Queen Mary
University of London

Hand Gesture Recognition using a Low-Cost Sensor with Digital Signal Processing

By

Hussein Walugembe

Submitted in partial fulfilment of the requirements of the

Degree of Doctor of Philosophy

Supervisors: Dr Chris Phillips, Dr Jesus Requena-Carrion

School of Electronic Engineering & Computer Science

Queen Mary, University of London

March, 2022

Statement of Originality and Authorisation

I, Hussein Walugembe, declare that the research presented in this thesis is to the best of my knowledge and belief original and my own work, or that where it has been conducted jointly or cooperatively with others that this is appropriately acknowledged on the following page and my novel research contribution indicated as well. In addition, I acknowledge research papers that were published prior to writing this thesis.

I confirm that the material in this thesis has not been previously submitted, either in whole or part for a degree award at this or any other university.

I grant Queen Mary University of London right to archive, reproduce, distribute, and display this thesis in any forms including electronic format via any digital mechanisms supported by the university.

Future scientists are granted permission for individual research and non-profitable reproduction of this work for educational purposes. Any copying or publication of this thesis for commercial purposes or financial gain is not possible without my written consent.

Signature: Hussein Walugembe

Date: 01-March-2022

Publications

Hussein Walugembe, Chris Phillips, Jesus Requena-Carrion, and Tijana Timotijevic, "**Comparing Dynamic Hand Rehabilitation Gestures in Leap Motion Using Multi-dimensional Dynamic Time Warping,**" IEEE Sensors Journal, Vol. 21, No. 6, pp. 8002-8010, March 2021.

Hussein Walugembe, Chris Phillips, Jesus Requena-Carrion, and Tijana Timotijevic, "**Gesture Recognition in Leap Motion Using LDA and SVM**," In proceedings of IEEE International Conference on Computing, Electronics and Communications Engineering (IEEE iCCECE '19), pp. 56-61, London, UK, August 2019.

Hussein Walugembe, Chris Phillips, Jesus Requena-Carrion, and Tijana Timotijevic, "**Characterizing and Compensating for Errors in a Leap Motion using PCA**," In the proceedings of IEEE International Conference of Signal Processing and Information Security (IEEE ICSPIS '18), pp. 1-4, Dubai, UAE, 7-8, November 2018.

Abstract

Our research concerns a hand gesture recognition framework that makes use of a low cost "off-the-shelf" device. The device is a visual markerless sensor system called the Leap Motion controller (LM). However, before deploying the LM, we investigate its accuracy and limitations in measuring finger joint angles. We consider a user that flexes and extends all the fingers on the hand i.e. users with missing fingers are not considered in this research. In addition, we assume a user's hand does not shake and can maintain a required position for the duration of the experiments.

During finger joint angle error analysis, we conducted a series of experiments to assess the accuracy of the LM in terms of parameters such as elevation, lateral (side-to-side) positioning, forward-backward positioning, and rotation of the hand relative to the LM. We used an "artist's hand" placed above the LM. The artist's hand is more accurate than a human hand in performing static hand gestures as it can maintain a fixed position as long as is necessary. According to the results of the error analysis, we apply Principal Component Analysis (PCA) to the LM raw data to see whether it can compensate for these errors. Reasons for choosing PCA are described in Section 1.1. The experimental results show that the PCA is feasible, effective and can be applied such that accurate measurements can be obtained. Specifically, PCA was able to reduce AEs (Absolute Errors) by 37.5%, 28.3%, 33.0%, and 22.4% for the experimental results of elevation, lateral (side-to-side), forward-backward, and rotation, respectively.

Furthermore, we have applied machine learning techniques such as Linear Discriminant Analysis (LDA) and Support Vector Machines (SVMs). The reasons for choosing these techniques rather than others can be found in Section 1.1. These techniques help in recognising and classifying performed hand gestures. In addition, while classifying hand gestures, these techniques can learn about measurement errors and compensate for them. Experimental results show a significant benefit when applying LDA and SVM, yielding a performance accuracy above 88.0%, which is far better than the baseline performance of 67.1%. The baseline performance is the accuracy obtained when we directly observe and assign all the test samples to the gestures they supposedly represent. Further explanation of

the baseline performance can be found in the fourth paragraph of Section 5.2.3.

We also propose and evaluate the use of Multi-dimensional Dynamic Time Warping (MDTW) for simulating a comparison of dynamic hand gestures that would be performed by a patient relative to hand gestures that could be prepared by a physiotherapist. MDTW enables us to determine how similar or different a query dynamic hand gesture is to a reference one whilst filtering out unwanted sources of error resulting from positional, rotational or speed differences between the query and the reference actions. It produces a minimum-distance value of a warp path after aligning a query dynamic hand gesture with a reference one. A low minimum-distance value implies the two gestures being compared are similar and high minimum-distance value implies the two gestures vary to a greater extent.

When we deliberately compare a specific hand gesture with itself, we obtain a minimum-distance value of 0.0^{o} indicating the similarity is 100.0%. Furthermore, when we compare two closely similar hand gestures i.e. gesture 1 and gesture 4 as described in Section 6.1.4, a minimum-distance value of 35.9^{o} is obtained. However, when we compare two quite different gestures i.e. gesture 2 and gesture 3, a minimum-distance value of 248.5^{o} is obtained. Therefore, one can establish whether a user performs hand gestures satisfactorily or an adjustment is required based on the minimum-distance values of the warp paths.

Finally, we propose and implement PCA to investigate whether it is capable of improving the performance of LDA, SVM and MDTW. During PCA implementation, a feature vector that consists of the retained Principal Components (PCs) should be carefully selected. When we discard the first PC and retain the remainder as the feature vector, we obtain superior results where the performance of LDA, SVM and MDTW improves.

Acknowledgment

I would like to thank the Islamic Development Bank (IsDB) for the merit scholarship granted to me to pursue my PhD at Queen Mary University of London. IsDB has been paying the tuition fees for the whole PhD duration and support me with the monthly living costs for 3 and half years. In addition, they have facilitated some of the conferences I have attended and presented some of our research. I am really so grateful and may Allah reward them abundantly.

I extend my sincere thanks to my two supervisors Dr Chris Phillips and Dr Jesus Requena-Carrion for tireless guidance, tremendous advice, great support, and prompt feedback. I have really learnt from both of them. I also thank all the teachers who have trained me to acquire researcher development skills.

I thank my mum and dad for frequent support and encouragement. I thank my twin brother, Hassan for encouragement and spiritual support. We have been in the same struggle to attain PhD award at the same time. I thank my wife Ramlah and my twin daughters (Inam and Inaaya) for the support and courage they have given me during my final stage of the PhD study. Further, I thank all my family members for their love and support.

Finally, I thank my fellow students in the Networks research group and other PhD students who have supported me in one way or another. These students have been there for me in good and challenging moments throughout my PhD study.

Table of Contents

List of Figures

List of Tables

List of Acronyms

3D	Three Dimensions
ADL	Activities of Daily Living
AEs	Absolute Errors
API	Application Programming Interface
cm	centimetre
CMC	CarpoMetaCarpal
CI	Confidence Interval
DIP	Distal InterPhalangeal
DOF	Degree of Freedom
DTW	Dynamic Time Warping
EDR	Edit Distance on Real sequences
ERP	Edit distance with Real Penalty
fMRI	Functional Magnetic Resonance Imaging
HMM	Hidden Markov Model
IKF	Interval Kalman Filter
IMU	Inertial Measurement Unit
INS	Inertial Navigation System
IP	InterPhalangeal
IPF	Improved Particle Filter
IR	Infrared
LCSS	Longest Common SubSequence
LDA	Linear Discriminant Analysis

LEDs	Light-emitting diodes
LM	Leap Motion controller
MCP	MetaCarpoPhalangeal
MDH	Modified Denavit-Hartenberg
MDTW	Multi-dimensional Dynamic Time Warping
ms	millisecond
mHMM	modified Hidden Markov Model
NHS	National Health Service
PC	Principal Component
PCA	Principal Component Analysis
PIP	Proximal InterPhalangeal
SDK	Software Development Kit
SVD	Singular Value Decomposition
SVM	Support Vector Machine
TM	TrapezioMetacarpal
TMC	TrapezioMetaCarpal
TWED	Time Warp Edit Distance
USB	Universal Serial Bus

1 Introduction

This research focusses on hand gesture recognition. The implemented hand gestures/exercises, as discussed in Chapters 5 and 6, are similar to those employed in a hand rehabilitation[1] setting although there are other possible applications. These include sign-language recognition, gaming, and so forth.

Although our research does not involve direct clinical support, we are motivated by a recent publication by the United Nations [2] revealing that there has been a consistent increment in life expectancy worldwide hence rising the number of aged individuals. The publication further anticipates that there will be approximately 2 billion older people (aged 60 years or over) i.e. roughly 20.0% of the global population by the year of 2050. Around 90.0% of these older people are likely to live independently outside of an environment with ready access to clinical support. This provides strong motivation for investigating technologies that can operate satisfactorily in a home setting. Whether in a hand therapy context [3] or otherwise, gestures can be monitored via wearable devices or contactless alternatives, or a combination of the two. Wearable sensor devices are applied to collect physiological and motion data. On the other hand, contactless methods aided by computer vision techniques have been recently implemented and described in literature. These are low-cost markerless motion devices that can track fingers, wrist, and forearm movements [4, 5].

Treatment of hand impairments typically involve a clinician proposing a series of hand exercises that the patient should perform, typically in the presence of the therapist. This is demanding in terms of resources, which include time, costly instrumentation and testing equipment, the use of health-care facilities and the related overheads [6]. Various rehabilitative sensor frameworks have been proposed and implemented but they are

[1] Rehabilitation is a treatment or treatments where a patient performs several physical fitness training tasks to gain a physical functioning level that permits him or her regain initial physical, sensory and mental capabilities that were lost, due to an accident, a stroke, or surgery [1].

unnecessarily costly. For example, in [7] an intelligent game engine was proposed for home-based hand recovery exercises of patients recovering from strokes. The engine presents various games that aid patients to perform exercises and is merged into a patient station that offers monitoring and feedback. Another example is in [8] where a contactless finger displacement measurement system was proposed and implemented. The system consists of a non-contact measuring device and does not require a complicated set-up. Indeed, many researchers have turned their attention to proposing contactless systems with the help of computer vision techniques [4, 5, 9, and 10]. However, vision-based systems as described in literature still exhibit issues. Reliability of such frameworks depends on various circumstances e.g. the visual background, illumination and contrast, noise related factors, ease of use etc. [4]. For example, an approach proposed in [9] aids hand therapy exercises using a coloured ball fixed to a hand gripper. However, this approach may not support hand finger exercises effectively since a user is required to grasp when performing a hand procedure exercise [10].

We propose a prototype of low-cost vision–based system for hand gesture recognition to overcome some of the limitations existing in systems presented in literature. Our prototype makes use of a low-cost “off-the-shelf” device. The device is a visual markerless sensor system called the Leap Motion controller (LM). LM is a popular markerless sensor system developed for gesture interaction [4, 11]. Compared with other markerless sensors such as the Intel RealSense 3D Camera, Kinect sensor, IMU (Inertial Measurement Unit) sensor [12], tilt sensor [13], and camera-based systems [10] that can perform similar tasks, the LM is superior in terms of portability. In addition, RealSense, Kinect, IMU and camera-based systems are costly and some of them unable to detect small movements [11]. Moreover, the tilt sensor involves complicated set-up before use. However before deploying such a device in a hand therapy environment, a thorough analysis and evaluation of its performance is required.

