. Müller, "Collecting kinematic data on a ski track with optoelectronic stereophotogrammetry: a methodological study assessing the feasibility of bringing the biomechanics lab to the field," *PloS one,* vol. 11, no. 8, p. e0161757, 2016.

[10] A. Panjkota, I. Stancic and T. Supuk, "Outline of a qualitative analysis for the human motion in case of ergometer rowing," *WSEAS International Conference. Proceedings. Mathematics and Computers in Science and Engineering,* no. 5, 2009.

[11] M. Windolf, N. Götzen and M. Morlock, "Systematic accuracy and precision analysis of video motion capturing systems—exemplified on the Vicon-460 system," *Journal of biomechanics,* vol. 41, no. 12, pp. 2776-2780, 2008.

[12] A. Stelzer, K. Pourvoyeur and A. Fischer, "Concept and application of LPM-a novel 3-D local position measurement system," *IEEE Transactions on microwave theory and techniques,* vol. 52, no. 12, pp. 2664-2669, 2004.

[13] H. M. Schepers and P. H. Veltink, “Stochastic magnetic measurement model for relative position and orientation estimation,” *Measurement science and technology,* vol. 21, no. 6, p. 65801, 2010.

[14] J. S. Day, G. A. Dumas and D. J. Murdoch, “Evaluation of a long-range transmitter for use with a magnetic tracking device in motion analysis,” *Journal of Biomechanics,* vol. 31, no. 10, pp. 957-961, 1998.

[15] D. Zhong and S.-F. Chang, “Real-time view recognition and event detection for sports video,” *Journal of Visual Communication and Image Representation,* vol. 15, no. 3, pp. 330-347, 2004.

[16] E. Lluna, V. Santiago, B. Defez, L. Dunai and G. Peris-Fajarnes, “Velocity vector (3D) measurement for spherical objects using an electro-optical device,” *Measurement,* vol. 44, no. 9, pp. 1723-1729, 2011.

[17] M. Brodie, A. Walmsley and W. Page, “Dynamic accuracy of inertial measurement units during simple pendulum motion,” *Computer Methods in Biomechanics and Biomedical Engineering,* vol. 11, no. 3, pp. 235-242, 6 2008.

[18] “Perception Neuron,” [Online]. Available: https://neuronmocap.com/. [Accessed 14 11 2020].

[19] D. Roetenberg, H. Luinge and P. Slycke, “Xsens MVN: full 6DOF human motion tracking using miniature inertial sensors,” *Xsens Motion Technologies BV, Tech. Rep,* vol. 1, pp. 1-7, 2009.

[20] C. Zohlandt, L. Walk and W. & Nawara, “Classification of Vault Jumps in Gymnastics,” p. 1–9, 2012.

[21] J. B. Lee, B. J. Burkett, D. V. Thiel and D. A. James, “Inertial sensor, 3D and 2D assessment of stroke phases in freestyle swimming,” *Procedia Engineering,* vol. 13, pp. 148-153, 1 2011.

[22] “VICON VISUALIZATION TOOL,” Vicon Motion Systems Ltd UK, [Online]. Available: https://www.vicon.com/visualization/. [Accessed 14 11 2020].

[23] “Xsens MVN Analyze Price Request,” Xsens, [Online]. Available: https://content.xsens.com/mvn-analyze-pricing. [Accessed 14 11 2020].

[24] J. Collins and M. Huynh, “Estimation of diagnostic test accuracy without full verification: A review of latent class methods,” *Statistics in Medicine,* vol. 33, no. 24, pp. 4141-4169, 2014.

[25] J. D. Groot, “What is HIPAA Compliance?,” Digital Guardian, 8 10 2020. [Online]. Available: https://digitalguardian.com/blog/what-hipaa-compliance. [Accessed 14 11 2020].

[26] B. Pueo and J. M. Jimenez-Olmedo, “Application of motion capture technology for sport performance analysis,” *Retos,* vol. 32, p. 241–247, 2017.

[27] S. Bianco and F. Tisato, “Karate moves recognition from skeletal motion,” *Three-Dimensional Image Processing (3DIP) and Applications 2013,* vol. 8650, 2013.

[28] C. Samiullah, S. Devlin and U. Suess, “Classifying Martial Arts Motion from a Single Wearable Sensor,” 2016.

[29] T. Hachaj, “Improving Human Motion Classification by Applying Bagging and Symmetry to PCA-Based Features,” *Symmetry,* vol. 11, no. 10, p. 1264, 2019.

[30] B. P. Printy, L. M. Renken, J. P. Herrmann, I. Lee, B. Johnson, E. Knight, G. Varga and D. Whitmer, “Smartphone application for classification of motor impairment severity in Parkinson's disease,” *2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society,* pp. 2686-2689, 2014.

[31] W. Huo, P. Angeles, Y. F. Tai, N. Pavese, S. Wilson, M. T. Hu and R. Vaidyanathan, “A Heterogeneous Sensing Suite for Multisymptom Quantification of Parkinson’s Disease,” *IEEE Transactions on Neural Systems and Rehabilitation Engineering,* 2020.

[32] “Train Classification Models in Classification Learner App,” [Online]. Available: https://uk.mathworks.com/help/stats/train-classification-models-in-classification-learner-app.html. [Accessed 25 10 2020].

[33] P. Wang, Chinese Wushu English Teaching, Soochow University Press, 2019.

[34] T. K. a. T. D. D. Demirdjian, “Constraining Human Body Tracking,” *Proceedings Ninth IEEE International Conference on Computer Vision,* vol. 2, pp. 1071-1078 , 2003.

[35] J.-C. Eloy, “MEMS pace picking up,” *III-Vs Review,* pp. 34-35, 2005.

[36] H. Zhu, P. Wang and Z. Fan, “Evolutionary design optimization of MEMS: A brief review,” *Proceedings of the IEEE International Conference on Industrial Technology,* pp. 1683-1687, 2010.

[37] D. Titterton and J. Weston, Strapdown Inertial Navigation Technology, vol. 17, IET, 2004.

[38] C. Combettes and V. Renaudin, “Comparison of misalignment estimation techniques between handheld device and walking directions,” *2015 International Conference on Indoor Positioning and Indoor Navigation, IPIN 2015,* pp. 1-8, 2015.

[39] D. Roetenberg, “Inertial and Magnetic Sensing of Human Motion,” *These de doctorat,* vol. PhD, p. 126, 2006.

[40] E. M. Diaz, O. Heirich, M. Khider and P. Robertson, “Optimal sampling frequency and bias error modeling for foot-mounted IMUs,” *2013 International Conference on Indoor Positioning and Indoor Navigation, IPIN 2013,* pp. 1-9, 2013.

[41] L. Zhou, E. Fischer, C. Tunca, C. M. Brahms, C. Ersoy, U. Granacher and B. Arnrich, “How we found our IMU: Guidelines to IMU selection and a comparison of seven IMUs for pervasive healthcare applications,” *Sensors,* vol. 20, no. 15, p. 4090, 2020.

[42] “mbientlab MMR,” [Online]. Available: https://mbientlab.com/metamotionr/. [Accessed 24 10 2020].

[43] J. Ferguson, “Calibration of Deterministic IMU Errors,” 2015.

[44] E. Optics, “http://eksmaoptics.com/out/media/Mechanics.pdf,” 08 08 2018. [Online]. Available: http://eksmaoptics.com/out/media/Mechanics.pdf.

[45] X. Yuan, S. Yu, S. Zhang, C. Liu and S. Liu, “Modeling and analysis of wearable low-cost MEMS inertial measurement units,” *Proceedings of the Electronic Packaging Technology Conference, EPTC,* pp. 542-546, 2014.

[46] “Xsens - Motion Tracking - 3D Motion Tracking Technology,” [Online]. Available: https://www.xsens.com/. [Accessed 24 10 2020].

[47] N. J. Uke and R. C. Thool, “Motion tracking system in video based on extensive feature set,” *Imaging Science Journal,* vol. 62, no. 2, pp. 63-72, 2014.

[48] S. Corazza, L. Mündermann, E. Gambaretto, G. Ferrigno and T. P. Andriacchi, “Markerless motion capture through visual hull, articulated ICP and subject specific model generation,” *International Journal of Computer Vision,* vol. 87, no. 1-2, pp. 156-169, 2010.

[49] T. Von Marcard, G. Pons-Moll and B. Rosenhahn, “Human Pose Estimation from Video and IMUs,” *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 38, no. 8, pp. 1533-1547, 2016.

[50] L. Becker, “Evaluation of joint angle accuracy using markerless silhouette-based tracking and hybrid tracking against traditional marker tracking – evaluation for complex movements,” *Educations Days Simi,* 2015.

[51] L. Dong, J. Wu and X. Chen, “Real-time physical activity monitoring by data fusion in body sensor networks,” *FUSION 2007 - 2007 10th International Conference on Information Fusion,* pp. 1-7, 2007.

[52] G. B. Guerra-filho, “Optical motion capture: Theory and implementation,” *Journal of Theoretical and Applied Informatics (RITA},* pp. 61-89, 2005.

[53] Optitrack, “Build your own,” 08 08 2018. [Online]. Available: https://optitrack.com/systems/.

[54] Z. Q. Zhang and J. K. Wu, A novel hierarchical information fusion method for three-dimensional upper limb motion estimation, vol. 60, IEEE, 2011, pp. 3709-3719.

[55] “SMART-DX,” [Online]. Available: https://www.btsbioengineering.com/products/smart-dx/. [Accessed 24 10 2020].

[56] T. L. Baldi, F. Farina, A. Garulli, A. Giannitrapani and D. Prattichizzo, “Upper body pose estimation using wearable inertial sensors and multiplicative kalman filter,” *IEEE Sensors Journal,* vol. 20, no. 1, pp. 492-500, 2019.

[57] E. Ruffaldi, L. Peppoloni and A. Filippeschi, “Sensor fusion for complex articulated body tracking applied in rowing,” *Proceedings of the Institution of Mechanical Engineers, Part P: Journal of Sports Engineering and Technology,* vol. 229, no. 2, pp. 92-102, 2015.

[58] L. Sy, N. H. Lovell and S. J. Redmond, “Estimating lower limb kinematics using a Lie group constrained EKF and a reduced wearable IMU count,” *2020 8th IEEE RAS/EMBS International Conference for Biomedical Robotics and Biomechatronics (BioRob),* pp. 310-315, 2020.

[59] D. Roetenberg, H. Luinge and P. Slycke, “Xsens MVN: full 6DOF human motion tracking using miniature inertial sensors,” *Xsens Motion Technologies BV, …,* vol. 1, pp. 1-7, 2009.

[60] R. V. Vitali, R. S. McGinnis and N. C. Perkins, “Robust error-state Kalman filter for estimating IMU orientation,” *IEEE Sensors Journal,* vol. 21, no. 3, pp. 3561-3569, 2020.

[61] R. V. Vitali, S. M. Cain, R. S. McGinnis, A. M. Zaferiou, L. V. Ojeda, S. P. Davidson and N. C. Perkins, “Method for estimating three-dimensional knee rotations using two inertial measurement units: Validation with a coordinate measurement machine,” *Sensors,* vol. 17, no. 9, p. 1970, 2017.

[62] J. Betz, C. Klingspor and T. Seel, “IMU-based Assessment of Ankle Inversion Kinematics and Orthosis Migration,” *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC),* pp. 6395-6400, 2019.

[63] M. El-Gohary and J. McNames, “Human Joint Angle Estimation with Inertial Sensors and Validation with A Robot Arm,” *IEEE Transactions on Biomedical Engineering,* vol. 62, no. 7, pp. 1759-1767, 2015.

[64] “C3 Series of Compact 6-Axis Robots,” [Online]. Available: https://global.epson.com/company/corporate_history/milestone_products/55_c3.html. [Accessed 24 10 2020].

[65] W. Wang, N. He, K. Yao and J. Tong, “Improved Kalman filter and its application in initial alignment,” *Optik,* vol. 226, p. 165747, 2021.

[66] S. Q. Liu and R. Zhu, “A complementary filter based on multi-sample rotation vector for attitude estimation,” *IEEE Sensors Journal,* vol. 18, no. 16, pp. 6686-6692, 2018.

[67] B. Esme, “Kalman Filter For Dummies,” 24 10 2020. [Online]. Available: http://bilgin.esme.org/BitsAndBytes/KalmanFilterforDummies.

[68] S. Thrun, W. Burgard and D. Fox, Probabilistic Robotics, MIT Press, 2005.

[69] D. Cao, Q. Qu, C. Li and C. He, “Research of attitude estimation of UAV based on information fusion of complementary filter,” *ICCIT 2009 - 4th International Conference on Computer Sciences and Convergence Information Technology,* pp. 1290-1293, 2009.

[70] W. T. Higgins, A Comparison of Complementary and Kalman Filtering, Vols. AES-11, IEEE, 1975, pp. 321-325.

[71] T. Islam, M. S. Islam, M. Shajid-Ul-Mahmud and M. Hossam-E-Haider, “Comparison of complementary and Kalman filter based data fusion for attitude heading reference system,” *AIP Conference Proceedings,* vol. 1919, no. 1, p. 020002, 2017.

[72] H. Hyyti and A. Visala, A DCM Based Attitude Estimation Algorithm for Low-Cost MEMS IMUs, vol. 2015, 2015.

[73] “L-611 Precision Rotation Stage,” [Online]. Available: https://www.pi-japan.jp/en/products/rotation-stages/l-611-precision-rotation-stage-1202002/. [Accessed 13 10 2020].

[74] T. Von Marcard, B. Rosenhahn, M. J. Black and G. Pons-Moll, “Sparse inertial poser: Automatic 3d human pose estimation from sparse imus,” *Computer Graphics Forum,* vol. 36, no. 2, pp. 349-360, 2017.

[75] J. H. Geissinger and A. T. Asbeck, “Motion inference using sparse inertial sensors, self-supervised learning, and a new dataset of unscripted human motion,” *Sensors,* vol. 20, no. 21, p. 6330, 2020.

[76] “Metabase App,” [Online]. Available: https://mbientlab.com/tutorials/MetaBaseApp.html. [Accessed 25 10 2020].

[77] M. E. Hussein, M. Torki, M. A. Gowayyed and M. El-Saban, “Human action recognition using a temporal hierarchy of covariance descriptors on 3D joint locations,” *IJCAI International Joint Conference on Artificial Intelligence,* pp. 2466-2472, 2013.

[78] F. Lv and R. Nevatia, “Recognition and segmentation of 3-D human action using HMM and multi-class AdaBoost,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics),* vol. 3954 LNCS, pp. 359-372, 2006.

[79] Y. Sheikh, M. Sheikh and M. Shah, “Exploring the space of a human action,” *Proceedings of the IEEE International Conference on Computer Vision,* vol. I, pp. 144-149, 2005.

[80] J. Wang, Z. Liu, Y. Wu and J. Yuan, “Mining actionlet ensemble for action recognition with depth cameras,” *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition,* pp. 1290-1297, 2012.