Guna et al. [14] conducted an evaluation of the device. They tested the accuracy and precision of the device based on the elevation distance from its surface going upwards. They discovered that accurate measurements are not possible beyond an elevation of 25

cm. They did not test its accuracy based on other parameters such as lateral (side-to-side) and forward-backward positioning relative to the LM. They also did not suggest any scientific measures to mitigate the inaccuracy of the device. Separate from their work, we have tested the LM considering various parameters, namely: elevation, lateral (side-to-side), forward-backward movements and rotation of the hand. We have discovered that errors do exist as the hand is moved even a relatively small distance from a baseline central position above the LM. We have further implemented Principal Component Analysis (PCA) to compensate for these errors. The reasons for choosing PCA are described in Section 1.1.

The PCA technique has already been employed to detect online abnormal medical sensor readings [15-18]. The technique was used to analyse collected physiological measurements from sensors to detect the occurrence of multivariate anomalies based on the squared prediction error at runtime [15]. However, the researchers did not specify particular sensors where the proposed and implemented technique could work best. In our research, we apply PCA to finger joint angle data collected from the LM to compensate for absolute errors.

We further applied machine learning techniques i.e. Linear Discriminant Analysis (LDA) and Support Vector Machine (SVM) to compensate for measurement errors from the LM. Through training of these machine learning models, hand gestures can be classified into a set of available hand gestures [19] taking into consideration measurement errors and noise in the LM device. The reasons for choosing these techniques rather than others can be found in Section 1.1.

In addition, we propose and implement Dynamic Time Warping (DTW) and Multi-dimensional Dynamic Time Warping (MDTW) to quantify how dissimilar two dynamic gestures are from each other [20, 21]. The reasons for employing MDTW can be found in Section 1.1. We implement a MDTW algorithm that processes hand motion signals to establish whether a patient would perform right dynamic hand gestures in comparison to predefined ones that are recommended by a physiotherapist. Dynamic hand gestures are represented as a sequence of positions changing over time and each position is described by a set of joint angles. A dynamic hand gesture that would be performed by a patient is

aligned against a dynamic reference hand gesture in a database. MDTW produces a minimum-distance value of a warp path as the end result after aligning a query dynamic hand gesture with a reference one. Low minimum-distance value implies the two gestures being compared are similar and a high minimum-distance value implies the two gestures vary to a greater extent. [22]. The minimum-distance value between two dynamic hand gestures is computed by adding up the Euclidean distances between pairs of points of a warping path.

Our proposed markerless visual sensor-based methodology can detect precise movements of fingers within the interaction area of the sensor (See Section 3.1.1). The objective of the LM system is to allow individuals with hand injuries to practice hand movement exercises at home or clinic, without the need of an always-present therapist. Inspired by in-the-field low-cost systems, this research proposes a prototype in order to precisely assist patients performing hand therapy exercises outside a hospital environment using readily available devices such as LM.

1.1 Motivation

Traditional hand therapy is executed one-on-one, suggesting one physiotherapist (or occasionally many) helping one patient. Thus the costs are high especially for demanding patients e.g. those suffering from traumatic brain complications or spinal cord related illness. We have proposed and implemented a cheap and “comfortable to use” device that is suitable for hand recovery treatment. Several hand motion capture frameworks presently use devices that are relatively costly [23, 24] and uncomfortable for patients to operate due to huge batteries. In our research the focus is on a low-cost computer-vision assisted hand therapy framework in order to mitigate these two challenges and in the process improve people's quality of life by facilitating a would-be affordable and convenient hand exercise recovery set-up.

In a hand exercise recovery settings, costs of medical related services and an increasing number of aging people are directing the provision of medical care and other related services away from medical centres i.e. at home [25]. Hand injury survivors often

need a lengthy and costly hand therapy procedure to regain some of the hand functions that are lost due to hand injuries [26]. Issues such as making appointments to suit both a patient and a physiotherapist, travel related challenges to and from a medical centre, etc can hinder the quality of hand restoration related services [26] and this further suggests the need for employing home-based hand therapy using convenient and accessible hand recovery treatment devices. However, these systems greatly rely on fairly precise measurements and representation of hand motion. This can be easily facilitated by use of low-cost devices that have been developed, tested and readily available in the market.

Sensor technologies for human hand motions can be grouped into two types i.e. contact-based sensors and non-contact-based sensors [27]. The contact-based sensors obtain sensing information when fixed on the human hand while non-contact-based sensors obtain sensing information without being directly fixed on the human hand [27]. It is important to note that all these hand sensing technologies have issues, for example hand data gloves, categorised as contact-based sensors, experience some challenges. Therefore, researchers need to devise techniques to overcome these challenges of comfort, rigidity, and robustness when implementing hand sensory gloves for medical and industrial applications [28]. Some challenges of attached force sensors, that are categorised as contact-based sensors, are electromagnetic noise, being sensitive to temperature, drift of sensor output for piezoelectric sensors etc. [27]. Surface electromyograph sensors, also categorised as contact-based sensors, experience issues resulting from crosstalk, issues due to electrode displacement, information redundancy, and a lack of simple mechanisms for selecting suitable features from the raw data [29]. Contact-based optical markers also experience issues while tracking hand movements. The issues include marker placement errors caused by skin deformation and unsuitable markers that make users feel uncomfortable [30] when using them. All these issues need to be addressed by researchers. Compared to the above-mentioned sensors, LM is categorised as a non-contact-based sensor system and has an extra advantage since all the above issues are not experienced when using it. In addition, LM is a lightweight and affordable device [31, 32]. Moreover, it can be easily operated by a user since it only requires a simple calibration procedure when

operated for the first time.

In our proposed hand injury restoration prototype, we first conduct finger joint angle error analysis where a series of experiments are performed to assess the accuracy of the LM in terms of parameters such as elevation, lateral (side-to-side) positioning, forward-backward positioning, and rotation of the hand relative to the LM. If no measurement errors and noise were experienced, our dataset would have 0.0^{o} as variance for each of the five variables (See Sections 4.5, 4.6, 4.8, and 4.9 in Chapter 4) however this was not the case as each variable experienced variance. We devise a filtering technique that is able to reduce the absolute errors (AEs) in our dataset. Some of the known techniques that can be employed include a moving average filter, Savitzky-Golay filter, local regression filter [33], Principal Component Analysis (PCA) [34] etc. Compared to other filtering techniques, PCA is more efficient and easier-to-use in relation to variance reduction in a dataset. PCA computes a new set of variables i.e. principal components (PCs) that expresses a dataset in order of high variance represented in the first PC. In order to reduce absolute errors (AEs), we employ PCA on our dataset. PCA reduces variance and hence absolute errors as illustrated in Equation (4-5) in Section 4.2.3. We delete the first PC that contains the highest variance and retain the remaining PCs that form the feature vector. According to our dataset, deleting the first PC whilst retaining the remaining PCs in the feature vector is suitable means of reducing absolute errors. Details on how PCA was implemented can be found in Section 4.2.3.

We propose and implement LDA and SVM in order to recognise and classify hand gestures. LDA and SVM are categorised as supervised machine learning techniques [35]. In supervised learning, models are trained using a labelled dataset; however, in unsupervised learning, models need to find the mapping function to map the input variable with the output variable [35] in a non-labelled dataset. We have implemented supervised machine learning techniques since our dataset contains labelled observations (data points) of four hand gesture categories. Both LDA and SVM can learn about measurement errors and compensate for them. LDA performs relatively well compared to other models such as a logistic regression model when the classes are separated relatively well [35]. Under this

condition the parameter estimates for the logistic regression model are inaccurate. Furthermore if the number of observations in a dataset is small and the distribution of the measured features are relatively normal in each of the groups, the LDA model also performs better than a logistic regression model [35]. SVM has been proved to perform well in various settings and is normally known to be a superior classifier [35]. Basically SVM is employed to recognise and classify measured observations of a dataset that consists of only two classes. However multiclass recognition and classification is possible when many classifiers are combined [36]. Our objective is to create an SVM model after training a fraction of measured observations on our dataset with an aim of correctly recognising and classifying test samples of the hand gestures. Both LDA and SVM can learn about measurement errors in LM and hence compensate for them.

We consider and implement both DTW and MDTW, however there exist other possible dissimilarity measures such as Longest Common SubSequence (LCSS) [20, 37], Edit distance with Real Penalty (ERP) [20], Edit Distance on Real sequences (EDR) [20, 38], and Time Warp Edit Distance (TWED) [20, 39]. However DTW has an extra advantage since its implementation is simple and efficient. Furthermore, DTW is superior because it is not essential that both time-series being compared are of equal length as required by typical distances and this behaviour is termed elasticity [20]. DTW is therefore an elastic dissimilarity value that estimates the greatest match within two time-series by reducing a distance between them. We have noticed that dynamic hand gestures may vary e.g. in duration, speed, rotations, etc. and our MDTW approach takes this into account.

1.2 Contributions

Our novel contributions are as follows:

1. Characterise joint angle absolute errors based on parameters such as elevation, lateral (side-to-side), forward-backward and rotation movements of the hand relative to the Leap Motion controller (LM).
2. Implement and evaluate Principal Component Analysis (PCA) to compensate for the absolute errors in the LM as well as improving the accuracy of LDA, SVM and DTW.

3. Employing Linear Discriminant Analysis (LDA) and Support Vector Machine (SVM) models to recognise and classify hand gestures taking into consideration LM measurement errors and noise. We finally evaluate LDA and SVM models in comparison to a baseline.
4. Implement and evaluate a Multi-dimensional Dynamic Time Warping (MDTW) technique. MDTW establishes how similar or different a query dynamic hand gesture is in comparison to a reference one. MDTW produces a minimum-distance value of a warp path as the end result after aligning a query dynamic hand gesture with a reference one.

1.3 Thesis outline

Chapter 1 provides an introduction, motivation and the novel contributions of our research. This is followed in Chapter 2 by a state of the art review. In this chapter we describe different sensor technologies related to hand motion capture systems. We discuss types of hand gestures, provide a hand overview that is essential for our research, and finally discuss a brief kinematic human hand model that is relevant to our research.

Chapter 3 illustrates background theory of LM device, PCA, LDA, SVM and DTW. We also formulate a specific problem regarding gesture recognition and classification using LDA and SVM. We further formulate a specific problem regarding the comparison of a dynamic query hand gesture and a dynamic reference hand gesture using DTW and MDTW. We finally review research where LDA, SVM, DTW and MDTW have been implemented in hand gesture recognition. Then in Chapter 4, we characterise finger joint absolute errors, illustrate how PCA compensates for these errors, and present results and discussion regarding PCA. Next, in Chapter 5, we represent static gestures using a feature vector in 3D space. We also train LDA and SVM models to recognise and classify sample gestures taking into consideration LM measurement errors and noise. We finally evaluate LDA and SVM models in comparison to a baseline. In Chapter 6, we implement and evaluate the efficacy of MDTW technique. Following this, in Chapter 7, we implement PCA to see if it can improve on the accuracy of LDA, SVM and MDTW. Finally, concluding remarks and future work are presented in Chapter 8.

2 State of the Art Review

Human hand movement tracking devices have the ability to gather hand motion data through various approaches that dynamically illustrate the position differences of a human hand on the basis of motion sensor related technologies. In hand motion data analysis, the hand is commonly represented as a system of rigid elements joined by revolving joints. In most scenarios, measurement strategies can obtain an acceptable field of view on a human hand model or near where sensor devices are positioned. Motion of sensor devices is defined in relation to a reference coordinate system and from sensor position and direction. In [40-42], researchers noticed that most of the hand motion capture techniques are highly dependent on accurate positioning of sensors and initial calibration of sensor devices is a prerequisite before measurement. However, in our hand gesture recognition system, we do not rely on accurate sensor positioning and there is no need for a complex calibration before measurement, though a user is required to move his or her hand within the LM interaction area (See Section 3.1.1). It is important to note that a simple calibration procedure is required when the LM device is used for the first time. Details of the calibration procedure are described in Section 3.1.3.