[81] X. Yang and Y. L. Tian, “Eigenjoints-based action recognition using naive-bayes-nearest-neighbor,” *2012 IEEE computer society conference on computer vision and pattern recognition workshops,* pp. 14-19, 2012.

[82] Y. Zhu, W. Chen and G. Guo, “Fusing spatiotemporal features and joints for 3D action recognition,” *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops,* pp. 486-491, 2013.

[83] L. Xia, C. C. Chen and J. K. Aggarwal, “View invariant human action recognition using histograms of 3D joints,” *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops,* pp. 20-27, 2012.

[84] Y. Yacoob and M. J. Black, “Parameterized Modeling and Recognition of Activities,” *Computer Vision and Image Understanding,* vol. 73, no. 2, pp. 232-247, 1999.

[85] R. Chaudhry, F. Ofli, G. Kurillo, R. Bajcsy and R. Vidal, “Bio-inspired dynamic 3D discriminative skeletal features for human action recognition,” *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops,* pp. 471-478, 2013.

[86] D. M. Gavrila and L. S. Davis, 3-D model-based tracking of humans in action: a multi-view approach, IEEE, 1996, pp. 73-80.

[87] F. Ofli, R. Chaudhry, G. Kurillo, R. Vidal and R. Bajcsy, “Sequence of the most informative joints (SMIJ): A new representation for human skeletal action recognition,” *Journal of Visual Communication and Image Representation,* vol. 25, no. 1, pp. 24-38, 2014.

[88] E. Ohn-Bar and M. M. Trivedi, “Joint angles similarities and HOG2 for action recognition,” *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops,* pp. 465-470, 2013.

[89] A. Abdulhafedh, “How to detect and remove temporal autocorrelation in vehicular crash data,” *Journal of transportation technologies,* vol. 7, no. 2, pp. 133-147, 2017.

[90] O. Wasenmüller and D. Stricker, Comparison of kinect v1 and v2 depth images in terms of accuracy and precision, Springer, 2016, pp. 34-45.

[91] M. P. Allen, “The problem of multicollinearity,” in *Understanding regretion analysis*, Boston, Springer, 1997, pp. 176-180.

[92] D. A. Heldman, J. P. Giuffrida, R. Chen, M. Payne, F. Mazzella, A. P. Duker, A. Sahay, S. J. Kim, F. J. Revilla and A. J. Espay, “The modified bradykinesia rating scale for Parkinson's disease: reliability and comparison with kinematic measures,” *Movement Disorders,* vol. 26, no. 10, pp. 1859-1863, 2011.

[93] J. Padgette, “Guide to Bluetooth Security,” *NIST Special Publication SP 800-121 revision 2,* vol. 800, p. 11, 2017.

[94] U.S. Department of Health and Human Services Office for Civil Rights, “HIPAA Administrative Simplification Regulation Text,” 2013. [Online]. Available: http://www.hhs.gov/sites/default/files/hipaa-simplification-201303.pdf. [Accessed 01 10 2021].

[95] “L-611 Precision Rotation Stage,” 13 10 2020. [Online]. Available: https://www.pi-japan.jp/en/products/rotation-stages/l-611-precision-rotation-stage-1202002/.

[96] M. Belleza, “Skeletal System Anatomy and Physiology,” 25 10 2020. [Online]. Available: https://nurseslabs.com/skeletal-system/.

[97] CrossFit, “Movement About Joints, Part 6: The Knee,” 25 10 2020. [Online]. Available: https://www.crossfit.com/essentials/movement-about-joints-part-6-knee.

[98] N. Hamilton, W. Weimar and K. Luttgens, Kinesiology: Scientific Basis of Human Motion, 9th ed., McGraw-Hill Higher Education, 1996.

[99] CrossFit, “Movement About Joints, Part 7: The Ankle,” 25 10 2020. [Online]. Available: https://www.crossfit.com/essentials/movement-about-joints-part-7-the-ankle.

[100] CrossFit, “Movement About Joints, Part 5: The Hip,” 25 10 2020. [Online]. Available: https://www.crossfit.com/essentials/movement-about-joints-part-5-the-hip.

[101] CrossFit, “Movement About Joints, Part 1: The Shoulder,” 25 10 2020. [Online]. Available: https://www.crossfit.com/essentials/movement-about-joints-part-1-shoulder.

[102] CrossFit, “Movement About Joints, Part 2: The Elbow,” 25 10 2020. [Online]. Available: https://www.crossfit.com/essentials/movement-about-joints-part-2-the-elbow.

[103] CrossFit, “Movement About Joints, Part 8: The Vertebral Column,” 25 10 2020. [Online]. Available: https://www.crossfit.com/essentials/movement-about-joints-part-8-the-spine.

[104] InvenSense, “MPU-6000 and MPU-6050 Product Specification Revision 3.4,” *Datasheet Revision 3.4,* pp. 1-52, 2013.

[105] InvenSense Inc, “MPU-6000 and MPU-6050 Register Map and Descriptions,” 2013. [Online]. Available: https://invensense.tdk.com/wp-content/uploads/2015/02/MPU-6000-Register-Map1.pdf. [Accessed 03 11 2020].

[106] L. Stitches, “E-tectiles for Education,” 08 08 2018. [Online]. Available: http://spiratronics.com/data/B503.pdf.

[107] pro-POWER, “Solderable Enamelled Copper Wire,” Farnell, 2015.

[108] Flexivolt, “RS Pro Yellow, Unscreened Single Core Equipment Wire,” RS Pro, 2018.

[109] Sheldahl, “Novaclad CF G2600,” Multek, Northfield, 2014.

[110] F. Semiconductor, “K20 Sub-Family,” Freescale Semiconductor, 2012.

[111] B. M. Heravi, “Accessible Routes from Crowdsourced Cloud Services,” 08 08 2018. [Online]. Available: https://www.arccs.org/.

[112] dialog-Semiconductor, “DA14580 Low Power Bluetooth Smart SOC,” Dialog-Semiconductor, 2015.

[113] B. Sensortec, “BMX055 Small, versatile 9-axis sensor module,” Bosch, 2013.

[114] Y. Lu, “Unsupervised Learning for Image Classification,” University of Helsinki, Helsinki, 2015.

[115] A. Navlani, “Support Vector Machines,” 25 10 2020. [Online]. Available: https://www.datacamp.com/community/tutorials/svm-classification-scikit-learn-python.

[116] hefinioranrhys, “Support Vector Machines with the mlr package,” 25 10 2020. [Online]. Available: https://www.r-bloggers.com/2019/10/support-vector-machines-with-the-mlr-package/.

[117] “Decision Tree Classification Algorithm,” 25 10 2020. [Online]. Available: https://www.javatpoint.com/machine-learning-decision-tree-classification-algorithm.

[118] B. Ma, Y. Su and F. Jurie, “BiCov: A novel image representation for person re-identification and face verification,” *BMVC 2012 - Electronic Proceedings of the British Machine Vision Conference 2012,* 2012.

[119] M. Koestinger, M. Hirzer, P. Wohlhart, P. M. Roth and H. Bischof, “Large scale metric learning from equivalence constraints,” *2012 IEEE conference on computer vision and pattern recognition,* pp. 2288-2295, 2012.

[120] S. M. Rajesh, S. Bhargava, S. B and K. Sivanathan, “Mission planning and waypoint navigation of a micro quad copter by selectable GPS co-ordinates,” *International Journal,* vol. 4, no. 4, pp. 2-11, 2014.

[121] R. Vemulapalli, F. Arrate and R. Chellappa, “R3DG features: Relative 3D geometry-based skeletal representations for human action recognition,” *Computer Vision and Image Understanding,* vol. 152, pp. 155-166, 2016.

[122] Y. Tong, Y. Wang, Z. Zhu and Q. Ji, “Robust facial feature tracking under varying face pose and facial expression,” *Pattern Recognition,* vol. 40, no. 11, pp. 3195-3208, 2007.

[123] L. Mündermann, S. Corazza and T. P. Andriacchi, “The evolution of methods for the capture of human movement leading to markerless motion capture for biomechanical applications,” *Journal of NeuroEngineering and Rehabilitation,* vol. 3, no. 1, pp. 1-11, 2006.

[124] C. D. Manning, H. Schütze and P. Raghavan, Introduction to information retrieval, Cambridge university press, 2008.

[125] L.-Y. Hu, M.-W. Huang, S.-W. Ke and C.-F. Tsai, “The distance function effect on k-nearest neighbor classification for medical datasets,” *SpringerPlus,* vol. 5, no. 1, pp. 1-9, 2016.

[126] B. G. Batchelor, Pattern recognition. Ideas in practice, Springer US, 1978.

[127] R. S. Michalski, R. E. Stepp and E. Diday, “A recent advance in data analysis: Clustering objects into classes characterized by conjunctive concepts,” in *Progress in pattern recognition*, Elsevier, 1981, pp. 33-56.

[128] P. P. Ippolito, “SVM: Feature Selection and Kernels,” 27 10 2020. [Online]. Available: https://towardsdatascience.com/svm-feature-selection-and-kernels-840781cc1a6c.

[129] S. A. Schurr, “TWO-DIMENSIONAL VIDEO ANALYSIS IS COMPARABLE TO 3D MOTION CAPTURE IN LOWER EXTREMITY MOVEMENT ASSESSMENT,” *Int J Sports Phys Ther,* pp. 163-172, 2017.

[130] H. Loose, “Gait patterns in standard scenarios: Using Xsens MTw inertial measurement units,” in *2015 16th International Conference on Research and Education in Mechatronics (REM).* , Bochum, 2015.

[131] O. Bischoff, N. Heidmann, J. Rust and S. Paul, “Design and implementation of an ultrasonic localization system for wireless sensor networks using angle-of-arrival and distance measurement,” *Procedia Engineering,* vol. 47, pp. 953-956, 2012.

[132] N. B. Schuler, M. J. Bey, J. T. Shearn and D. L. Butler, “Evaluation of an electromagnetic position tracking device for measuring in vivo, dynamic joint kinematics,” *Journal of Biomechanics,* vol. 38, no. 10, pp. 2113-2117, 2005.

[133] Mikus, “Deinterlacing - Wikipedia,” [Online]. Available: https://commons.wikimedia.org/wiki/File:Interlaced_video_frame_(car_wheel).jpg. [Accessed 13 11 2020].

[134] DataVedas, “HOLDOUT CROSS-VALIDATION,” [Online]. Available: https://www.datavedas.com/holdout-cross-validation/. [Accessed 13 11 2020].

[135] “Cross-validation: evaluating estimator performance,” Scikit-learn developers, [Online]. Available: https://scikit-learn.org/stable/modules/cross_validation.html. [Accessed 13 11 2020].

[136] L. Reveret, S. Chapelle, F. Quaine and P. Legreneur, “3D Motion Analysis of Speed Climbing Performance,” Front Psychol, 2020.

[137] Crystal Market Research Ltd., “The State of Indoor Climbing in the UK Quantatitative Market Research Findings,” Association of British Climbing Walls, Bristol, 2019.

[138] W. Group, “Speed Climbing Walls,” Walltopia, [Online]. Available: https://www.walltopia.com/en/climbing-walls/speed-walls. [Accessed 15 11 2020].

[139] A. I. Cuesta-Vargas, A. Galán-Mercant and J. M. Williams, “The use of inertial sensors system for human motion analysis,” *Physical Therapy Reviews,* vol. 15, no. 6, pp. 462-473, 2010.

[140] “Xsens - 3D Motion Tracking Technology,” [Online]. Available: https://www.xsens.com/. [Accessed 24 10 2020].

[141] “SMART-DX,” [Online]. Available: https://www.btsbioengineering.com/products/smart-dx/. [Accessed 24 10 2020].

[142] “WELCOME TO SHAOLIN TEMPLE UK,” [Online]. Available: https://shaolintempleuk.org/. [Accessed 27 10 202].

[143] T. Maruyama, M. Tada, A. Sawatome and Y. Endo, “Constraint-based real-time full-body motion-capture using inertial measurement units,” *2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC),* pp. 4298-4303, 2018.

10 APPENDIX A

10.1 HUMAN JOINT LIMITS

Diarthrodial joints, such as the knee and hip, have the primary function to facilitate human movement and locomotion. They are characterised by their large degree of articulating motion.

The diarthrodial joints are categorised by their rotational degrees of freedom and in the case of the spine, translational freedom. The number and axis of the degrees of freedom is determined by the geometry of the articulating joint. The range of motion of the joint is also affected by the surrounding soft tissue, such as muscles and ligaments.

10.1.1 Mechanical joint review

The following mechanical joints are approximations of the diarthrodial joints; the shapes of the joints are more complicated and deformation of the cartilage at the joint can result in additional degrees of motion.

Hinge Joint

The hinge joint has one rotational degree of freedom and zero translational degrees of freedom, refer to Figure 64. Examples of the hinge joint are the joint between the Humerous and Ulna of the elbow, and the joint between the Femur and Tibia of the knee.

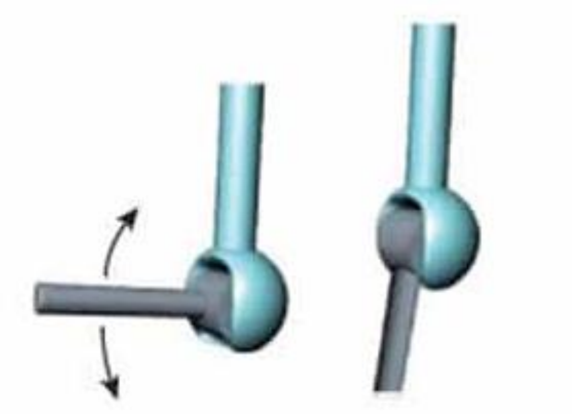


Figure 64 A 3D model of a basic hinge joint with its one rotational degree of freedom indicated by the arrows. [96]

Rolling Joint

The rolling joint has one rotational degree of freedom and zero translational degrees of freedom, refer to Figure 65. Examples of the rolling joint are the joints on either end of the Radius and Ulna, allowing pronation of the forearm.

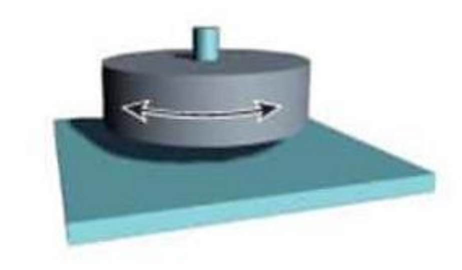


Figure 65 A 3D model of a basic rolling joint with its one rotational degree of freedom indicated by the arrows. [96]

Saddle Joint

The saddle joint has two rotational degrees of freedom and zero translational degrees of freedom, refer to Figure 66. Examples of the saddle joint are the Sternoclavicular joint in the shoulder and the joint between the Carpus and Metacarpal of the thumb, allowing the thumb to be apposable.