2.1 Sensor Technologies for Human Hand Motion

Sensor technologies for human hand motion can be grouped into two types i.e. contact-based sensors and non-contact-based sensors [27]. The contact-based sensors obtain sensing information when fixed on the human hand while non-contact-based sensors obtain sensing information without being directly fixed on the human hand [27]. Contact-based hand sensor technologies include hand data gloves, attached force sensors, surface electromyography, and optical markers. Non-contact-based sensor technologies include ordinary cameras, depth cameras, and Leap Motion controller (LM). Figure 1 illustrate various human hand sensor technologies.

2.1.1 Contact-based Sensor Technologies

a) Hand Data Gloves

A hand data glove is an electronic device that possesses various types of sensors

that track hand movements. Hand data gloves can be employed to grasp, move, and rotate objects in a virtual environment [27]. Hand data gloves can detect finger bends and utilise magnetic position sensors to locate the hand position in 3D space [27]. Some of the available and mostly used hand data gloves on the market are illustrated in Figure 2. In the figure, (a) is a DG5-V glove, (b) is a CyberGlove III, (c) is a ShapeHand glove, (d) is a VHand DGTech, (e) is a 5-DT 14 glove, and (f) is a CyberWorld P5 glove. Glove-based systems can be employed in medical applications, industrial applications, etc. This is because they can successfully operate with multiple degrees of freedom (DOFs) for each finger [27], because of having high accuracy, high response speed, and strong operability.

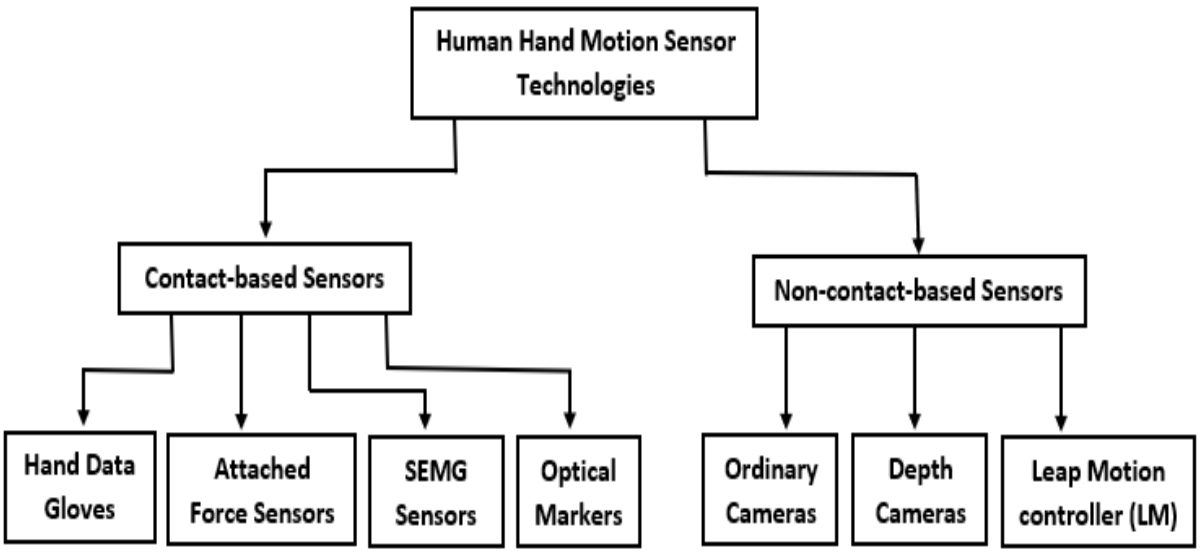

Figure 1: Human hand motion sensor technologies [27].

Researchers in [43] proposed a hand gesture recognition system based on a data glove and a Back Propagation (BP) neural network. The data glove is able to collect data of the hand and forearm, it illustrates all aspects of arm movement, and the BP neural network algorithm is employed to process and classify the collected data. Experimental results demonstrate that hand data can be classified effectively by BP neural network.

In [44], they propose a versatile soft sensing glove using a simple process for

implementing sensors of various sizes. Ten sensors were fixed to the back side of the glove to measure flexion-extension and four sensors are fixed on the glove at the interdigital folds between fingers to measure abduction-adduction. The sensory glove successfully replicates hand motion. The authors employed machine learning techniques to obtain the angles of the joints in the hand and also to recognise 15 hand gestures. The implemented sensory glove solves the issues of comfort, rigidity, and robustness. The implemented glove can also be employed in both industrial and medical settings.

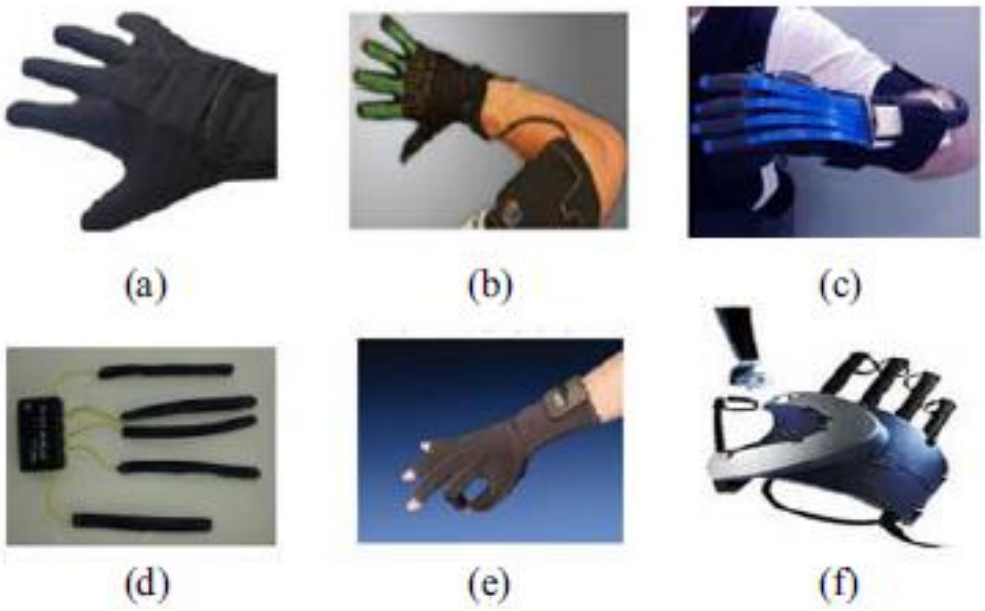

Figure 2: Examples of data gloves available in the market [27].

The authors in [45] implemented a hand recognition system that constituted a pair of gloves i.e. a sensory glove and a motor glove. Both gloves were implemented with a soft and flexible material that provides greater comfort and safety than conventional rigid hand recognition systems. The sensory glove which is fixed on the nonaffected hand possesses the force and flex sensors that assist in measuring the gripping force and bending angle of each finger joint for hand motion tracking. On the affected hand, the motor glove is driven by micromotors. This provides a driving force that helps in performing hand recognition tasks. Finally, machine learning techniques are applied to identify the hand

gestures from the sensory glove and to facilitate the hand recovery tasks for the affected hand [45].

It is important to notice that when implementing hand data gloves for medical and industrial applications, issues such as comfort, rigidity, and robustness are experienced [28]. Therefore, researchers need to devise techniques to overcome these challenges.

b) Attached Force Sensors

Several force sensors and techniques for measuring exerted forces have been proposed and implemented by researchers [27]. The most essential design criteria of force sensors in manipulation functions are the spatial resolution, robustness, sensitivity, and frequency response. Attached force sensors, which include four common force sensors i.e. capacitive, piezoresistive, piezoelectric, and strain-gauge need to meet the demands of object characterisation, identification, and manipulation [46].

Capacitive sensors acquire the displacement, force, and speed depending on the change in distances between the upper and lower electrodes due to external force variations. Piezoresistive sensors register the variations in the resistivity of the sensing material established on a silicon substance. Piezoelectric pressure sensors make use of the piezoelectric substances to develop an electric charge that is proportional to the pressure generated. Strain-gauges measure the variation in resistance [27].

Some challenges of attached force sensors that need to be addressed by researchers are electromagnetic noise, sensitivity to temperature, drift of sensor output for piezoelectric sensors etc. [27].

c) Surface Electromyography (SEMG)

SEMG provides a means of assessing the biofeedback of the motion of muscles by measuring the Electromyography (EMG) signal on the surface of the skin. By observing muscle contraction arrangements of human hand movements, the sensor system can establish the human's intention and carry out corresponding measures or communication, e.g. completing hand movements through prosthetic hands [47]. Research conducted by Al-Timeny et al. [48] applied an offline measure to assess the classification performance

depending on multiple-channel SEMGs. In research conducted by Y. Xue et al. [49], an SEMG-based in-hand motion recognition system was proposed to recognise various types of hand gestures. In line with regular movements in executing in-hand object manipulations, some sets of in-hand movements, including translation, transfer, and rotation were implemented in their sensor device. Then a nonlinear series analysis approach to SEMG signal processing was proposed to capture the nonlinearity of the hand movements. Experimental results demonstrated that a human in-hand motion recognition system could be implemented effectively to recognise various in-hand movements. Some challenges of SEMG sensors that need to be addressed by researchers are discussed in Section 1.1.

d) Optical Markers

Optical marker-based systems use cameras to track markers placed on a hand so as to track and monitor the motion of the hand. Various researchers have proposed many different platforms to implement marker-based hand motion capture.

In the work of [50], they proposed and implemented a robust online technique for recognising and monitoring of passive motion capture markers fixed on the fingers of hands. By implementing various assignment hypotheses and soft decisions, the system can strongly recover from a difficult situation with numerous simultaneous occlusions and incorrect observations (ghost markers). While the majority of state-of-the-art applications can regularly recognise and monitor markers on bigger components of the human hand, the markers fixed to fingers present unique challenges and generally necessitate a comprehensive set-up procedure before operation [50]. The biggest challenge in optical motion capture that implements markers is the identification and monitoring of the markers, usually stated as labelling. Figure 3 displays a choice of sparse marker collections and details of (a) to (f) can be found in [50].

The difficulties experienced with automatic labelling of finger markers derive from several factors. For instance, finger markers are susceptible to occlusion specifically when fingers are bent towards ground or the body. Another concern is that fingers have great mobility over a confined space, and markers arranged adjacent to each other (for instance close fingertips or close joints) can be inaccurately recreated as one single marker

or cause short-lived spans of noisy data [51]. Although they have devised techniques to reduce occlusions in their implemented system [50], occlusions are only partly reduced and further research is required to devise techniques that eliminate occlusion related issues. For example, when the LM approach is implemented, minimal and insignificant occlusion is experienced in hand motion tracking if a hand is moved in the area of interaction (see Section 3.1.1, [32], and [34] for more details).