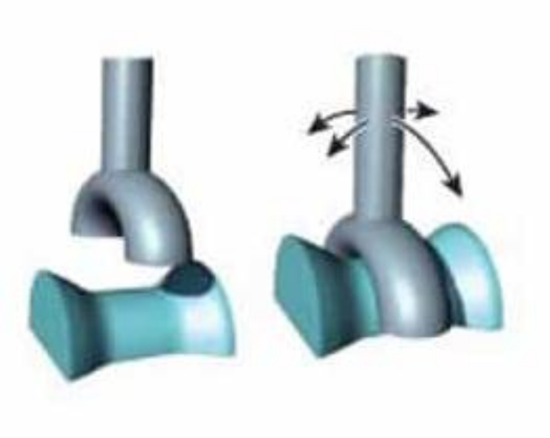


Figure 66 A 3D model of a basic saddle joint with its two rotational degrees of freedom indicated by the pair of arrows. [96]

Ellipsoidal Joint

The ellipsoidal joint has two rotational degrees of freedom and zero translational degrees of freedom, refer to Figure 67. However, unlike the saddle joint, one rotational degree of freedom has a much lower range of motion and is susceptible to soft tissue restrictions. An example of an ellipsoidal joint is the joint between the Radius and the Carpus of the wrist.

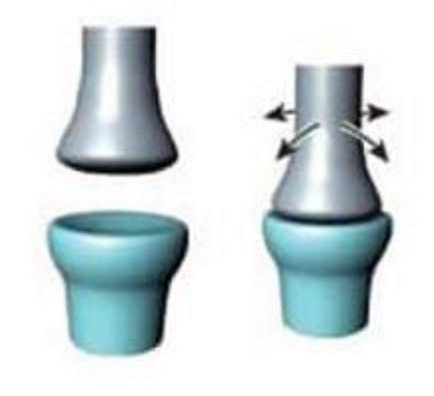


Figure 67 A 3D model of a basic ellipsoidal joint with its two rotational degrees of freedom indicated by the pair of arrows. [96]

Ball-and-socket Joint

The ball-and-socket joint has three rotational degrees of freedom and zero translational degrees of freedom, refer to Figure 68. The depth of the socket determines the range of motion. Examples of the ball and socket joints are the Glenohumeral joints of the shoulder and the hip.

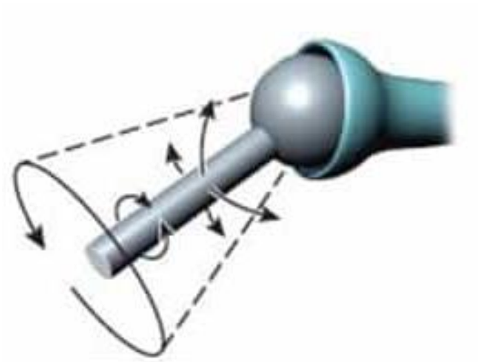


Figure 68 A 3D model of a basic ball-and-socket joint with its three rotational degrees of freedom indicated by the group of arrows. [96]

Planar Joint

The planar joint has one rotational degree of freedom and two translational degrees of freedom, refer to Figure 69. Examples of the planar joint are the Cervical spine joints between the vertebrae, and the Scapulothoracic joint of the shoulder.

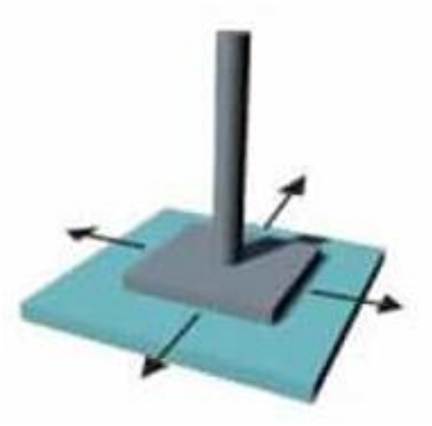


Figure 69 A 3D model of a basic planar joint with its two translational degrees of freedom indicated by the pair of arrows. [96]

The relevant joints for the Kung Fu dataset are the ankles, knees, hips, back/spine, shoulders, and elbows. The relevant joints when considering the Parkinson's dataset are Phalangeal joints of the fingers and the wrists.

10.1.2 Diarthrodial joint descriptions and range of motion

Knee joint

Although the knee is classified as a hinge joint, its structure resembles two condyloid joints lying off parallel side by side. The movements which occur at the knee joint are primarily flexion and extension. A slight rotation can occur when the knee is flex and not support weight.

The movements of flexion and extension at the knee are not a true hinge, this is because there is a slight rotation orthogonal to the hinge axis during the initial phase of flexion and the final phase of extension. Additionally, to this nonvoluntary rotation a further rotation is possible at the knee. When the leg has been flexed at the knee to a right angle and beyond it is possible to rotate the lower leg at the knee.

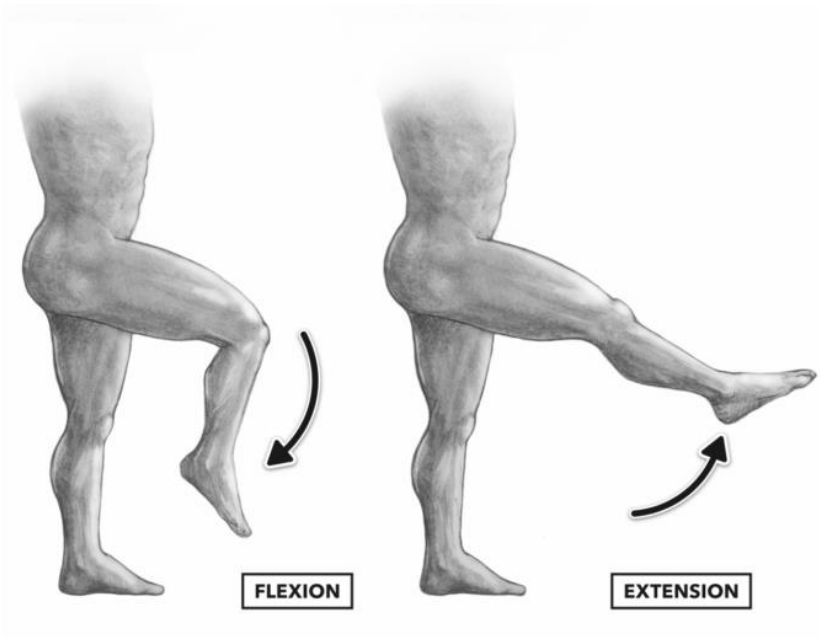


Figure 70 A simple diagram indicating the flexion and extension movement of the knee joint. [97]

Range of motion (ROM) of the knee for flexion/extension [98]			
150	120	140	130

Ankle joint

The foot is made up of the ankle, the tarsals and the toe joints. The ankle is the point of interest for this project. The ankle is a hinge joint with its axis described as near frontal-horizontal meaning when flexing the ankle, the foot moves up and down with little vertical rotation.

The movements of the ankle are known as dorsiflexion and plantarflexion. Dorsiflexion is the forward-upward motion resulting in the toes being pulled backwards closer to the lower leg. Plantarflexion is the forward-downward motion resulting in the toe pointing away from the lower leg.

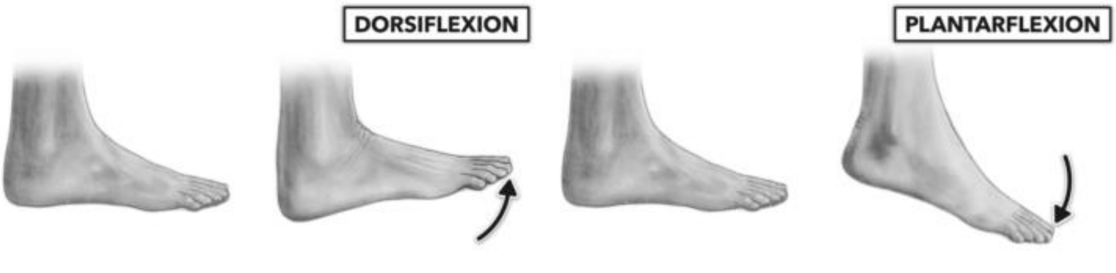


Figure 71 A simple diagram indicating the dorsiflexion and plantarflexion movements of the ankle joint. [99]

Range of motion (ROM) of the ankle for dorsiflexion and plantarflexion [98]				
Dorsiflexion	30	15	20	20
Plantarflexion	20	45	45	50

Hip Joint

The hip joint, a typical ball-and-socket, is formed by the articulation of the spherical head (ball) of the femur with the deep cup-shaped (socket) acetabulum.

The hip being a ball-and-socket joint has three pairs of rotational movement: flexion/extension, abduction/adduction, and outward/inward rotation.

Flexion: the forward movement of the thigh in the sagittal plane. If the knee is flexed the movement is restricted by the flexibility of the hamstring often resulting in reduced rotation. Extension is the returning motion from flexion. Hyperextension is the backward motion of the thigh on the sagittal plane.

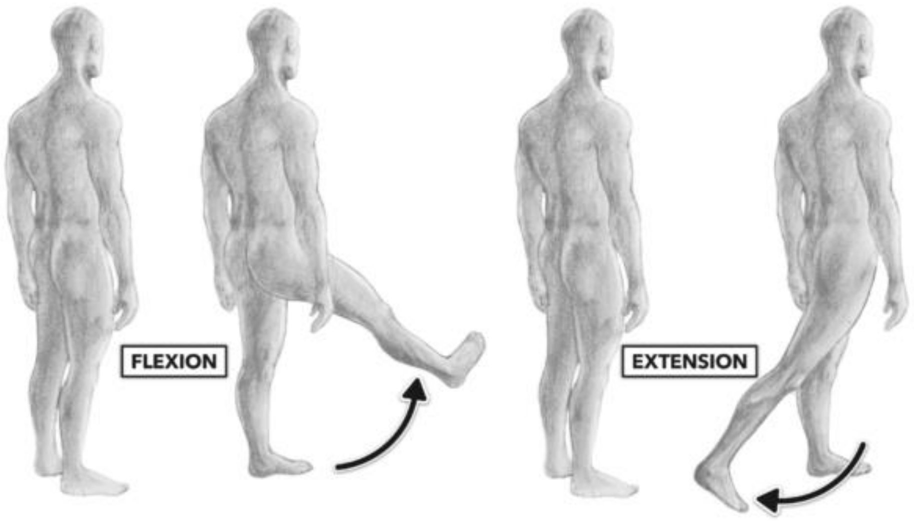


Figure 72 A simple diagram indicating the flexion and extension movement of the hip joint. [100]

Range of motion (ROM) of the hip joint for flexion/extension and hyperextension [98]				
Flexion	100	120	125	120
Hyperextension	30	10	10	30

Abduction: the sideward movement of the thigh moving away from the midline of the body. A greater range of motion is possible when the femur is rotated outward. Adduction is the returning motion from abduction. Hyperadduction is possible if the other leg is moved out of the way.

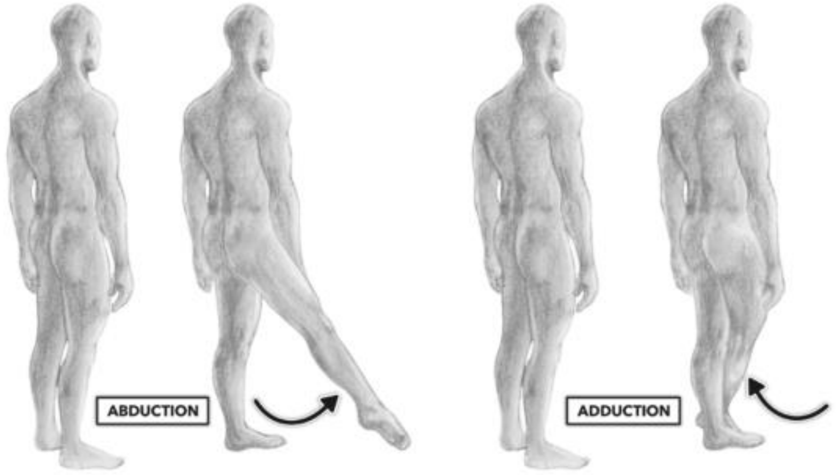


Figure 73 A simple diagram indicating the abduction and adduction movement of the hip joint. [100]

Range of motion (ROM) of the hip joint for abduction and adduction [98]				
Abduction	40	45	45	45
Hyperadduction	20	10	25	

External rotation: the rotation of the thigh along the longitudinal axis of the femur so that the knee turns outward. Internal rotation is the opposite rotation resulting the knee to turn inward.

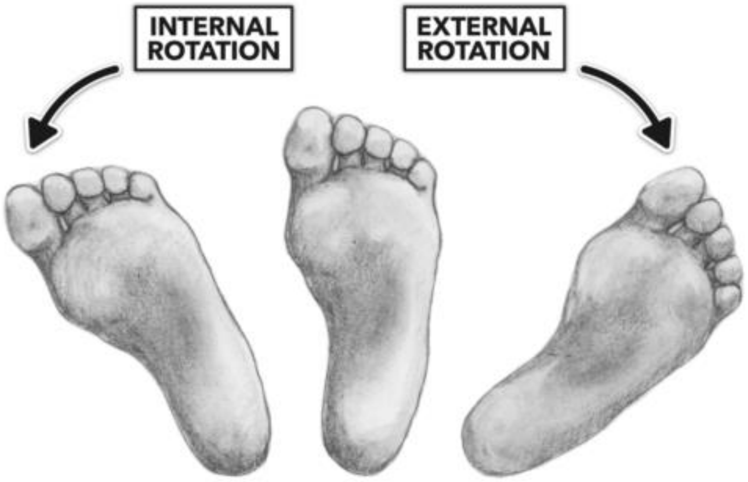


Figure 74 A simple diagram indicating the internal and external rotation of the hip joint. [100]

Range of motion (ROM) of the extended hip joint for internal and external rotation [98]				
External rotation	50	45	45	45
Internal rotation	40	35	45	45

Shoulder Joint

The shoulder joint is a ball-and-socket joint however the looseness of the capsule, permitting 2-5 centimetres of separation of the joint, allows greater mobility than a standard ball-and-socket joint.

The shoulder being a ball-and-socket joint has three pairs of rotational movement: flexion/extension, abduction/adduction, and outward/inward rotation.

Flexion: the forward upward motion of the arm on the sagittal plane, once the arm exceeds 180 degrees the motion becomes hyperflexion. Extension is the returning motion from flexion until the arm points downward. Hyperextension is the backward upward movement of the arm on the sagittal plane.