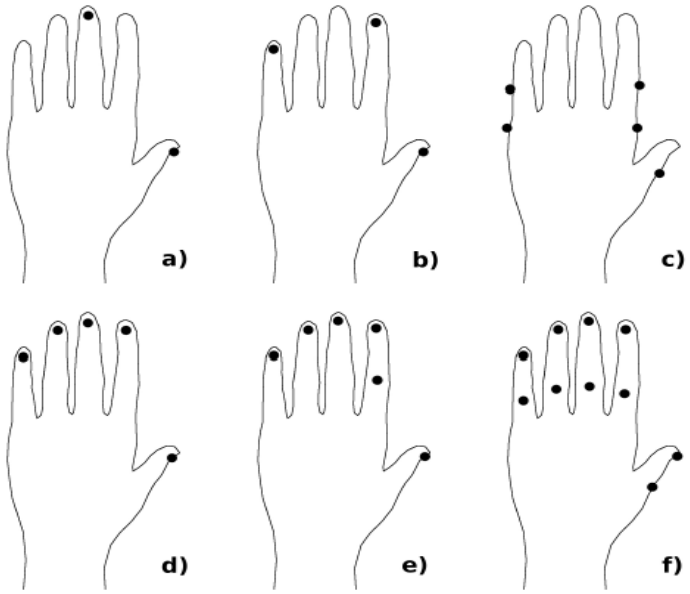

Figure 3: A choice of sparse marker collections [50].

2.1.2 Non-contact-based Sensor Technologies

a) Ordinary Cameras

Employing low-cost commercial markerless cameras in a hand recognition system, industrial setting etc is a possible solution to overcome issues related to contact-based sensor techniques. The issues are complex connection wires, surface properties, hysteresis, and sensitivity [27].

Research conducted by [52] proposes a multi-view set-up employing a readily convenient colour camera from a smart device such as a phone, and plane mirrors to

generate multiple views of the hand. Their system overcomes the challenges of synchronizing numerous cameras to reduce occlusion related issues. Further their experimental results demonstrate that a multi-view set-up could be helpful in reducing measurement errors when obtaining the flexion angle of finger joints [52].

Stereo camera comprise two same-specification digital camera. Through focusing, zooming and sensitizing, the 3D structure of the hand is created from various viewpoints. Stereo cameras possess specific lens angles and internal pre-calibrations, which grant the camera freedom of movement, but the angle between the two lenses is often too small to cover the occlusion space in hand movements. Furthermore, owing to the complexity of stereo geometry estimation, stereo intensity images are sensitive to light variations, therefore it is challenging to correspondingly match images for triangulation [27].

b) Depth Camera

As opposed to the ordinary cameras, depth cameras have ability to capture depth information and are more flexible and convenient to deploy in a 3D vision system than ordinary cameras when tracking and analysing hand movements [27]. The Kinect sensor is the most popular sensor employed in research in this category.

An example of a Kinect sensor system is proposed and implemented in [53]. The system needs a Microsoft Kinect hung from a rig over a table (ideal height for reliable tracking was 80 cm and variable rig 50-125 cm), permitting the users to put their hands above the table in order to make use of the system. The implemented system is illustrated in Figure 4. The Kinect sensor consists of both infrared depth and colour cameras and has an Application Programming Interface (API). In their system, they preferred to implement Kinect in its default mode (0.8-4 m) instead of near mode field of vision. The depth camera was configured to recognise a hand and fingers.

As reported by [54], with the implementation of infrared-based sensors, e.g. the Kinect, it has become simpler and more robust to recognise hand gestures. It has been implemented in a wide variety of applications e.g. tabletops, distant displays and 3D desktops. Keskin et al. [55] implemented an application for identifying finger spelling from

the depth data. However, while these kinds of system work efficiently for application-specific hand gesture recognition, they are not able to generalise and track a full range of hand movements [54].

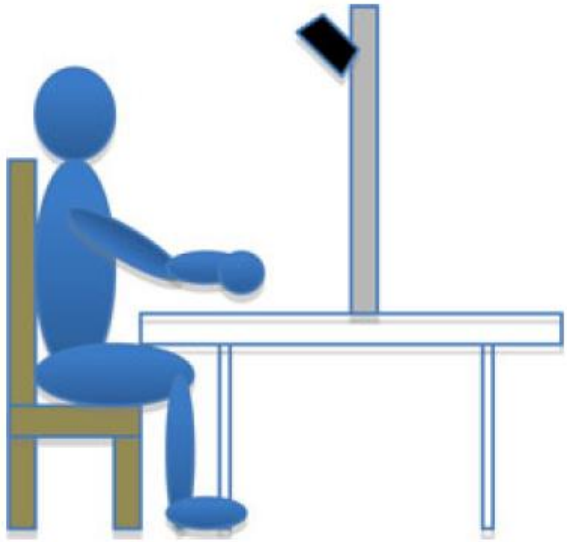

Figure 4: System set-up displaying a user and Microsoft Kinect [53].

c) Leap Motion controller (LM)

The introduction of LM has initiated recent opportunities for hand gesture recognition related research [56]. Unlike Kinect, LM was specifically invented for hand gesture recognition where it directly computes the position of a hand and direction or orientation of the fingers. An example on how to obtain this information is described in Section 4.1. In our research, we obtain vector positions of the bones of the fingers from data acquired by the LM. A collection of relevant features is derived from the data produced by the LM sensor and we compute finger joint angles for further analysis.

Different from the Kinect and other similar devices, the LM produces more limited information (only a few keypoints rather than a complete depth map) and its interaction area is smaller. However, the extracted information that is used to estimate finger movements and hand orientation is more accurate [56].

In Figure 5, the palm centre C approximately represents the centre of the palm area in 3D space. Hand orientation with reference to two unit-vectors, h, is directed from the palm centre to the fingers while n is at right angles with the palm plane directed downward from the palm centre. However, it is important to notice that this kind of estimation may not be obtained accurately and relies on how the fingers are arranged.

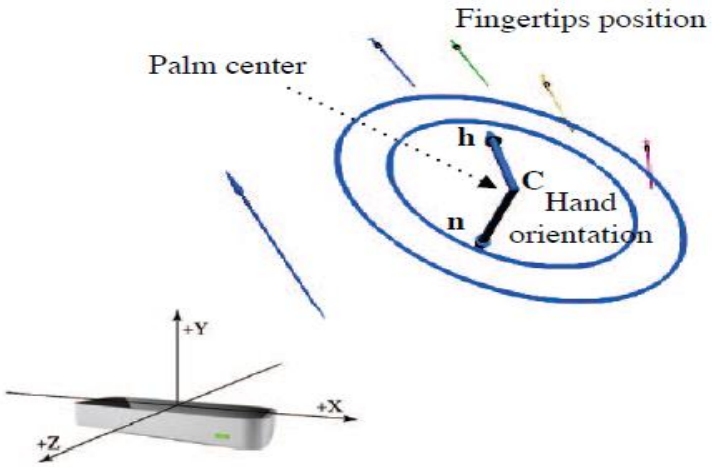

Figure 5: Data acquisition in Leap Motion [56].

2.2 Types of Hand Gestures

Hand gestures are useful in enhancing strength and dexterity irrespective of whether a sick person is gradually starting to restore hand movement or already possesses an acceptable range of hand motion. The gestures illustrated in Section 2.2.1 and 2.2.2 may assist in enhancing fine motor skills of patients that experience some form of weakness after a stroke or hand injury. The gestures to be performed by users undergoing a hand treatment restoration procedure can be classified into two types, i.e. static and dynamic gestures [57].

2.2.1 Static Hand Gestures

Characteristics for static gestures are primarily established on the basis of palm and finger relative lengths [57]. Figure 6 illustrates some examples of static gestures we may

efficiently recognise. The first line of three gestures are index L gestures, ILY gestures, and a fist. The second line of three gestures are thumb up, index pointing, and index and middle pointing. The third line of three gestures are V gesture, OK gesture, index and middle L gesture.

The length between the thumb and index finger is utilised to determine the OK gesture. The length between the index and middle finger is utilised to identify V gesture and the index and middle pointing gesture. The remaining related gestures can be obtained from a combination of the aforementioned gestures.

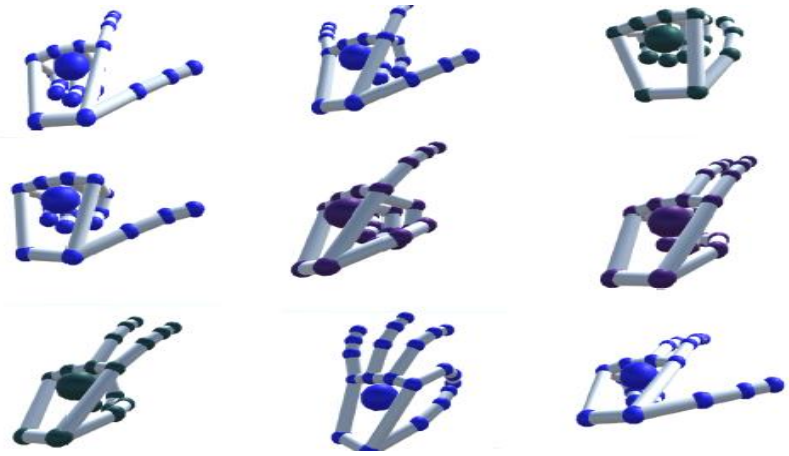

Figure 6: Examples of static gestures [57].

2.2.2 Dynamic Hand Gestures

Dynamic gestures can be readily differentiated from static gestures. Dynamic gestures commonly employ velocity of fingertips and orientation of the palm to determine movement configurations of the particular dynamic gesture. In comparison to static gestures, dynamic gestures are much more complex. An illustration of dynamic gestures is presented in Figure 7 (a). This type of gesture is defined as index finger key tapping. Index finger key tapping is constructed with reference to the index finger pointing static gesture. It is important to notice that the index finger moves vertically for index finger key tapping dynamic gesture. Figure 7 (b) demonstrates an example of an index finger circling direction

dynamic gesture. With this kind of gesture, the circle direction can either be clockwise or anticlockwise while the index finger is moving along a circle.

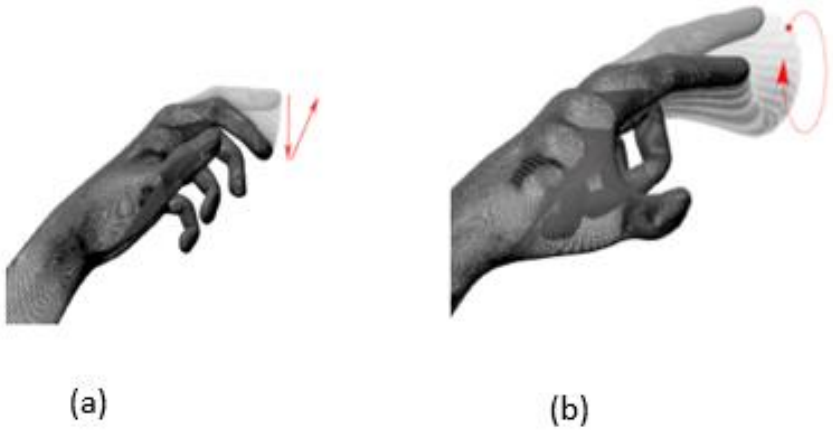

Figure 7: (a) Index finger key tapping. (b) Example of index finger circling [57].

2.3 Hand Overview

We briefly provide a hand overview that is essential for our research. Details that are not relevant are omitted.

2.3.1 Bones of the Hand

The hand consists of metacarpal bones; that is, 5 bones that run from the wrist to first joint of each finger. The hand also consists of 14 thin bones called phalanges. The thumb has only 2 phalanges whereas the remaining fingers have 3, i.e. the distal, middle, and proximal [58-60].

2.3.2 Joints of the Fingers

In Figure 8 the joints and bones of the hand are illustrated. The joints are formed whenever two bones of the finger intersect. It is important to note that the 4 fingers (index, middle, ring, and little) have three joints and the thumb only has two. We briefly explain all the types of joints [59].

a) Distal Inter-Phalangeal (DIP) Joint

The DIP joint is located near the tip of the finger, just before where the fingernail starts. It is the joint between the intermediate phalanx and the distal phalanx. This joint exists in all fingers with the exception of the thumb [59].