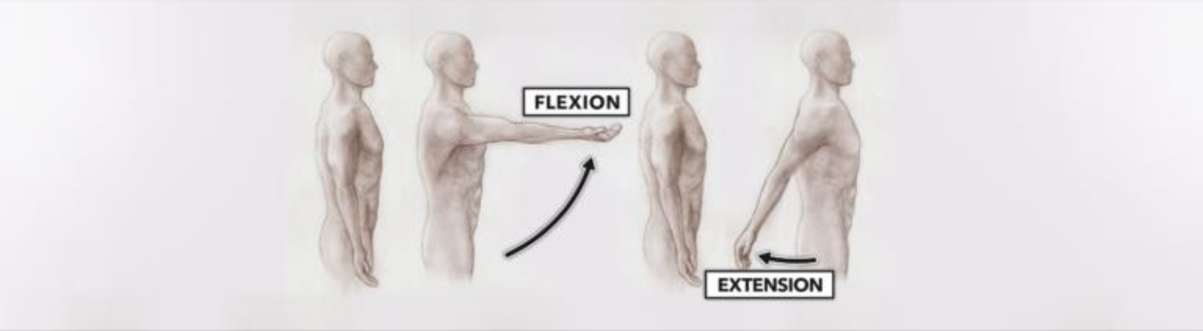


Figure 75 A simple diagram indicating the flexion and extension movement of the shoulder joint. [101]

Range of motion (ROM) of the shoulder joint for flexion/extension and hyperextension [98]				
Flexion	180	170	130	130
Hyperextension	50	30	80	60

Abduction: the sideward upward movement of the arm moving away from the midline of the body, Figure 76. Adduction is the returning motion of the arm from abduction.

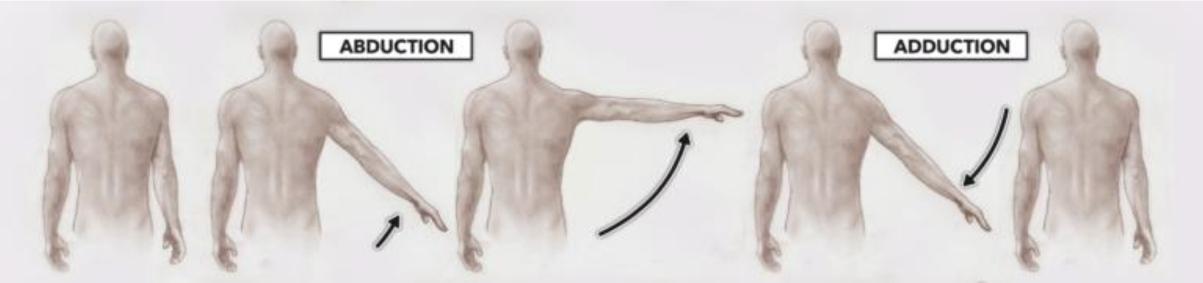


Figure 76 A simple diagram indicating the abduction and adduction movement of the shoulder joint. [101]

Range of motion (ROM) for the shoulder joint for the abduction/adduction movement [98]			
180	170	180	180

External rotation: the rotation of the upper arm along the longitudinal axis of the humerous so that when the arm is pointing downward the anterior aspect turns laterally, Figure 77. Internal rotation is the rotation resulting in anterior aspect turning medially.

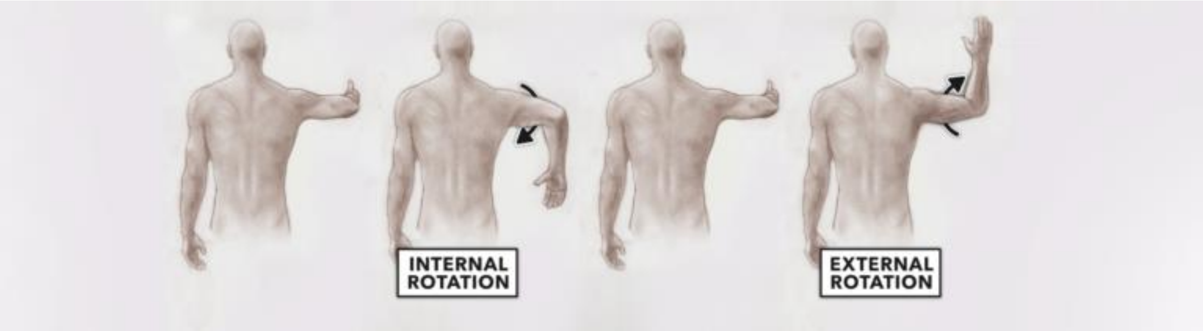


Figure 77 A simple diagram indicating the external and internal rotation of the shoulder joint. [101]

Range of motion (ROM) of the shoulder joint (with abducted upper arm) for internal and external rotation [98]				
External rotation	90	90	70	90
Internal rotation	90	90	70	90

Elbow/Forearm Joint

The elbow is a combination of two joints: a hinge joint and a rolling joint. For the sake of this explanation the hinge joint will be considered the elbow joint and the rolling joint will be considered the forearm joint as it is located on either end of the forearm. As the elbow joint is a hinge joint it has one pair of movements: flexion and extension.

Flexion: With the elbow facing backwards, the forward upward movement of the forearm in the sagittal plane, Figure 78. Extension is the returning motion from flexion. Hyperextension is possible for some individuals.

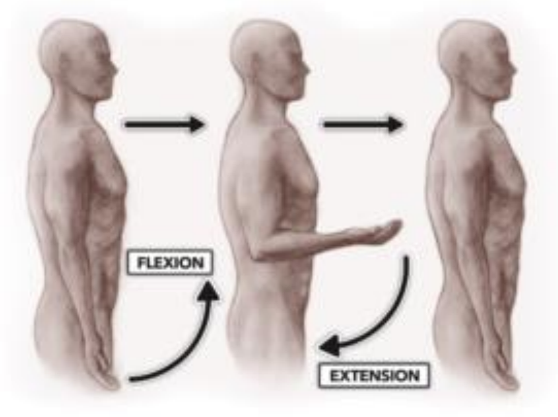


Figure 78 A simple diagram indicating the flexion and extension movement of the elbow joint. [102]

Range of motion (ROM) of the elbow joint for the flexion/extension movement [98]			
140	145	145	145

As the forearm joint is a rolling joint it has one pair of movements: pronation and supination.

Pronation: the rotation along the longitudinal axis of the forearm resulting in the palm to turn medially, Figure 79. Supination is the opposite rotation turning the palm laterally. When the elbow is fully extended pronation and supination causes the upper arm to rotate with the forearm.

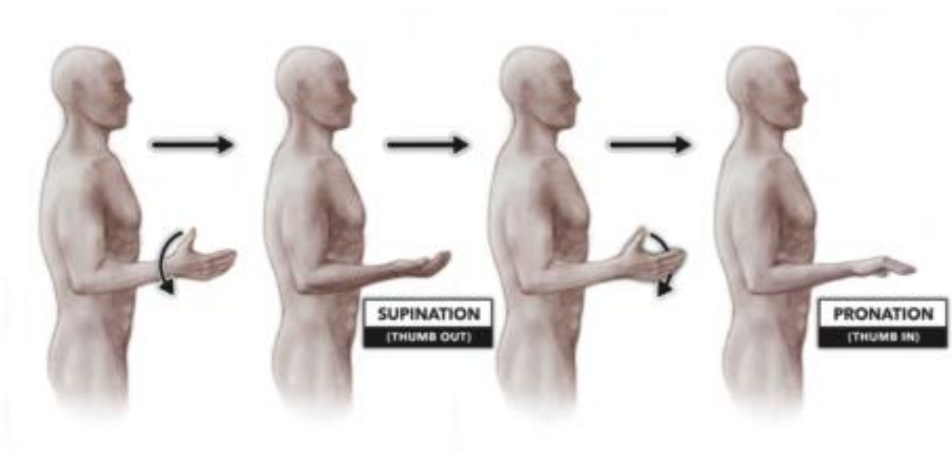


Figure 79 A simple diagram indicating supination and pronation of the elbow joint. [102]

Range of motion (ROM) of the elbow joint for the supination and pronation movement [98]				
Supination	80	85	90	90
Pronation	80	90	90	80

Back/Spine Joint

The movement of the spinal column can be represented as a ball-and-socket joint despite anatomically being several planer joints. Like a standard ball-and-socket joint, the spinal column has 3 groups of movement: flexion/extension/hyperextension, lateral flexion and rotation.

Flexion: the forward downward bending in the sagittal plane, Figure 80. Extension is the returning motion from flexion. Hyperextension is the backward downward movement in the sagittal plane.

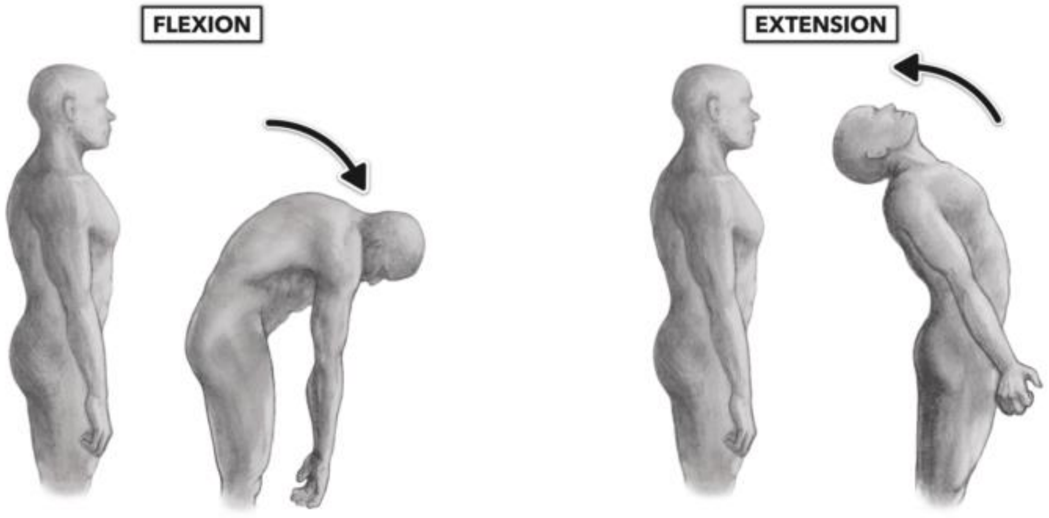


Figure 80 A simple diagram indicating the flexion and extension movement of the spinal column. [103]

Range of motion (ROM) of the lumbar-thoracic spine for the flexion/extension and hyperextension movement [98]		
Flexion	50	45
Hyperextension	25	35

Range of motion (ROM) of the cervical spine for the flexion/extension and hyperextension movement [98]		
Flexion	60	40
Hyperextension	75	40

Lateral flexion: the sideward bending movement in the frontal plane about the sagittal-horizontal axis, Figure 81.

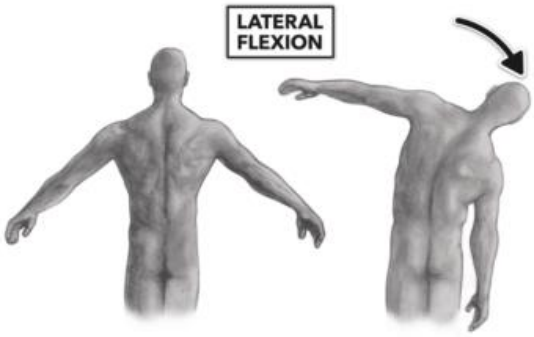


Figure 81 A simple diagram indicating the lateral flexion movement of the spinal column. [103]

Range of motion (ROM) of the lumbar-thoracic spine for the lateral flexion movement [98]	
25	30
Range of motion (ROM) of the cervical spine for the lateral flexion movement	
45	45

Rotation: the rotatory movement of the torso in the horizontal plane about the vertical axis, Figure 82.

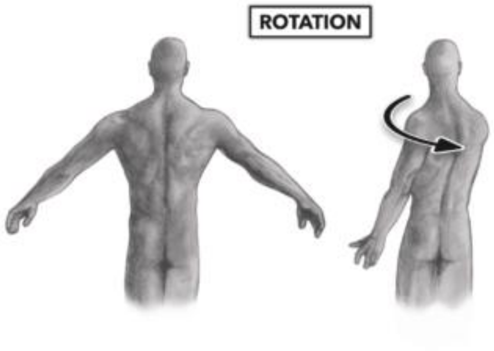


Figure 82 A simple diagram indicating the rotation of the spinal column about its longitudinal axis. [103]

Range of motion (ROM) of the lumbar-thoracic spine for rotation about the longitudinal axis of the spine [98]	
30	45
Range of motion (ROM) of the cervical spine for rotation about the longitudinal axis of the spine	
80	50

APPENDIX B

10.2 FIRMWARE/HARDWARE

10.2.1 IMU parameter considerations

To begin the project several MEMS sensors were compared to determine the most applicable sensor for testing. Due to variety of possible applications for the sensors it was decided that the demo sensor should be adjustable allowing a dynamic range of measurement values. Another important specification that was considered when choosing the first MEMS sensor was the available serial protocols. Power consumption and price was also considered however as this would only be for a prototype these were not the priority when choosing the sensor.

10.2.2 I2C vs SPI

I2C and SPI are both bus protocols that allow short-distance, serial data transfer. Both protocols are commonly used in electronic devices like smartphones, TV's and laptops to control peripherals like power management chips, input devices and DACs. Essentially both protocols achieve the same outcome, however due to the architecture of each the process of reaching this outcome can be very different. In terms of this project the three most important aspects of the protocols are: the bus topology, the speed, and their unique features.

10.2.2.1 Bus topology

I2C requires 2 lines, SDA and SCL, for any number of devices, however, requires several bits to address the chosen slave device, reduces the number of bits available for data. SPI requires 4 lines, SCLK, MISO, MOSI and SS, for a single device and for each further device added requires an additional SS to select that device. The total number of lines required for SPI is $n + 3$, where n is the number of slave devices. Therefore, it is clear when using many slave devices, SPI routing will become very complicated and will require a microprocessor with many I/O pins or an additional router.

If the reduced word length of the I2C protocol is not a problem, in terms of bus topology, I2C is clearly a more suitable protocol for this project.

10.2.2.2 Speed

I2C utilises half-duplex transmission, whereas SPI is full duplex, meaning technically the rate of data transmission can be doubled using SPI. However, this is only true if the system is constantly sending and receiving data from the slave devices, which in many cases, including this project, is not necessary. Due to it being physically connected separately to each slave device the SPI protocol does not have a speed limit of its own, often going beyond 10Mbps depending on the master’s clock speed. On the other hand, I2C uses slave addresses to multiplex the communication to each slave device. This process limits the speed of data transmission dramatically and without including computationally heavy I/O buffers the top speed is approximately 1Mbps.

10.2.2.3 Features

I2C offers multi-master conflict handling, which is almost always essential for systems which require multi masters controlling the same slaves. This project may require multi-masters when it comes to setting up a smart power network. It is possible to handle multiple masters in SPI; however, this requires several delays and buffers, meaning both computationally expensive and slow. SPI is often considered a simpler protocol to implement in terms of firmware due to the 4 lines each having a distinct role in the data transmission. This is also the reason for SPI being flexible to extensions and variations, as the entire data word can be used freely by the firmware.

10.2.3 MPU-6000/6050

The MPU-6050 was chosen due to it matching all the requirements specified as well as having extensive material on both its usage and its physical layout. The relevant parameters of the MPU-6000 are seen below.

<table>
<tr><th colspan="3">MPU-6000/MPU-6050</th></tr>
<tr><th>I2C</th><th colspan="2">SPI</th></tr>
<tr><td>Yes</td><td colspan="2">Yes (MPU-6000 only)</td></tr>
<tr><th></th><th>Range</th><th>Resolution /LSB</th></tr>
<tr><th>Accelerometer</th><td>±2g</td><td>0.98 mg</td></tr>
<tr><td></td><td>±4g</td><td>1.95 mg</td></tr>
<tr><td></td><td>±8g</td><td>3.91 mg</td></tr>
<tr><td></td><td>±16g</td><td>7.81 mg</td></tr>
<tr><th>Gyroscope</th><td>±250 °/s</td><td>61.0 m°/s</td></tr>
<tr><td></td><td>±500 °/s</td><td>30.5 m°/s</td></tr>
<tr><td></td><td>±1000 °/s</td><td>15.3 m°/s</td></tr>
<tr><td></td><td>±2000 °/s</td><td>7.6 m°/s</td></tr>
</table>

Table 17 Important parameters of the MPU-6000/6050 [104]

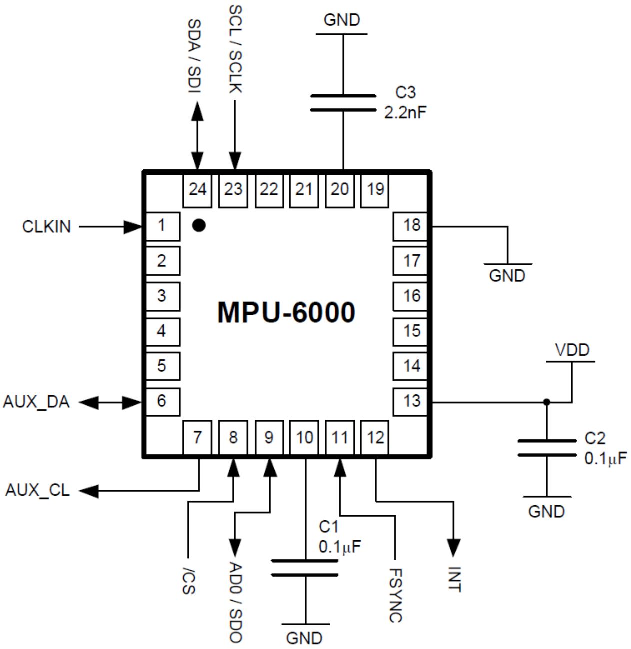


Figure 83 Operating circuit for the MPU-6000 [104]

Another fortunate advantage this module had over others is that it was widely available on a breakout board, Figure 84. This is very convenient when writing up the firmware for the sensor as the breakout board can be simply plugged into an Arduino, meaning during the debugging process hardware debugging may be overlooked as assumed functional. The IC on the breakout board is the MPU-6050, which is another model with identical specifications as the MPU-6000 apart from it only being setup for the I2C protocol. It was therefore decided that to begin with the firmware would be written for the I2C protocol using Arduino language.

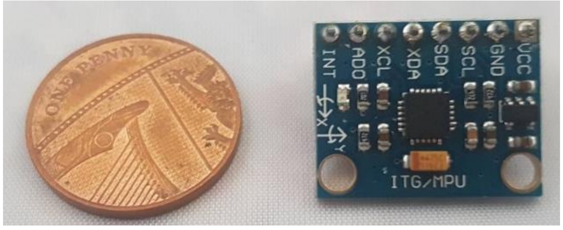


Figure 84 MPU-6050 breakout board next to a penny

10.2.4 Firmware

As mentioned previously the MPU-6000 has extensive material regarding its physical structure as well as its registers. By having all the registers described in as much detail as "MPU-6000 and MPU-6050 Register Map and Descriptions Revision 4.2" the task of writing firmware for the sensor is simplified significantly [105]. Once all the necessary registers are identified, a header file is constructed listing these registers, and a CPP file is written to establish some essential methods, such as *whoami()*. The loop of the Arduino code is summarised below, indicating the main blocks processed within the firmware.

10. Initialise an instance specifying parameters

The main difference, other than the registers used, when initialising I2C over SPI is that the parameter required by I2C is the slave address whereas the parameter required by SPI is the pins allocated to the slave select (SS).

20. Call whoami

Whoami is method often used in programming which returns a type of identification which allows the code to determine whether it has successfully flashed to a device. In this case, the type of identification is the slave address of the MPU-6000/6050, 104.

30. Select coordinate system (e.g. Euler angles, quaternion, yaw/pitch/roll)

The default output of the accelerometer and gyroscope is raw data that means very little without knowing the correct conversion algorithm as described in the datasheet [105]. As this algorithm was computationally cheap it was decided to convert the raw data on the microprocessor before transmitting the data to the serial monitor. Once it became clear what the raw data represented several expressions were created so that the system could be adjusted to display the results in multiple coordinate systems. The available coordinate systems are quaternion, Euler angles, yaw/pitch/roll, real acceleration, and world acceleration.

Quaternions use 4 dimensions to represent 3D reflection, rotation and scaling, where the first three dimensions relate to x, y, and z and the last dimension represents the transformation.

Euler angles represents the orientation of a rigid body using three angles.

Yaw/pitch/roll is another coordinate system which represents a position with only angles, however the sequence of the applied rotations is in a different order to Euler angles.

Real acceleration displays the acceleration values not including gravity.

World acceleration displays the acceleration values including acceleration due to gravity. This can be used to calibrate the accelerometer.

40. Set accelerometer scale

Used to set the range of measurable accelerations, with options of: ±2g, ±4g, ±8g, and ±16g. As mentioned previously the resolution of the measurement decreases as the range increases.

50. Set gyroscope scale

Used to set the range of measurable angular velocities, with options of ±250 ^{o}/s, ±500 ^{o}/s, ±1000 ^{o}/s, and ±2000 ^{o}/s. As mentioned previously the resolution of the measurement decreases as the range increases.

60. Read accelerometer calibration

Used to read the initial accelerometer measurements to determine how well they have been calibrated. Ideally this method would return a zero value for all three axes.

70. Read accelerometer

Reads the raw data from the allocated register and converts it to the coordinate system selected.

80. Read gyroscope

Reads the raw data from the allocated register and converts it to the coordinate system selected.

90. Send results to serial monitor

This block of code sends the converted data to the serial monitor, where can be read real time to determine the success of the firmware. Once the firmware was successfully debugged this block of code was adjusted so that the data was saved to csv file.

10.2.5 Jig demonstration

To assess the viability of the MPU-6050 using the firmware described above, a demo was designed. The idea behind this demo was to come up with a method which can evaluate the static performance of the sensors without additional hardware and with only minimal post processing. This demo was used to assess the sensors in their simplest form.

A jig was designed with multiple joints with an MPU-6050 sensor attached to each joint, as seen in Figure 85. This design was 3D printed using ABS material. The jig was constructed to simulate the simplest motion expected from a limb, folding on a single plane. Due to the MPU-6050 sensors only having two unique slave addresses available only two sensors can work in conjunction, limiting the jig to two joints. The sensors on the jig were wired up to an Arduino mega via a breadboard.

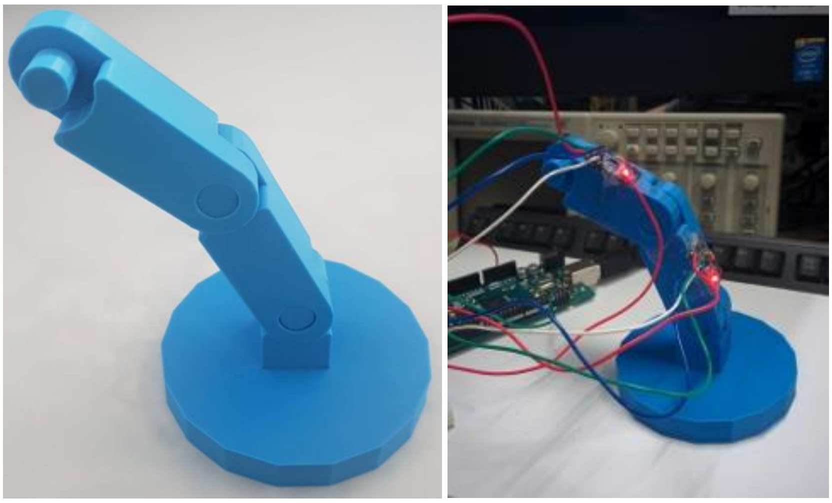


Figure 85 3D printed jig prototype with 2 joints. Left: Jig without sensors, right: jig with sensors

Static measurements were used to evaluate the accuracy of the sensors without any post processing. The test procedure started with the sensor's z-axis angled upwards against gravity. As the data was collected, the sensor was rotated by 90° until was perpendicular to gravity. The system continued to take several measurements from the sensor once the rotation was complete determining the converged orientation value. Averages were then taken of these final stationary measurements. These trials are performed 10 times using 6 different sensors to obtain a relatively accurate evaluation of the sensors' performance. The trials are repeated not using the calibration block of code in the firmware for comparison reasons. The uncertainties stated refer to the value of two standard deviations for each type of measurement.

Rotation Direction	Calibrated	Uncalibrated
α	89.0° ± 0.8°	103° ± 1°
β	89.2° ± 0.8°	82° ± 7°

Table 18 Static accuracy results obtained from jig demo for each axis

α and β are defined as the rotations about the x and y axis of the sensors. The large error seen for the un-calibrated measurements can be accounted for misalignment error within the sensor. The low uncertainty for α's uncalibrated measurements is due to the misalignment error being constant for all 6 sensors used, whereas the high uncertainty for β's uncalibrated measurements is due to the direction of the misalignment being non-constant for the 6 sensors used.

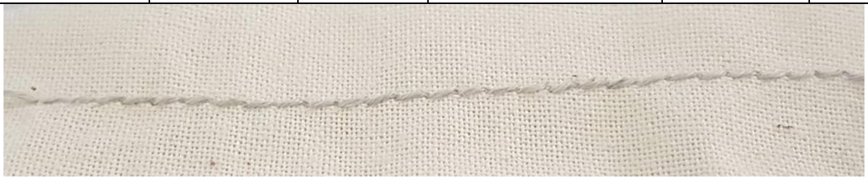
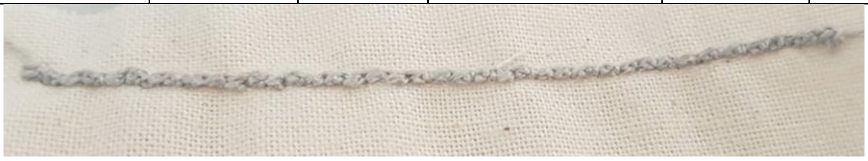
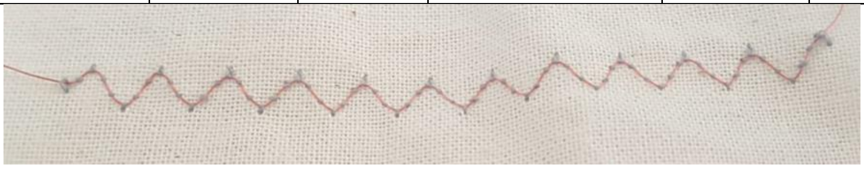
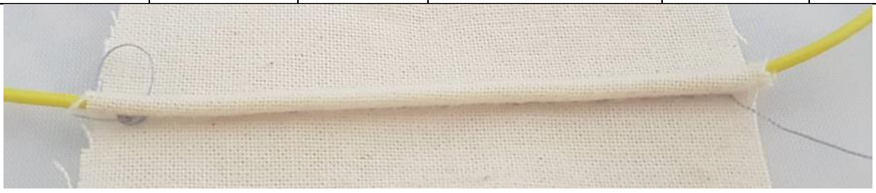
10.2.6 Hardware modifications

With the MPU-6050 module tested without any post- or pre-processing, the next step was to modify the hardware available to suit the aims of the project. These modifications fall into two categories: circuit design and connections.

10.2.6.1 Connections

In wearable technology the interconnections between sensor points are very important as they determine the usability of the system. In this project the connections must be structured in such a way that they do not hinder the dancer's movement whilst still being tough enough to remain intact throughout the dancer's routine. During prototyping 4 main connection types were tested: conductive thread, enamel wire, PVC clad wire and flexible ribbon cable.

The table below summarises the results of the tests done on the connection types. Note that due to it acting significantly different to a single stitch, multiple stitches of conductive thread have been taken as a separate connection type.

	Resistivity ohm*m	Tensile strength	Flexibility	Insulation	Gap requirements per bus	Ease of fitting
Single conductive thread [106]	2.87	High	High	None	4-5mm	Sewing machine
 *Figure 86 Single conductive thread stitching*						
Multiple conductive thread [106]	2.96	High	High	None	5-6mm	Sewing machine
 *Figure 87 Multiple conductive thread stitching*						
Enamel wire [107]	0.0031	Medium	High	Full	5-6mm	Sewing machine
 *Figure 88 Enamelled wire sew into fabric in a zigzag stitch*						
PVC clad wire [108]	0.0032	High	High	Full	2mm	Piping, cannot be done retrospectively
 *Figure 89 PVC clad wire piped into fabric*						
Flexible ribbon cable [109]	4.56E-07	Low	High, however only in one direction as twists can cause tearing	Bottom side only	70um	Fabric to plastic adhesive

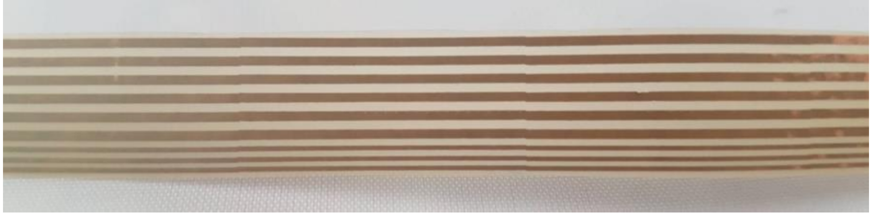


Figure 90 Flexible ribbon cable with multiple 1mm and 0.5mm tracks

Figure 91 shows an example of the complex stitches that can be achieved with the enamelled wire if the setting is correct. The wire must be wound and placed on the under bobbin of the sewing machine; the tension of the machine must be slightly high than default; and the stitch must have hooks approximately every 1-2mm.

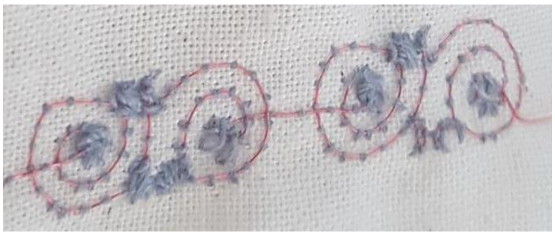


Figure 91 Example of complex stitching achieved using enamelled wire

A key factor to consider when selecting a connection type is the frequency response. This is particularly important when considering the SPI protocol due to its potentially high operating frequency. The operating frequency of the SPI protocol is limited by the clock frequency of the microprocessor, which for most microprocessors is 16MHz. It is clear from the graph below that none of the connection types will be able to achieve such a high operating speed, however for the most part an operating frequency of 2-3 MHz is achievable. Except for the multiple stitch conductive thread connection type which falls off around 500kHz. Another test worth considering is the BER of each connection type, however this only seemed to be an issue at frequencies beyond expect operating frequencies (approx. 20MHz).