Figure 8: Hand bones and joints [58].

b) Inter-Phalangeal (IP) Joint

The IP joint is located near the tip of the thumb, just before where the nail starts. It is the joint between the intermediate phalanx and the distal phalanx [59].

c) Meta-Carpo-Phalangeal (MCP) Joint

The MCP joint is found at the base of all the five fingers. It is the joint between the metacarpal bone and the first phalanx bone [59].

2.4 Kinematic Human Hand Model

The model of interest to us can be employed in post-stroke or post-hand injury and is illustrated in the research work of [61-65]. According to [64], the kinematic hand model consists of 19 links that reflect the respective finger bones of a hand, and 24 DOFs

that represent all the finger joints. The Degree of freedom (DOF) is defined as the minimum number of independent coordinates required to represent a system's position. The only difference with this model is the inclusion of the Carpo-Meta-Carpal (CMC) joint on each of the four fingers (index, middle, ring and little) and the motion concatenation in the MCP joint [64]. The CMC joint permits simulation of the palm arc that illustrates the deformation in the palm while the hand is in the situation of grabbing tiny objects.

Two kinematic configurations are considered in this model, one for the thumb and other for the rest of the fingers (Details can be found in [65]). Therefore, the same kinematic configuration is employed for the index, middle, ring, and the little finger that are described by 4 links and 5 DOFs each. In these fingers, MCP joint is modelled by a 2 DOFs whereas the CMC joint, the PIP joint and DIP joint possess 1 DOF each. The thumb finger is modelled by 3 links and 4 DOFs. The Trapezio-Meta-Carpal (TMC) thumb joint is also described by 2 DOFs whereas MCP and IP joints are described by 1 DOF each [64].

According to [61], the simple Flexion/Extension (F/E) and Abduction/Adduction (Ab/Ad) of the thumb and the fingers are executed by the articulation of the 21 DOFs. F/E movements take place at every joint throughout the hand. Ab/Ad movements only take place at every finger's MCP joint and at the thumb's MCP and TM joints. The additional two interior DOFs are at the bottom of the metacarpals of the ring and the little finger that execute the curve or bend gestures of the palm [61]. If we carefully relate this to our preferred model in [64], F/E and Ab/Ad of the thumb and the fingers are illustrated by the articulation of only 20 DOFs; 4 DOFs at the CMC, i.e. 1 DOF on every finger apart from thumb (index, middle, ring and little), describe the curve or bend gestures of the palm.

Considering that the human hand is largely articulated with all 23 internal DOFs, it is also largely restricted [61]. By applying these restrictions, one may lower the number of DOFs in the hand and this renders human hand motion monitoring affordable.

A general restriction applied on the basis of the hand anatomy stipulates that to bend the DIP joints of the index, middle, ring, and little fingers, the respective PIP joints should also be bent, and this is well illustrated in Figure 9. In addition, a general technique employed to lower the total DOFs of a hand may be obtained on the basis of hand anatomy,

where the bending angles of DIP joints of the index, middle, ring, and little fingers are related with the PIP joints depending on the following mathematical relation: $\theta_{DIP} = 2/3\ \theta_{PIP}$ where θ_{DIP} is the flexion angle of the DIP joint and θ_{PIP} is the flexion angle of the PIP joint [61].

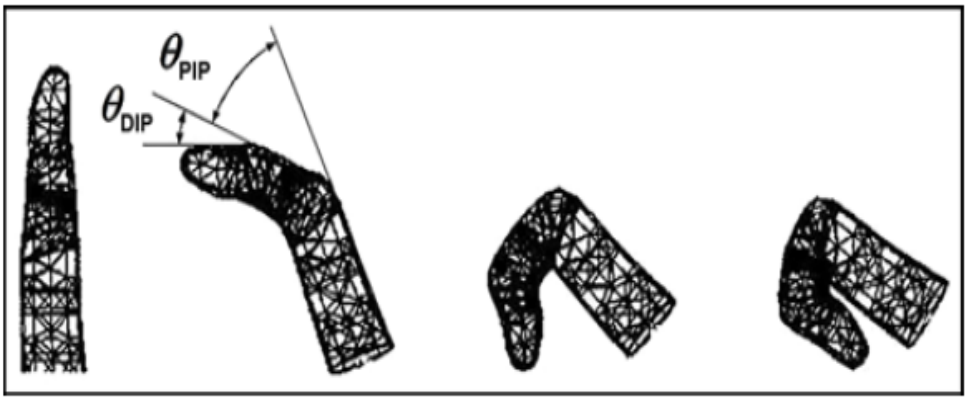

Figure 9: Bending restrictions between the DIP joint and the PIP joint [61].

2.5 Summary

In this chapter, different sensor technologies for human hand motion have been discussed. Specifically, these sensor technologies can be categorised into two types, i.e. contact-based sensors and non-contact-based sensors. We have considered how these types of sensor can be used for hand gestures and the issues that arise. Hand gestures can be classified into static and dynamic varieties. We provide a brief overview of the hand that is essential for our research in Section 2.3, where we discuss the bones of the hand and joints of the fingers. We also discussed a kinematic human hand model that is relevant to our research in Section 2.4 to introduce the basic scientific concepts required in human hand motion analysis.

We have reviewed sensor technologies for human hand motion to help us choose a suitable approach from the possible alternatives. As will be further explained at the beginning of Chapter 3, we selected LM because is one of the most accurate low-cost

hand tracking devices. The types of hand gestures we have reviewed have helped us to broaden our understanding with regard to hand gesture implementation and also to select appropriate hand gestures to evaluate with our proposed machine learning and signal processing techniques (as described in Chapters 4-7). In the next chapter the focus shifts to a review of the relevant background theory for LM device and multi-dimensional data processing.

3 Background Theory of LM, PCA, LDA, SVM, DTW, and MDTW

In this chapter, Leap Motion controller (LM), machine learning and signal processing techniques that have been employed in this thesis are discussed. In addition, some relevant literature that is closely related to our research is presented where these techniques are applied. The techniques that are discussed are PCA, LDA, SVM, DTW, and MDTW.

The authors in [66] have made a thorough comparison of hand tracking sensors on the market. They compared sensors such as the Kinect version 1, Kinect version 2, Xtion, Xtion Pro Live, Intel RealSense SR300, Intel RealSense D415, Myo Armband, Creative SENZ3D, etc. They found that the LM is one of the most accurate low-cost hand tracking devices. Owing to this finding, we employ an LM connected to a computing device using a USB cable. We then collect finger-joint data associated with hand gestures.

We employ the PCA filtering technique as described in Chapter 4 to reduce variance and hence absolute errors in our dataset. Compared to other filtering techniques such as moving average filters, the Savitzky-Golay filter, local regression filters, etc, PCA is more efficient and easier-to-use in relation to variance reduction and is able to reduce absolute errors in our dataset [33, 34]. LDA and SVM are implemented in Chapter 5 to recognise and classify hand gestures. In addition, LDA and SVM have the ability to learn about the measurement errors in the LM and hence compensate for them [4]. SVM has been proven to perform well in different settings [35] and hence is referred to as a superior classifier. LDA performs relatively well compared to other models such as a logistic regression model when the classes are separated relatively well [35]. In Chapter 6, we implement both DTW and MDTW to quantify how similar or different two dynamic hand gestures are from each other [20]. Compared to other dissimilarity measures [20, 37-39], DTW and MDTW have an additional advantage since their implementation is simple and efficient. Furthermore, DTW and MDTW are remarkable since it is not necessary that both time-series being compared are of equal length as needed by typical distances and this is referred to as elasticity [20].

3.1 LM Motion Capture System

We use a low cost “off-the-shelf” markerless sensor known as LM. LM is a popular device developed for hand gesture interaction [4]. It is a portable USB device of dimension 0.5x1.2x3.0 inches that connects to a computer. It comprises of two cameras and three infrared light-emitting diodes (LEDs) that detect hands and all 10 digits as they move through the open space between a person and a computing device. There is free LM software from its Software Development Kit (SDK) that must be installed to work with the LM. The software together with any customised application can detect hand and fingers and translate motion data into information that is required for further analysis. Figure 10 shows the front side of the LM.

To use the LM, the user needs to connect it to a computer using a USB cable, start the LM software, and then place his or her hands above the LM. Figure 11 illustrates how to use the LM. In the figure, The LM in the centre is connecting to a computing device on the right. The hand above the LM is tracked and used to interact with virtual objects [67, 68]. The LM can detect palm and finger movements. The tracking data, in form of frames, can be accessed using its SDK.

Figure 10: The Front Side of the LM with the Green Signal.

A study concerning LM latency related issues has been made in [66]. The study demonstrated fluctuations in the frame rates. The nominal frame rates were 50, 100, and 200 fps in the modes of high precision, balanced tracking, and high speed. The three modes

experienced delays of 20, 10 and 5 ms, respectively. These latency related issues have no impact on our experimental results for both static and dynamic hand gestures since the delays are small compared to the durations of dynamic hand gestures.

Figure 11: The LM Usage. The hand above the LM interacts with virtual objects [53].

3.1.1 Interaction Area of LM

The interaction area of the device is approximately 60 cm above the LM, roughly 60 cm wide on every side (150^{o} angle), and roughly 60 cm deep on every side (120^{o} angle) [67]. Owing to its extensive angle lenses, the device possesses a large interaction area of eight cubic feet that takes the form of an upside-down pyramid which is the intersection of the binocular cameras' fields of view. In the earlier versions of LM, the field of view was restricted to approximately 2 feet (60 cm) above the sensor device. With the implementation of Orion beta related software, this was upgraded to 2.6 feet (80 cm). It is important to note that this range is restricted by LED light transmission through space, because it becomes more complex to detect a hand's location in 3D above and beyond a specific distance [67]. LED light intensity is constrained by the highest current that can be carried by the USB connection cable. Figure 12 (a) shows the interaction area of the LM.

3.1.2 Internal Structure of LM

The internal structure of the LM can be easily recognised when someone first determines its centre. The centre is at the second IR LED as shown in the Figure 12 (b). When the LM is working, all 3 IR LEDs are clearly observable displaying red illumination.

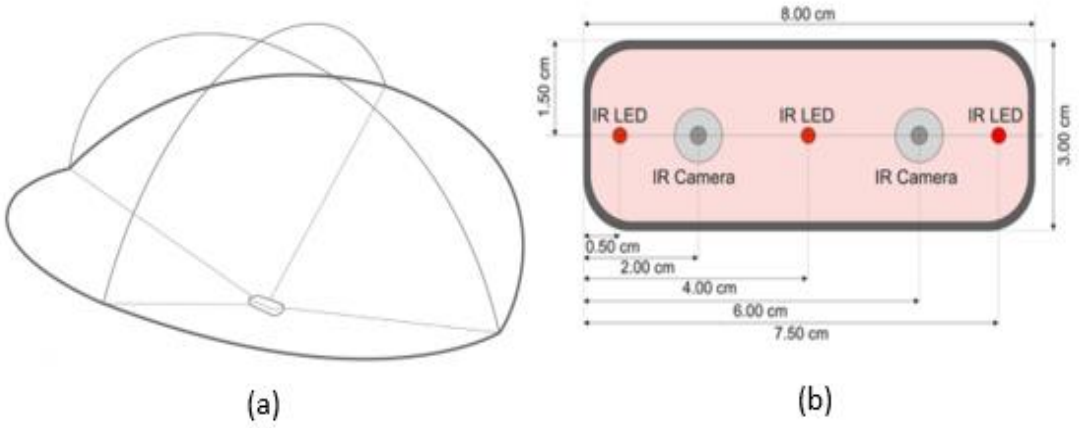

Figure 12: The LM's (a) Interaction area (b) Internal Structure [67].