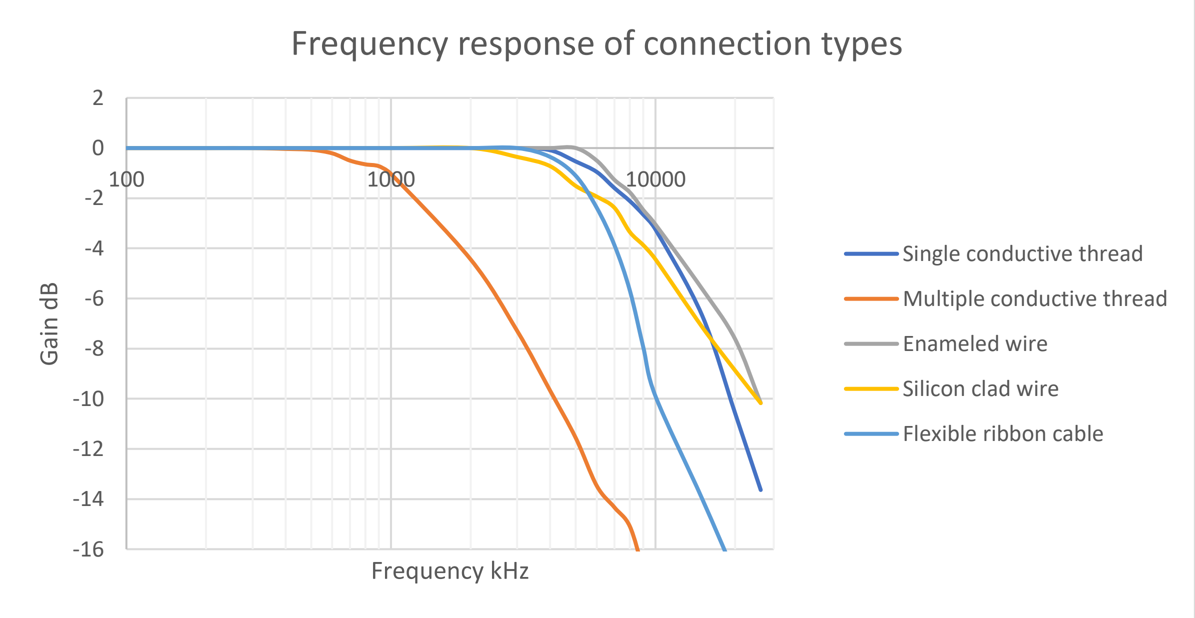


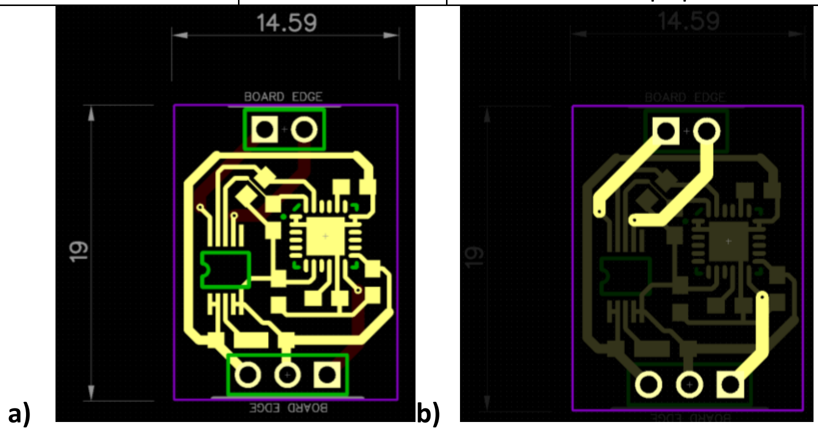
Figure 92 Frequency response of the five connection types discussed previously

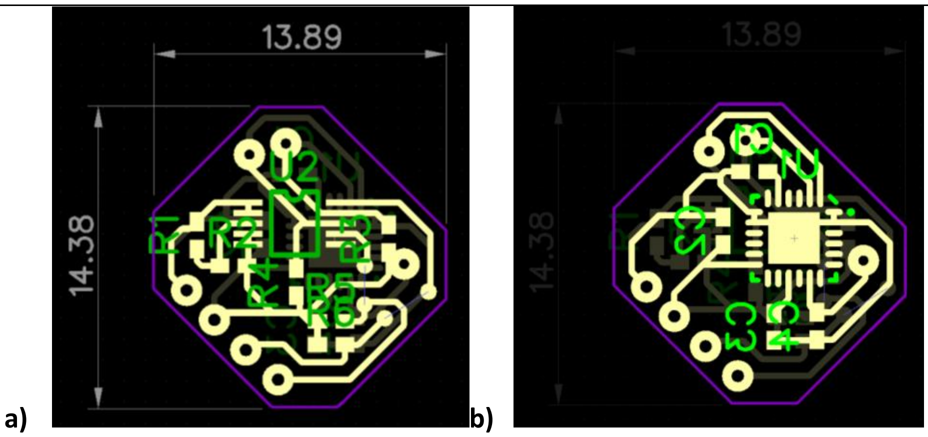
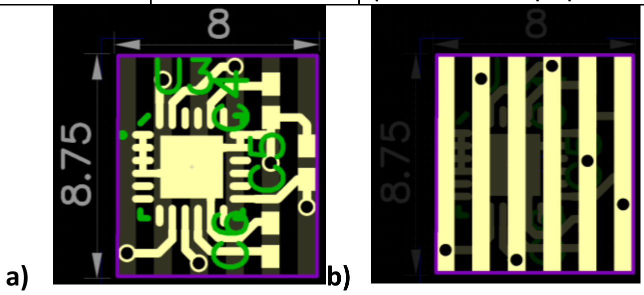
10.2.6.2 Circuit design

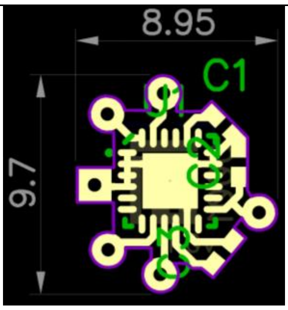
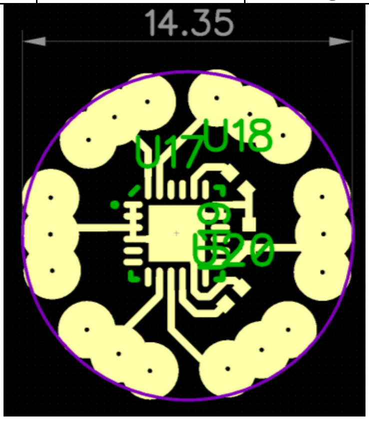
Being a simple digital circuit, the MPU-6050 module does not have a lot of room for improvement, however the following aspects were considered improvements on the design for this project:

- Ground the XCL, XDA and AD0 pins as these are used for external components (such a compass) which is not necessary here. By grounding these pins interference can be eliminated
- Reduce the distance from the bias components and the IC to reduce noise
- Remove the voltage regulator as most microprocessors can output 3.3V, and keeping the regulator would increase noise present in the circuit
- Remove the LED and associated components as these are not needed and can cause interference in the circuit
- Include a slave address translator so more than 2 devices can be used in conjunction (I2C only)

The table below shows the PCB layouts of the designs printed. Each design had a different purpose in mind resulting in various shapes, sizes and materials used. The main considerations made were: the serial protocol, the desired connection type and the purpose of the board. The purposes varied from breadboard testing to use in demonstrations.

Serial Protocol	Connection type	Material	Design benefit
I2C	PVC clad wire, enamel wire	1.6mm FR-4	Larger design for testing on breadboards. ICs on the same side allowing a reference for tests to determine any IC interference. Includes interrupt pin connection.
 *Figure 93 'I2C breadboard test PCB' design a) top layer b) bottom layer*			
I2C	PVC clad wire, enamel wire	0.8mm FR-4	Compact design allowing it to be smaller than the original MPU-6050 module while still including the address translator. Overlapping of the ICs does not cause any significant interference when compared to the previous circuit design. This was determined by comparing bit error rates over a range of frequencies (100kHz-400kHz). Interrupt pin has been grounded.

 13.89 14.38 U2 R1 R2 R3 R4 R5 R6 a) 13.89 14.38 U1 C1 C3 C4 b) *Figure 94 'I2C compact PCB' design a) top layer b) bottom layer*			
SPI	Flexible ribbon cable	0.8mm FR-4	The compact design and the connection type for this PCB makes it ideal for retrofitting around close joints. This is due to the continuity along a single axis of the connection layer, allowing the module to be placed anywhere along the ribbon cable. The connection can be made either by securing the module on the ribbon cable mechanically or by soldering the module in place. Interrupt pin has been grounded.
 8 8.75 U3 C4 C3 C5 C6 a) 8 8.75 b) *Figure 95 SPI 'spied' PCB design a) top layer b) bottom layer*			
SPI	Conductive thread, enamel wire	0.8mm FR-4	By having the board outline cut around the connection pads a groove is created allowing conductive thread to tie around the pad forming a connection. The smaller pads seem to give a more consistent connection with the conductive thread than the standard sized pads. The PVC clad wire is unable to fit in the smaller pads. Interrupt pin has been grounded.

 *Figure 96 SPI 'spider' PCB design*			
SPI	Conductive thread	18-25-18 (um) Nanoclad flexible PCB	This circuit has been specifically designed for conductive thread. Using thin flexible PCB material allows the conductive thread to be sewn into the circuit. By looping between the three pads a strong mechanical connection can be made, increased by the pinching caused by piercing the material. Interrupt pin has been grounded.
 *Figure 97 SPI 'round' PCB design*			

10.2.7 Glove demonstration

To demonstrate the use of the flexible ribbon cable connectors and the corresponding module a simple motion capture glove was designed. Originally the purpose of this system was to fully capture the motion of all the fingers and the wrist, however due to the number of GPIO pins available on the microprocessor used only 3 fingers were hooked up. The Teensy 3.2 was used due to its several GPIO pins available for SPI communication and the availability of floating-point calculations. [110]

The design of the module allowed them to be place precisely between the knuckle joints of the 3 fingers used, however this was only possible when used in conjunction with the flexible ribbon cable which turned out a poor wearable connection type. Once it became clear that the method for fastening the flexible ribbon cable to the glove would result in extensive independent motion of the sensors, it was decided to monitor 'taps' rather than full motion.

To determine the performance of the system in terms of measuring 'taps', an arbitrary sequence of 100 taps were recorded and tallied. This was then compared to the system's measurements and the number of true positives, false positives, true negatives and false negatives were determined. This was repeated 10 times with a new arbitrary sequence each time and an average was determined.

	Predicted label						
		Thumb tap	Thumb no tap	Index tap	Index no tap	Middle tap	Middle no tap
True labels	Tap	88	12	97	3	95	5
	No tap	24	176	37	163	33	167
Overall accuracy		0.88		0.867		0.873	

Table 19 Statistically results obtained from glove demonstration 'tap' test

From the results above, despite currently lacking the accuracy requirements of a click-based controller, with some adjustments such an application could be possible. Note that during the final tests of the glove the flexible ribbon cable began to tear around the turn of the thumb. Considering that the tests applied to the glove only took a few hours, in terms of robustness the flexible ribbon cable does not meet the requirements of the project.

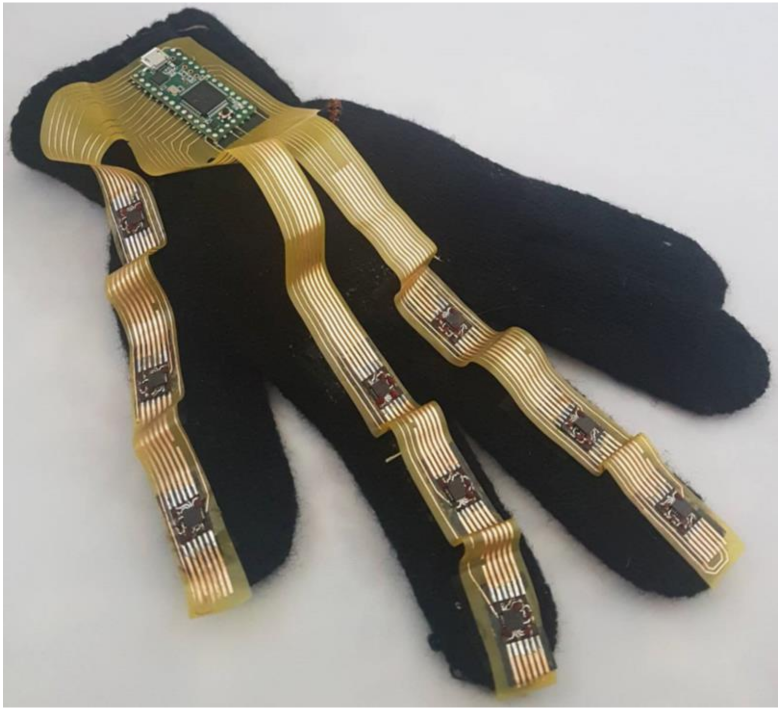


Figure 98 Glove with three fingers setup for motion capturing, using 'striped' module and flexible ribbon cable connectors. Teensy 3.2 microprocessor used at the wrist of the glove

10.3 SOFTWARE (POST PROCESSING)

Following on from the background research the 3 variations of the Kalman filter were applied to the raw data collected from the MEMS sensors. This was done by storing the data on a CSV file and using MATLAB to read and apply the filter algorithms to this data; when considering a single sensor only the accelerometer and gyroscope is required, however when multiple sensors are used the timing data is also needed. The flow diagrams below show the steps taken in the MATLAB code to implement the algorithms. Due to their similarities, the Kalman filter and the Extended Kalman filter have been placed on the same flow diagram, with their differences noted of the diagram. The Unscented Kalman filter being more complicated to implement requires a more detailed flow diagram.