3.1.3 Simple Calibration Procedure of the Leap Motion Device

For the LM to be used initially, a user is required to perform a simple calibration procedure since the sensors on the LM may have been knocked out of their initial arrangement [67]. If a calibration procedure is not performed issues such as jumpiness, persistent disruptions in the tracking data, irregularities in tracking data that happen only in particular fields of observation, and a smaller tracking area can be experienced.

In order to calibrate the LM, the following steps are performed [67].

(a) Start LM application on a computing device.

(b) Click on File tab and then click on Controller Settings.

(c) Click on the Troubleshooting tab.

(d) Click on Recalibrate Device.

(e) Pan and tilt the LM to turn the cursor over to paint the screen.

(f) Aim at obtaining a calibration mark of 80 or beyond.

3.1.4 Set-up and Comfortable Use of the Leap Motion Device

According to [67], the following are suggestions to help make employing the LM feel natural and easy. The “do’s” are listed as follows:

(a) Put down the LM on a desk or flat surface where it is convenient to use.

(b) When standing or sitting high on a chair, hold your elbows close to your side. Maintain your forearms parallel to the floor and perpendicular to your body.

(c) When sitting at a normal desk height, keep your elbows or forearms on your desk. Keep away from uneven and piercing edges. Keep your hands simply above the LM, your wrist and hands in approximately a straight line. Figure 13 illustrates the do’s when using LM.

Figure 13: Illustration of the do’s when using LM [67].

The don’ts are listed as follows:

(a) Don't hunch over the LM. Maintain its field of view free from obstacles, including yourself.

(b) Don't flex your elbows and your wrists with your arms nearly together.

(c) Don't maintain your arms straight ahead of you in the space.

(d) Don't hold your arms on a surface where it compels your elbows to be pointed out to the side. Figure 14 illustrates the don’ts when using LM.

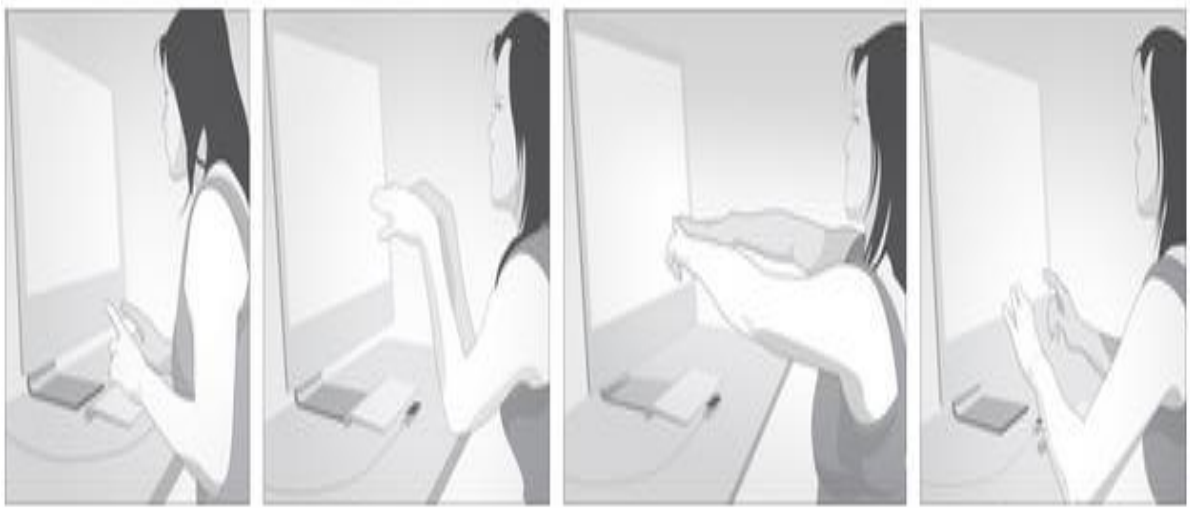

Figure 14: Illustration of the don’ts when using LM [67].

3.2 Principal Component Analysis (PCA)

Principal Component Analysis (PCA) is a multivariate approach that transforms a collection of correlated data points into a new collection of orthogonal and uncorrelated variables referred to as principal components (PCs) that are linear combinations of the original variables [69, 70]. The conversion of the data to the PCA domain is executed by the decomposition of the covariance matrix into eigenvalues and eigenvectors, and this method has been employed in various application areas under diverse techniques, such as a denoising approach, with the advantage of being a suitable tool from a computational point of view [71, 72].

3.2.1 PCA as a Filtering Mechanism

Principal Component Analysis (PCA) starts with the arrangement of a dataset in a matrix X of dimension M x N, in which M represents the number of observations and N the

number of variables [70], as illustrated in Equation (3.1).

$$\mathbf{X} = \begin{bmatrix} x_{11} & \cdots & x_{1N} \\ \vdots & \ddots & \vdots \\ x_{M1} & \cdots & x_{MN} \end{bmatrix} \quad (3\text{-}1)$$

To prevent data points that are distant from the data centre having a significant impact rather than closer points, the average of each variable is subtracted from the data point. This step is referred to as centralisation of data and is demonstrated by Equation (3-2).

$$\mathbf{y_i} = \mathbf{x_i} - \mu(\mathbf{x_i}) \quad (3\text{-}2)$$

where y_i is the data vector centralised around the average. x_i are the *N* sample vectors and $\mu(x_i)$ is a vector which is a collection of averages of the sample vectors, which can be obtained by Equation (3-3).

$$\mu(\mathbf{x_i}) = \frac{1}{m}\sum_{j=1}^{m} x_i(j) \quad (3\text{-}3)$$

The variables or sample vectors for each column of *X* [71] are demonstrated mathematically by Equation (3-4).

$$\mathbf{x_i} = [x_{1i} \;\; x_{2i} \;\; \cdots \;\; x_{Mi}]^T \quad (3\text{-}4)$$

The data matrix arranged and centralised on the average is used to obtain the covariance matrix as shown in Equation (3-5).

$$\mathbf{C_Y} = E(\mathbf{Y} \cdot \mathbf{Y}^T) \quad (3\text{-}5)$$

in which Y and Y^T are the data matrices centred on the average and its transpose, respectively, and C_Y is the covariance matrix.

The diagonal elements of C_Y indicate the statistical variance while the off-diagonal elements demonstrate the covariance between variables. Null diagonal covariance signifies the random variables are uncorrelated [73]. In addition, the covariance matrix is real and symmetric, and this allows us to decompose C_Y into a collection of eigenvalues and orthogonal eigenvectors [72] as illustrated in Equation (3-6).

$$\mathbf{C_Y} = \mathbf{V} \cdot \Lambda \mathbf{V}^T \quad (3\text{-}6)$$

where $V = [\Phi_1\ \Phi_2\ \cdots\ \Phi_m]$ is a M × M orthogonal eigenvectors matrix and $\Lambda = diag\ \{\lambda_1, \lambda_2, \cdots, \lambda_m\}$ is the diagonal matrix of eigenvalues, in which $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_m$.

The eigenvectors show the contribution to each of the initial axes to the structure of new axes, i.e. the principal components. The eigenvalues, consecutively, are related to the original amount of variance described by each of the eigenvectors [74].

The final step of the analysis is the creation of the uncorrelated data matrix that is referred to as the principal component scores, and which is created by the product of the orthonormal eigenvector matrix V and the data matrix arranged and centralised on the average Y, as illustrated in Equation (3-7).

$$\mathbf{PC} = \mathbf{V}^{\mathrm{T}}\mathbf{Y} \tag{3-7}$$

where PC is the matrix of uncorrelated principal component scores.

From the data in the PCA domain, it is feasible to obtain signal characteristics. The signal and the noise in a dataset can be easily separated in the PCA domain, because the signal energy and noise energy will settle in various subsets of the uncorrelated data. Considering this ability, PCA can be employed as a statistical data filtering approach [74].

The inverse PCA transform can also be considered, which is employed to back transform the principal component scores (uncorrelated data), consequently building the original dataset. Equation (3-8) illustrates the mathematical expression of the inverse PCA transform.

$$\mathbf{X} = (\mathbf{V} \cdot \mathbf{PC}) + \mu(\mathbf{X}) \tag{3-8}$$

The inverse PCA transformation is an important operation since rebuilding of original data with only several PCs, discarding the remainder, can strengthen relevant features not formerly easily observed in the data and/or separate the contribution of unwanted features such as noise [71, 72, 74].

3.2.2 Application of PCA in Areas Related to our Research

As we have discussed in Section 3.1.1, PCA is a multivariate statistical approach that transforms an orthogonal projection of the data onto a new dimensional linear space,

known as the principal subspace [74]. We briefly describe some related applications where PCA is employed.

a) Application of PCA to Detect, Recognize and Interpret Hand Gestures

PCA was applied together with the LM and the Kinect sensor to detect, recognise and interpret hand gestures of Arabic sign language letters. PCA was implemented as a means to overcome the challenge of time complexity of acquiring and interpreting data for their system [33]. They deleted the least significant principal components they believed contributed redundant and irrelevant data due to noise.

Differently to how they implemented PCA, we apply PCA to compensate for the errors that would have originated from the user's hand moving away from the centreline of the LM when performing hand gestures. For reducing absolute errors, we employ PCA on our dataset. PCA reduces variance and hence absolute errors as illustrated in equation (4-6). We delete the first PC that contains the highest variance and retain the remainder PCs that form the feature vector. Details on how PCA is implemented are discussed in Section 4.2.3, Chapter 4.

b) PCA Employed to Lessen Redundant Information of Hand Movement Signals

In the work [75], PCA is employed to create the hand gesture recognition system in order to lessen the redundant information of electromyography (EMG) signals, improve recognition efficiency and accuracy, and increase the possibility of real-time hand gesture recognition. By employing mechanisms for obtaining key information of human hand movements, the required action mode can be detected. In their research, nine hand static gestures were implemented where the surface EMG signal of the arm was collected using an EMG device to obtain four types of characteristics of the signal [75]. After employing PCA, the overall recognition rate of the implemented system achieved 95.1% accuracy.

3.3 Linear Discriminant Analysis (LDA)

LDA takes into account multivariate analysis. It makes use of independent variables to establish a distinction among groups or categories of dependent variables [4, 76]. The technique generates discriminant functions that help to determine the group to

which a test sample is attached.

3.3.1 How LDA Solves the Problem

LDA is a supervised learning technique where the outcome variable is categorical and hence can be employed in a classification scenario. LDA can be immensely effective when dealing with two response classes [4]. However, it can also be extended to handle a multi-dimensional dataset where more than two classes can be classified [4]. In Chapter 5, we demonstrate how LDA is implemented to recognise static hand gestures.

We let a collection of K independent gestures be described in a set G i.e. $G = \{g_1, g_2, g_3, \cdots, g_K\}$, that represent the position and orientation of the fingers. A gesture is represented using a set of measured features describing finger joint angles and these features can be measured using an LM. A feature vector in dimensional space of P features, i.e. $y = \{y_1, y_2, y_3, \cdots, y_P\}$, is used to represent the position and orientation of the fingers for a particular gesture.

We are required to obtain an appropriate gesture $g_k \in G$ which will maximise the probability $P(g_k|y) = P(g_k|y_1, y_2, y_3, \cdots, y_P)$ as,

$$P(g_k|y) = \frac{P(g_k)P(y|g_k)}{P(y)} \tag{3-9}$$

where $P(g_k)$ and $P(y)$ can be computed, assuming the P features in y are independent of each other. The probability of a gesture $P(g_k)$ is the ratio of the number of samples in a test sample that belong to the category of g_k to the total number of samples in the test sample. Estimation of the likelihood, i.e. $P(y|g_k)$ is required using a machine learning technique on a labelled dataset.

A user performing hand gestures described in Section 5.1.4, Chapter 5 is instructed to perform a number of gestures that are recorded in the system and measurement features, representing these gestures in form of feature vectors, are recorded. The problem to solve is, given a set of gestures G, our objective is to recognise and classify which gesture a user is performing taking into consideration measurement errors and noise within the LM.

We have employed a multiclass LDA model since we have more than two gestures in the set *G*. The steps involved in multiclass analysis are well described in [4, 76, 77].

Step 1: LDA starts by finding intraclass S_1 and interclass S_2 scatters from K independent gestures in set G [78-80]. The values of S_1 and S_2 can be estimated from the training dataset as,

$$S_1 = \sum_{k=1}^{K} \sum_{x \in g_k} (x - \overline{x_k})(x - \overline{x_k})^T \quad (3\text{-}10)$$

$$S_2 = \sum_{k=1}^{K} (\overline{x_k} - \bar{x})(\overline{x_k} - \bar{x})^T \quad (3\text{-}11)$$

where $\overline{x_k}$ is the average of each category of gesture k and $\bar{x}$ is the overall average. These aforementioned averages are easily computed as,

$$\overline{x_k} = \frac{1}{m_k} \sum_{x \in g_k} x \quad (3\text{-}12)$$

$$\bar{x} = \frac{1}{m} \sum_{i=1}^{n} m_k \overline{x_k} \quad (3\text{-}13)$$

where m_k and m represent the number of observations in category k and the total number of observations in all categories, respectively.

Step 2: After obtaining S_1 and S_2, we can find the transformation Φ that maximises

$$\ell(\Phi) = \frac{|\Phi^T S_2 \Phi|}{|\Phi^T S_1 \Phi|} \quad (3\text{-}14)$$

The transformation Φ can be finally obtained by finding a solution to the generalised eigenvalue problem [81, 82] given as

$$S_2 \Phi = \lambda S_1 \Phi. \quad (3\text{-}15)$$

Step 3: Finally, using the transformation Φ, recognising and classifying any performed gesture g_k is achieved in the transformed space. This is made possible by using Euclidean distance [4] as a measure that provides a distinction between different classes or categories. Any gesture g_k can be recognised and hence classified as

$$\arg\min_k d(g_k \Phi, \bar{x}_k \Phi) \quad (3\text{-}16)$$

where $\bar{x}_k$ in any class k is its centroid.

3.3.2 Applications of the LDA Technique in Areas Related to our Research

In this section we discuss various applications where LDA has been implemented. In particular, we are considering applications where LDA has been implemented in hand gesture recognition. This has helped us to review the most recent, relevant and feasible techniques that are applicable to hand gesture recognition.

a) 3D Hand Gesture Recognition based on Polar Rotation Feature and LDA

In their work [83], researchers implemented a hand gesture recognition system using LDA. The hand gestures in their system were obtained from a 3D laser scanner which creates depth data. During the system implementation, hand area segmentation, hole-filling and normalization are performed first, followed by extraction of a feature of the polar rotation distance via polar-coordinate transformation. Through implementation of a combination of PCA and LDA, the algorithm proved to be robust and accurate and achieved 96.7% recognition rate under a set of six different hand gestures.

b) Multi-feature based Hand Gesture Recognition

In [84], the researchers implemented an extensive method for recognising hand gestures [85-87]. They demonstrated that motion measurements related to the hand position, orientation and finger bending can be regarded as time-series data collections and be used for the recognition of hand gestures. In order to address the challenges of hand gesture recognition, given its multi-feature nature, a novel approach for identifying important features for each hand gesture class was implemented [84]. A two-stage comparison method with the implemented stratification of hand gesture categories based on their important features helped the approach to handle the available huge number of hand gesture categories. Finally, hand gesture comparison based on the subspace created by LDA of temporal features was implemented in such a way that rhythmic differences between hand gesture trials were minimised.

c) Classification of Hand Motions using LDA

In their study [88], LDA is employed as a machine learning classifier to recognise

six hand motions using surface electromyogram (SEMG) signals recorded from eight muscles of the right hand. They obtained 24 features per muscle. Three feature sets, i.e. the original feature, the features created by a discriminant analysis (DA), and features selected by a multiple regression analysis (MRA) [89, 90], were utilised during machine learning classification. LDA performed comparatively well with accuracy above 90.0%. Generally, the index finger extension (IFE) had higher classification accuracy than other hand motions. The probability of thumb opposition (TO) misclassified as a key pinch (KP) was 1.1%, that of hand grasp (HG) misclassified as four-finger flexion (FFF) was 1.0%.

d) Implementing LDA on a Wearable Human Machine Interface

In their research [91], machine learning classifiers such as LDA are implemented on a dataset. They implemented eight gaming hand gestures i.e. clapping, index finger flicking, finger snapping, coin flipping, shooting, wrist extension, wrist flexion, and fist making. All these gestures were recognised in real-time. Their system was based on wearable human machine interface mechanomyogram (MMG) signals [92-95]. Furthermore, a three-axis accelerometer was attached to a specific watch strap to estimate the MMG signals that were created by the end of the extensor digitorum muscle.

In their study, they obtained features from both the time signals and the coefficients of the wavelet packet decomposition (WPD). Sequential forward selection (SFS) was employed to identify the important features to improve the classification accuracy and minimise the processing time. LDA achieved an accuracy of more than 90.0%.

e) Hand Gesture Recognition Based on Time Domain Features and LDA

In their approach [96], a hand gesture recognition system is proposed where three channels of surface electromyograph (SEMG) signals were used to classify nine different hand gestures. In their implemented system, the time domain features, root mean square ratio, and autoregressive model were employed to obtain the features of the SEMG and compared with the time-frequency domain features. In addition, LDA was employed as a machine learning classifier and achieved an accuracy of 91.7%.

Compared with other related work [97-101], their approach used only three

sensors to identify nine hand gestures. The processing time of the LDA classifier greatly reduced and therefore the system can be implemented in the real-time recognition applications.

3.4 Support Vector Machine (SVM)

SVM is a learning method that operates on the basis of the statistical learning theory [102, 103]. It is a non-linear classifier [35] which is often reported as a superior classifier compared to other machine learning techniques. It is an efficient classifier broadly applied in pattern recognition [104]. Despite the fact that SVM operates a binary classification, multiple classifiers can be merged to implement multiclass classification [36]. The main idea behind the SVM is to project the input data on to higher dimensional feature space that is not linearly associated with the input space and establish a hyperplane separating any two given classes of feature space with a larger margin [4].

3.4.1 How SVM Solves the Problem

Our objective is to create an SVM model after training a fraction of measured observations on our dataset with an aim of correctly recognising and classifying test samples of hand gestures.

If data points are linearly separable [35], the classifier can be expressed as

$$f(x) = \beta_o + \sum_{i=1}^{n} \alpha_i \langle x, x_i \rangle \quad (3\text{-}17)$$

where there are n parameters $\alpha_i, i = 1, \cdots, n$, one for each training observation. $\langle x, x_i \rangle$ represents the inner product of two observations defined as

$$\langle x, x_i \rangle = \sum_{j=1}^{P} x_j \, x_{ij}, \quad (3\text{-}18)$$

where P is the number of dimensions for every single measured observation.

In the process of estimating terms $\alpha_1, \cdots \alpha_n$ and β_o, we are required to compute $\binom{n}{2}$ inner products $\langle x_i, x_i' \rangle$ among the pairs of all training observations. To obtain $f(x)$, the inner product between any possible new point x and every training point x_i must be computed. However, it results in α_i being non-zero for any measured observations that constitute support vectors in the solution, i.e. if a training sample does not belong to

support vectors, then α_i becomes non-existent and equals zero [35].

If Δ is the set of indices of the support vectors, we can rearrange any possible solution function as

$$f(x) = \beta_o + \sum_{i\in\Delta} \alpha_i \langle x, x_i \rangle \quad (3\text{-}19)$$

which basically constitutes far fewer terms compared to Equation (3-17).

However, if data points are not linearly separable, it is easier to replace $\langle x, x_i \rangle$ with a generalisation of $K(x_i, x_i')$, where *K* represents a kernel [105]. A kernel can be defined as a mathematical relation that quantifies any similar features of two measured observations. Examples of mostly used kernels are linear where $K(x_i, x_i') = \sum_{j=1}^{p} x_j\, x_{ij}'$, polynomial where $K(x_i, x_i') = (1 + \sum_{j=1}^{p} x_j\, x_{ij}')^d$, radial where $K(x_i, x_i') = \exp(-\gamma \sum_{j=1}^{p} (x_j x_{ij}')^2)$, and sigmoid where $K(x_i, x_i') = (\tanh(1 + \sum_{j=1}^{p} x_j x_{ij}'))$. It is important to note that d is a positive integer that represents the degree of a polynomial kernel and γ is a positive constant.

Finally, the class label for the test gesture x^* can be predicted by the sign of

$$f(x^*) = sgn\ (\beta_o + \sum_{i\in\Delta} \alpha_i \langle x^*, x_i \rangle). \quad (3\text{-}20)$$

3.4.2 Relevant Applications of SVM Technique in Areas Related to our Research

In this section we review some applications where SVM has been implemented. We have considered only those applications that are closely related to our research work where researchers implemented SVM in hand gesture recognition.

a) Application of SVM to recognise hand gestures for hand therapy

Here SVM was implemented to classify and recognise isolated hand gestures in order to facilitate hand therapy for patients recovering from stroke or similar illness [4]. These isolated hand gestures were then combined to form hand gesture sequences that simulated hand exercises performed by the patients. The experimental results demonstrated that the SVM model was fully sensitive to a choice of any of the kernels together with their respective parameters.

b) Static hand gesture recognition using a mixture of features and SVM

In their work [106], they proposed a vision-based application for static hand gesture recognition. The application considers images of bare hands and permits recognition of any hand gesture in illumination and when slight rotated. The implemented application comprises three stages: pre-processing, feature extraction, and classification. During classification, features are used as input to a multiclass SVM model in order to recognise static hand gestures. The recognition accuracy on three different databases were 99.5%, 93.6%, and 98.3%.

c) Real-time hand gesture recognition using a bag-of-features and SVM

This research work considers a real-time system aimed at interacting with an application or video game by making use of hand gestures [107]. In addition, the system can detect and track a bare hand in a cluttered background employing skin detection and a hand posture contour comparison algorithm following face subtraction, recognising hand gestures using a bag-of-features and SVM and creating a grammar that develops gesture commands to manage an application.

In the training phase, following derivation of the keypoints for each training image employing the scale invariance feature transform (SIFT), a vector quantisation approach projects keypoints from each training image to a unified dimensional histogram vector (bag-of-words) following K-means clustering. The histogram is considered as an input for a multiclass SVM to create a training model. In the testing phase, for each frame taken from a webcam, the hand is detected by making use of their algorithm, afterwards the keypoints are derived for each small image that comprises the detected hand gesture only and is put into the cluster model to project them in a bag-of-words vector that is eventually put into the multiclass SVM training model in order to recognise a hand gesture.

d) Hand gesture recognition using PCA and SVM

In their research [108], they implement techniques to create a user independent finger and palm gesture recognition application taking into account related challenges such as illumination variations, difference in user hand shape, and higher inter class commonalities. In the implemented gesture recognition application, performance

evaluation was performed using pre-trained AlexNet characteristics. The deep characteristics were derived from fully connected (FC) layers for instance 'FC6' and 'FC7' of pre-trained AlexNet.