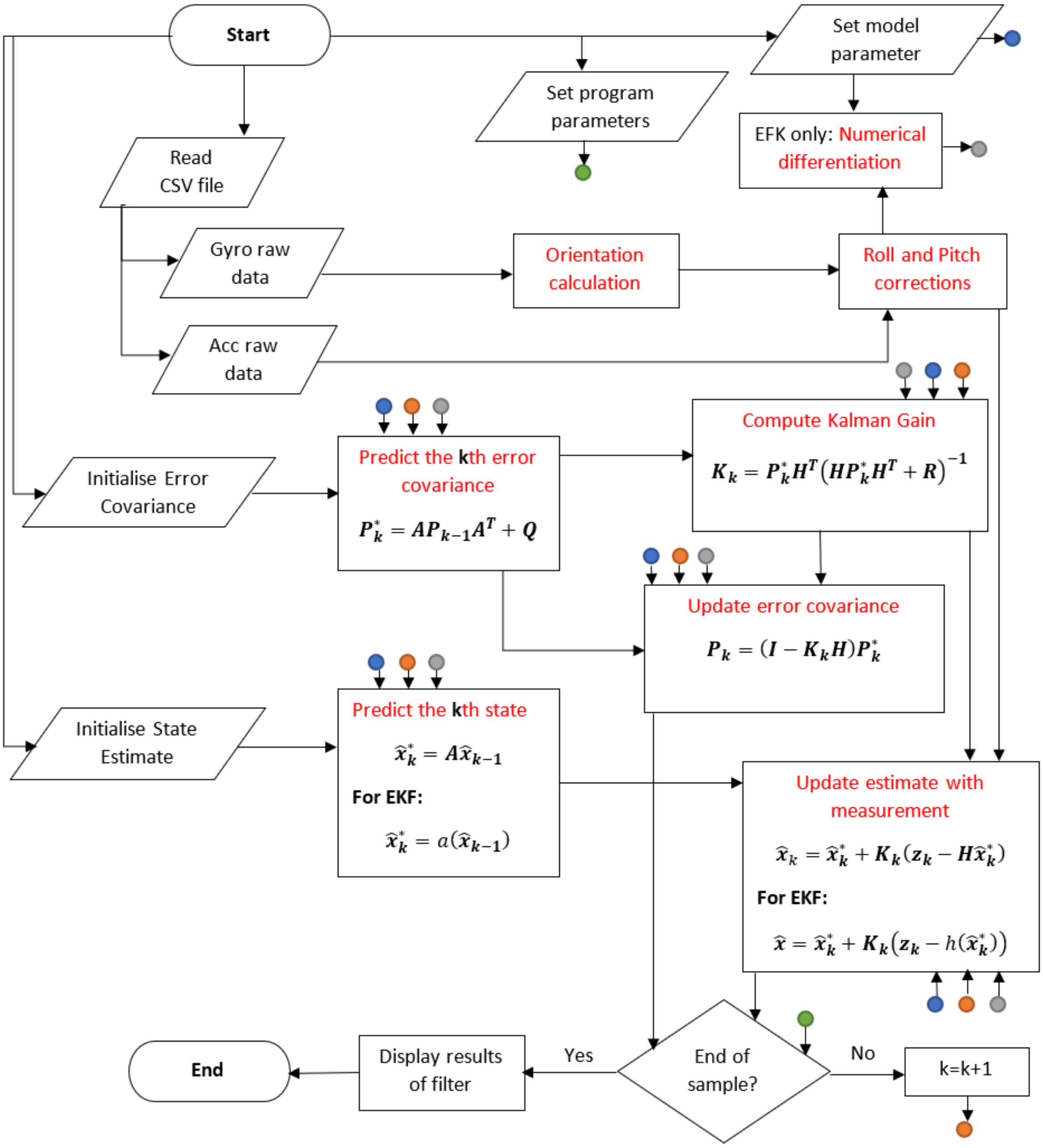

Start
Set model parameter
Set program parameters
EFK only: Numerical differentiation
Read CSV file
Gyro raw data
Orientation calculation
Roll and Pitch corrections
Acc raw data
Compute Kalman Gain
$K_k = P_k^* H^T (H P_k^* H^T + R)^{-1}$
Predict the kth error covariance
$P_k^* = A P_{k-1} A^T + Q$
Initialise Error Covariance
Update error covariance
$P_k = (I - K_k H) P_k^*$
Predict the kth state
$\hat{x}_k^* = A \hat{x}_{k-1}$
For EKF:
$\hat{x}_k^* = a(\hat{x}_{k-1})$
Initialise State Estimate
Update estimate with measurement
$\hat{x}_k = \hat{x}_k^* + K_k (z_k - H \hat{x}_k^*)$
For EKF:
$\hat{x} = \hat{x}_k^* + K_k (z_k - h(\hat{x}_k^*))$
End
Display results of filter
Yes
End of sample?
No
k=k+1

Figure 99 Flow diagram for standard Kalman filter and EFK. the difference between the two filters has been highlighted (For EKF and EFK only)

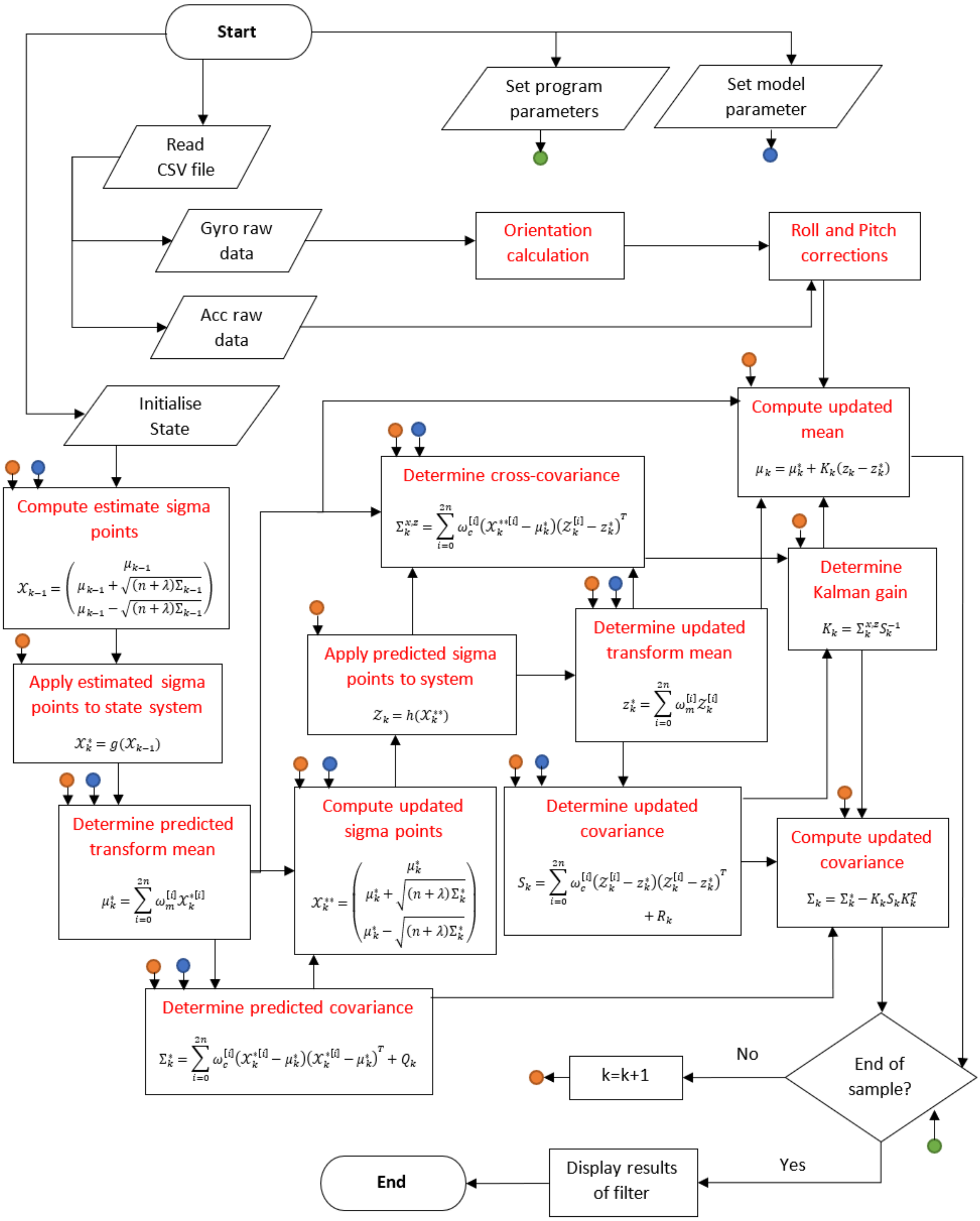

Start
Set program parameters
Set model parameter
Read CSV file
Gyro raw data
Orientation calculation
Roll and Pitch corrections
Acc raw data
Initialise State
Compute updated mean
$\mu_k = \mu_k^* + K_k(z_k - z_k^*)$
Determine cross-covariance
$\Sigma_k^{x,z} = \sum_{i=0}^{2n} \omega_c^{[i]}(\mathcal{X}_k^{**[i]} - \mu_k^*)(Z_k^{[i]} - z_k^*)^T$
Compute estimate sigma points
$\mathcal{X}_{k-1} = \begin{pmatrix} \mu_{k-1} \\ \mu_{k-1} + \sqrt{(n+\lambda)\Sigma_{k-1}} \\ \mu_{k-1} - \sqrt{(n+\lambda)\Sigma_{k-1}} \end{pmatrix}$
Determine Kalman gain
$K_k = \Sigma_k^{x,z} S_k^{-1}$
Apply estimated sigma points to state system
$\mathcal{X}_k^* = g(\mathcal{X}_{k-1})$
Apply predicted sigma points to system
$Z_k = h(\mathcal{X}_k^{**})$
Determine updated transform mean
$z_k^* = \sum_{i=0}^{2n} \omega_m^{[i]} Z_k^{[i]}$
Determine predicted transform mean
$\mu_k^* = \sum_{i=0}^{2n} \omega_m^{[i]} \mathcal{X}_k^{*[i]}$
Compute updated sigma points
$\mathcal{X}_k^{**} = \begin{pmatrix} \mu_k^* \\ \mu_k^* + \sqrt{(n+\lambda)\Sigma_k^*} \\ \mu_k^* - \sqrt{(n+\lambda)\Sigma_k^*} \end{pmatrix}$
Determine updated covariance
$S_k = \sum_{i=0}^{2n} \omega_c^{[i]}(Z_k^{[i]} - z_k^*)(Z_k^{[i]} - z_k^*)^T + R_k$
Compute updated covariance
$\Sigma_k = \Sigma_k^* - K_k S_k K_k^T$
Determine predicted covariance
$\Sigma_k^* = \sum_{i=0}^{2n} \omega_c^{[i]}(\mathcal{X}_k^{*[i]} - \mu_k)(\mathcal{X}_k^{*[i]} - \mu_k^*)^T + Q_k$
No
k=k+1
End of sample?
Yes
Display results of filter
End

Figure 100 Flow diagram for UKF

The algorithms described above will convert the raw data of the MEMS sensor into highly accurate angle measurements in real time. Once the algorithm is complete there are many ways to display the angle data, such as, angle vs time, position vs time, position vs frequency, dynamic 3D model.

10.3.1 Software simulation test

To compare the performance of the three digital filters a non-linear function with a known noise was applied to each of the filters. Specifically, a sine wave with a period of 125s multiples by random

Gaussian noise with a scaling factor of 0.2. Note that the unit of the period is arbitrary as the sampling rate is scaled to fit the unit used. The function was sampled at 10 per second which is the minimum expected sampling rate of the ARCCS sensor. To compare the performance of the filters fully the results were plotted in both the time and frequency domain.

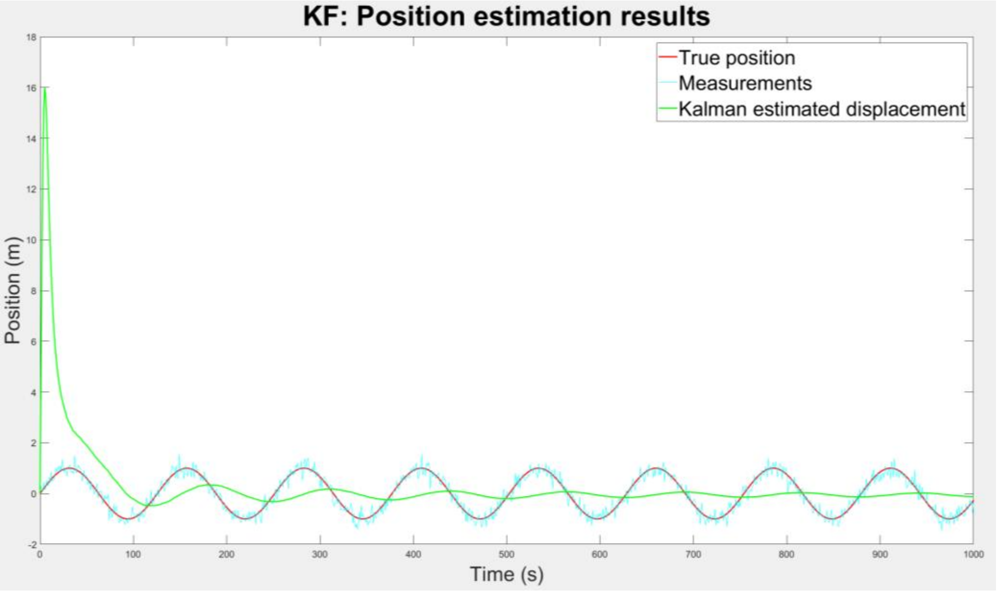


Figure 101 Position against time plot of the standard Kalman filter's estimated position

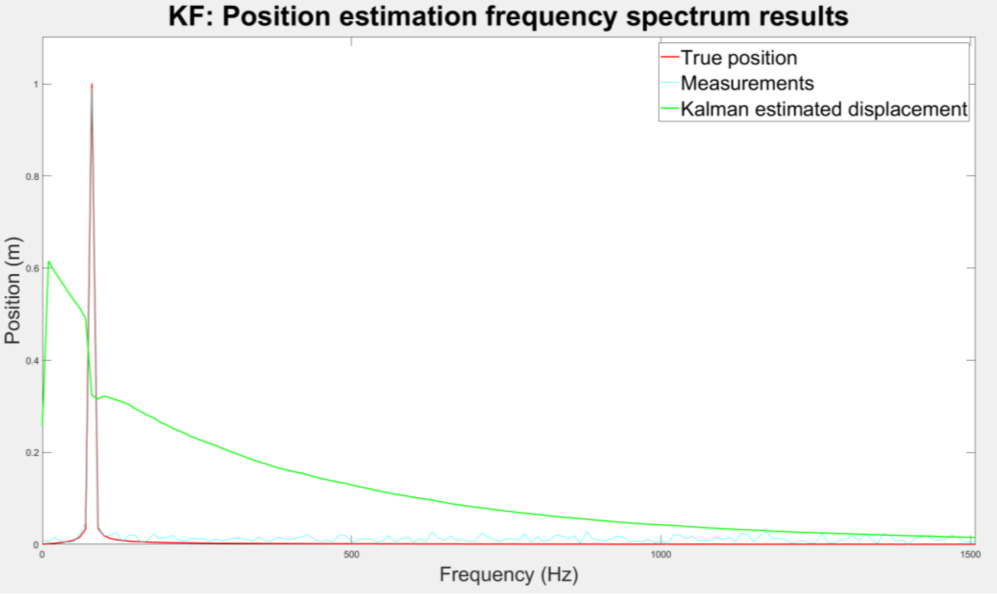


Figure 102 Position against frequency plot of the standard Kalman filter's estimated frequency spectrum

As expected from the filter design the standard Kalman filter has attempted to manipulate the function into a linear representation. To begin with this results in a steep incline as the filter tries to respond to the sudden change as the sine wave begins. The filter is slow to respond the next few seconds of the function and only begins to decline around 5 seconds in. As the filter assumes it is working with a linear function it puts more weight of previous estimated states than the measurements it is given. This can be seen clearly by the damping sine wave produced by the filter. As time goes on the filter’s results become more and more linear as the influence of the previous estimated state accumulates. This damping sine wave can be seen in the frequency domain as the square-like beginning. The rest of the frequency domain can be explained by the peak at the beginning of the filter's result. Using the raw measurements in this case would result in a better estimate than the estimate resulting from the filter.

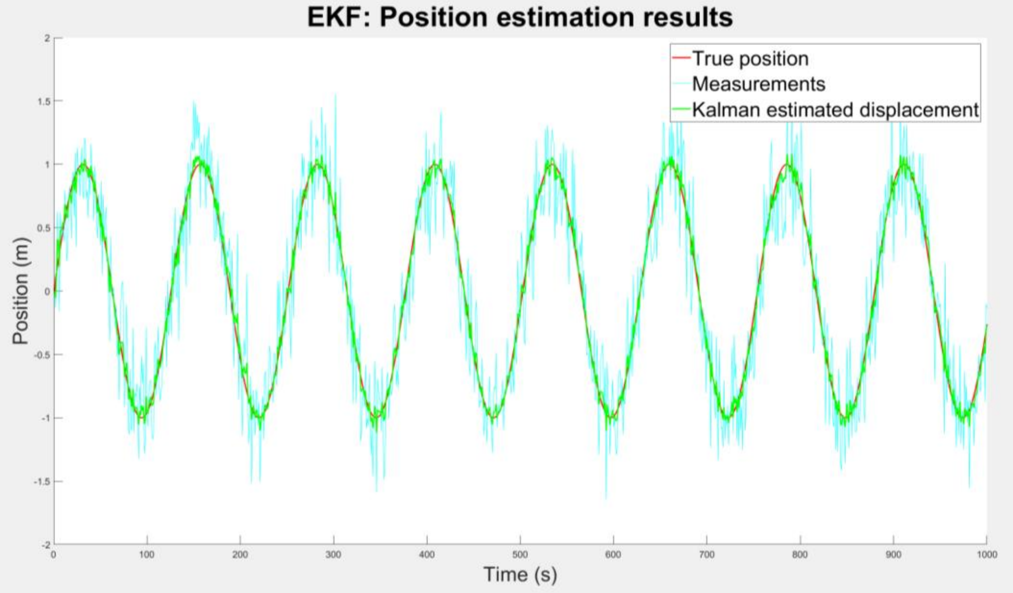


Figure 103 Position against time plot of the EKF's estimated position

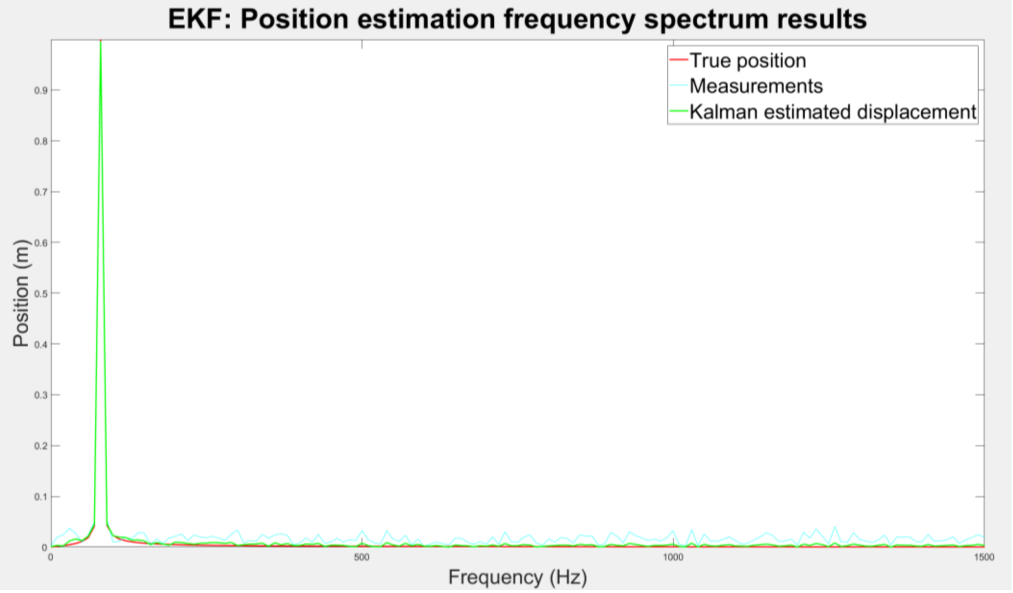


Figure 104 Position against frequency plot of the EKF's estimated frequency spectrum

Compared to the measurements given by the system the Extend Kalman filter has clearly given a more accurate estimation of the sine wave. Something worth noting is that the accuracy of estimation produced by the filter is constant. This is due its knowledge of the system, i.e. The differential of the system has been calculated. When looking at the frequency domain the filter has accurately estimated the correct frequency of the sin wave while minimising all other frequencies.

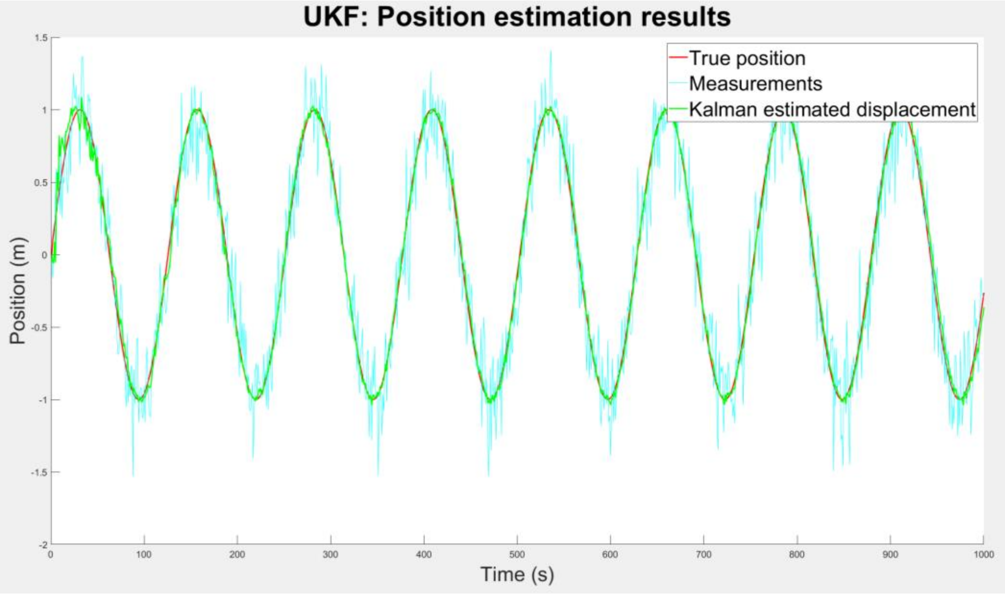


Figure 105 Position against time plot of the UKF's estimated position

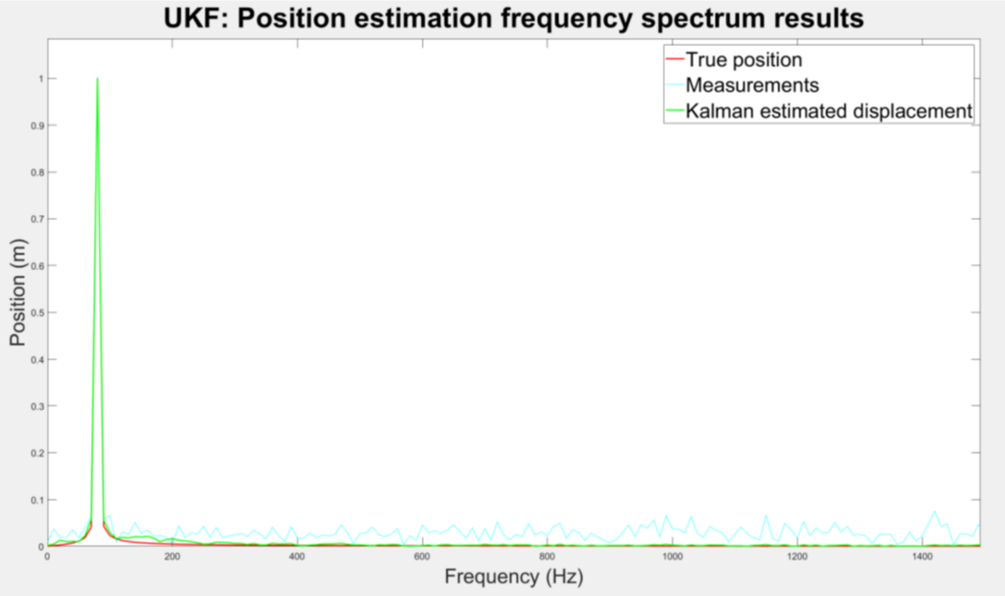


Figure 106 Position against frequency plot of the UKF's estimated frequency spectrum

As with the case of the Extend Kalman filter, the Unscented Kalman filter has given a more accurate estimation of the function than the measurements given by the system. From the position against time plot the filter is less accurate to begin with having a slight phase lag, but as time goes on its accuracy increases to beyond what is seen with the Extend Kalman filter. This can also be seen in the frequency plot. The filter again can determine the correct frequency of the function, however, has slightly more noise around this frequency, almost like a harmonic is present. Besides this harmonic the filter can reduce the noise around all other dramatically compared to the Extend Kalman filter.

10.3.2 ARCCS pendulum test

Working with the ARCCS group in UCL, the 3 algorithms were applied to the ARCCS device [111]. The device consists of a Smart Bluetooth system-on-chip (DA14580) [112], a Bosch Sensortec 9-axis Inertial sensor (BMX055) [113], micro SD-card storage and an onboard real-time clock (RTC). In terms of the range and resolution the BMX055 is identical to the inertial sensor in the MPU-6000. The SD card can store up to 8GB of IMU data, which equates several hours of measurements.

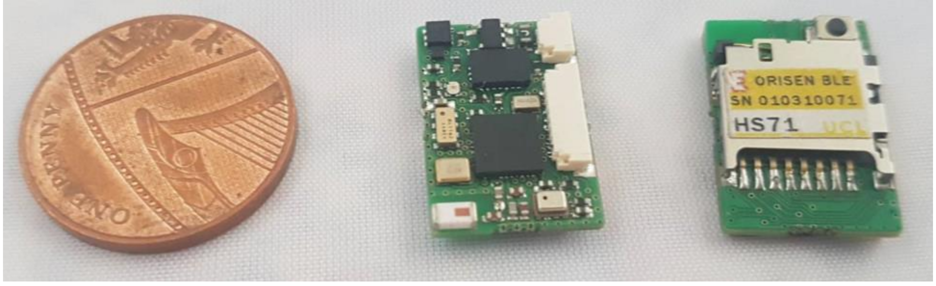


Figure 107 Top and bottom view of the ARCCS sensor device next to a penny for scale

To compare the performance of the simulated filer results to real world filtered measurements, an ARCCS sensor was mounted at the end of a pendulum. The pendulum was placed against a rough wall purposely creating a relatively high factor of noise. The pendulum was held at 60 degrees from its resting position for 5 secs before being released. Once released it was left to keep swinging until it came to a rest. This raw data was then applied to the three filters and results seen below were obtained.

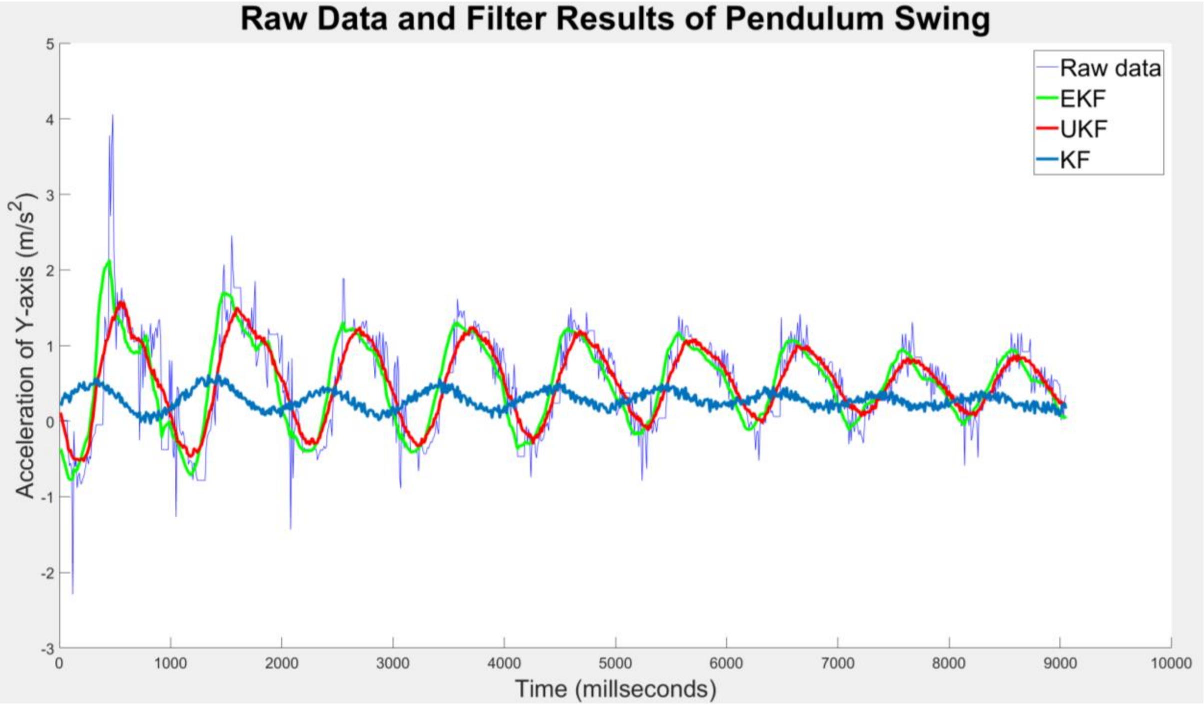


Figure 108 Accelerometer data and filtering results from ARCCS sensor mounted on a pendulum

Looking at these test results and comparing them to the simulated results a lot of similarities can be seen. In both cases the EKF and UKF have very similar results with the EKF following the noise a bit more. However due to there being no ideal data or golden standard it is not possible to say which of the filters have a better estimation. Once again, the standard Kalman filter has attempted to manipulate the data into a linear function, resulting in a phase lag and a major reduction in amplitude. The peak seen in the simulated Kalman filter is not present on the test Kalman filter results, this is mostly like due to the 5s delay before the pendulum is let loose. This delay makes the filter assume a linear function based solely on the noise of the sensor, which means when there is motioned introduced the filter is moving from that assumption rather than starting from no knowledge of the system. A difference between the simulated data and the recorded data is that the noise of the simulated has a constant factor, whereas with the recorded data the noise reduces as the velocity of the pendulum reduces and increases as the pendulum changes direction.