A SVM based classifier having a linear kernel is employed in order to recognise hand gestures. The greatest recognition accuracy is estimated by making use of the deep feature derived from FC6 and FC7 separately and an integration of the feature vector with the SVM classifier. Finally, PCA is employed to reduce the feature dimension of deep characteristics in order to improve hand gesture recognition accuracy.

e) Hand Gesture Recognition using Error Correction Output Code (ECOC) and SVM

In [109], the researchers proposed a low-cost capacitive sensor device [110-114] to recognise hand gestures. Specifically, they implemented a system of a wearable capacitive sensor units to obtain the capacitance values from the electrodes fixed on finger bones. They extracted 15 features for hand gesture classification training and testing tasks. They applied an error correction output code support vector machine (ECOC-SVM), and then introduced a feature compression technique obtained from correlation analysis to minimise the complexity of SVM and a recognition rate of 97.0% was achieved [109].

f) Application of SVM to Recognise Hand Gestures from an EMG Signal

SVM is implemented to recognise hand gestures. In particular, recognition of hand gestures is achieved from the given Electromyography (EMG) signal, obtained via a sensor-based band [115]. In order to minimise noise artifacts, the raw EMG signal has to pass through pre-processing steps. Subsequently, eight kinds of time-domain features are obtained from the raw EMG signal, then a feature matrix is generated. SVM is employed as a hand recognition technique and implemented in MATLAB 2019a and achieves an accuracy of 83.0%.

3.5 Dynamic Time Warping (DTW)

DTW has been in existence for decades, employed mainly to obtain the optimal alignment of two signals [21]. DTW derives the distance metric between each possible pair of points out of two signals with reference to their respective feature values [116]. It

employs these distances to derive a cumulative distance matrix and obtains the optimal path through the matrix [39]. This path illustrates the perfect warp i.e. the synchronisation of the two signals that creates the distance between their synchronised points to be as minimal as possible.

Usually, the signals are normalised and smoothed prior to the computation of the distances between points. DTW has been employed in a number of fields, for instance speech recognition, data mining, and movement detection [117, 118]. Earlier research in the field of DTW mostly examined accelerating the algorithm, having a complexity which is quadratic in terms of the length of the series that are being compared. Due to such complexity related challenges, researchers have devised means to mitigate those challenges. For example, constraints to the DTW implementation have been applied [119], some form of approximations of the DTW algorithm have been implemented in [120] and lower bounding approaches have been suggested in [121].

3.5.1 Description of DTW

The goal of DTW is to make a comparison between two time-dependent sequences: a query sequence, $x = (x_1, x_2, \cdots, x_N)$ of length $N \in \mathbb{N}$ and a reference sequence $y = (y_1, y_2, \cdots, y_M)$ of length $M \in \mathbb{N}$ [119]. In the subsequent discussion, to index the components in x and y, we shall employ the symbols $i = 1 \cdots n$ and $j = 1 \cdots m$ respectively. In order to establish a comparison between sequences, a local cost measure is needed [21, 122, 123]. This measure can be referred to as a local distance or dissimilarity measure [122]. To derive this measure, a non-negative function *f* is established between any pair of components x_i and y_i as illustrated in Equation (3-21).

$$\mathrm{d(i,j)} = f(x_i, y_j) \geq 0 \tag{3-21}$$

Generally, if $\mathrm{d(i,j)}$ is of a small value, then x and y are similar to each other and when $\mathrm{d(i,j)}$ is of a greater value, then *x* and *y* are different. The most frequently used distance function is the Euclidean distance, other distance functions are squared Euclidean, Manhattan, Gower coefficient etc. [117-122]. Let us consider Cartesian coordinates. Given $x = (x_1, x_2, \cdots, x_n)$ and $y = (y_1, y_2, \cdots, y_n)$ are two points in Euclidean n-space, the

distance d from x and y is defined in the subsequent Euclidean distance operation,

$$d(\mathbf{x},\mathbf{y}) = d(\mathbf{y},\mathbf{x}) = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + \cdots + (x_n - y_n)^2} \quad (3\text{-}22)$$

Using one of the distance functions aforementioned, a local cost measure for every pair of components of the respective sequences X and Y is determined [122]. This generates a cost matrix $C \in \mathbb{R}^{N+M}$illustrated by $C(n,m) \coloneqq d(i.j)$ [122]. The eventual goal is to obtain an alignment between *X* and *Y* in a manner that the global cost is minimum [118-121]. It is important to notice that an optimal alignment runs roughly along the diagonal of the cost matrix *C* [120]. Generally obtaining an optimal alignment requires getting the warping curve $\phi(k)$, where $k = 1 \cdots T$ [122].

$$\phi(k) = (\phi_x(k), \phi_y(k)) \quad (3\text{-}23)$$

where $\phi_x(k) \in \{1 \cdots N\}$ and $\phi_y(k) \in \{1 \cdots M\}$.

The warping mathematical operations ϕ_x and ϕ_y realign both time indices of *x* and *y* respectively. The mean aggregated distortion between the warped time-series *x* and *y* can be obtained as follows:

$$d_\phi(x,y) = \sum_{k=1}^{T} \frac{d(\phi_x(k),\phi_y(k))m_\phi(k)}{M_\phi} \quad (3\text{-}24)$$

where $m_\phi(k)$ is a non-negative weighting factor that regulates the input of every temporary distortion $d(\phi_x(k), \phi_y(k))$ [122]. Considering that this is normally associated with the slope of the local path constraints that will be shortly described in Section 3.4.2. This can also be regarded as slope weighting function. The denominator M_ϕ employs an entire normalisation to the aggregated distortion to obtain a mean path distortion which does not consider the lengths of any pair of sequences that are being compared [117-122]. Finally, dynamic programming is employed to obtain the optimal alignment ϕ in such a way that,

$$D(x,y) = \min d_\phi(x,y). \quad (3\text{-}25)$$

3.5.2 Warping Constraints of DTW

Generally, the various ways of feasibly warping paths along the grid of the cost matrix are numerous. This suggests that the search space must be limited. This limitation is

also essential to provide an appropriate time alignment between two sequences that are being compared. Following are the usual warping constraints that are considered essential [122-124].

a) Boundary constraints

The boundary constraints require application of the subsequent conditions [122]:

$$\phi_x(1) = \phi_y(1) = 1 \tag{3-26}$$

$$\phi_x(T) = N \tag{3-27}$$

$$\phi_y(T) = M \tag{3-28}$$

These guarantee that the time sequences' initial position and final position match each other. Consequently, the alignment does not take into account incompletely one of the sequences [120-124]. On the other hand, during partial time-series matching, these conditions can be neglected. The fundamental concept of boundary constraints was invented from the understanding that speech patterns under comparison generally possess distinct endpoints that indicate the first and the last frames of the pattern [122]. Hence, the endpoint data is required to be integrated to achieve an exact match.

b) Monotonicity conditions

Monotonicity conditions are described in the following inequalities:

$$\phi_x(k+1) \geq \phi_x(k) \tag{3-29}$$

$$\phi_y(k+1) \geq \phi_y(k) \tag{3-30}$$

The above inequalities ensure that the time-series' time arrangement is maintained. Hence this avoids the alignment path from moving backwards in time. Fundamentally, negative slopes of a warp path are eliminated [122-124].

c) Local Continuity Constraints

The primary goal of local continuity or step-size constraints is to guarantee that no component in *X* and *Y* is neglected, otherwise possible loss of data could happen [122]. Hence, a discontinuous warp path is eliminated [124]. Usually, local continuity constraints

could be implemented in different forms [122]. Considering the directions of matches between *i* and *j* that are permitted, they can be classified as symmetric or asymmetric [122]. An example of symmetric local constraints is proposed and implemented by [124] as follows:

$$\phi_x(k+1) - \phi_x(k) \geq 1 \quad (3\text{-}31)$$

$$\phi_y(k+1) - \phi_y(k) \geq 1 \quad (3\text{-}32)$$

The above set of constraints in the inequalities (3-31) and (3-32) is known as symmetric since it permits an unrestricted number of components of the query *X* to match with a single component of the reference *Y* , and vice versa [124, 125]. It is important to note that for an asymmetric step pattern, multiple components of the query sequence *X* are permitted to match with the same component in the reference sequence *Y*, but not vice versa [124, 125].

d) Global Path Constraints

Along with the local path constraints, global path constraints, also regarded as "windowing", can be employed to the warping functions to identify areas in the *(i, j)* plane where warping paths are not expected to exist [120-124]. It guarantees that the warped path is always as close to the diagonal as possible. Researchers in [125] propose and implement the subsequent adjustment window condition so that the time-axis variation does not generate too much of a timing difference:

$$\left|\phi_x(k) - \phi_y(k)\right| \leq r \quad (3\text{-}33)$$

where r represents a suitable positive integer known as the window length.

3.5.3 Dynamic Programming Algorithm

To compute the equation (3-25), iterative dynamic programming (DP) algorithms are employed [122]. In the subsequent equation, an algorithm that implements a symmetric step-pattern is illustrated.

Starting condition: $g(1,1) = d(1,1)$

DP equation:

$$g(i,j) = min\begin{bmatrix} g(i,j-1) \\ g(i-1,j-1) + 2d(i,j) \\ g(i-1,j) + d(i,j) \end{bmatrix} \quad (3\text{-}34)$$

In order to implement an asymmetric step-pattern, a change is required to be made on Equation (3-34) as illustrated in the subsequent equation [122].

$$g(i,j) = min\begin{bmatrix} g(i-1,j) + d(i,j) \\ g(i-1,j-1) + d(i,j) \\ g(i-1,j-2) + d(i,j) \end{bmatrix} \quad (3\text{-}35)$$

It is important to note that there is no need to specify and limit conditions for the global path constraints because normally windowing is not implemented. Equation (3-34) or (3-35) should be recursively computed in an ascending order with reference to coordinates *i* and *j*. The algorithm begins from the starting condition at *(1,1)* and terminates at *(N,M)* [116-125].

3.5.4 Problem Formulation for Comparing Dynamic Hand Gestures using DTW and MDTW

Our objective is to establish how similar or different a query dynamic hand gesture is in comparison to a reference dynamic hand gesture whilst compensating for differences in the duration of gestures, rotation of the hand, reasonable distance from LM sensor etc. Firstly, we define a dynamic hand gesture. Following that, we illustrate how traditional DTW works considering that the feature vector constitutes only one measured feature. We then extend DTW to Multi-dimensional Dynamic Time Warping (MDTW) so that it suits our proposed framework.

a) Definition of a Dynamic Hand Gesture

A dynamic hand gesture is represented as a sequence of hand positions changing over time and each position is described by a set of finger joint angles. Mathematically, a dynamic hand gesture is described using a set of *P* measured finger joint angles that evolve over time i.e. a dynamic hand gesture g_t at an instant *t* is described by $g_t = \{\theta_{1t}, \theta_{2t}, \cdots, \theta_{Pt}\}$.

b) Problem Formulation using DTW

We let a query dynamic hand gesture and a reference dynamic hand gesture be represented by x and *y*, respectively [126]. Specifically, $\mathbf{x} = (\mathrm{x}_1, \mathrm{x}_2, \cdots, \mathrm{x}_\mathrm{M})$ where $M \in \mathbb{N}$ and $\mathbf{y} = (\mathrm{y}_1, \mathrm{y}_2, \cdots, \mathrm{y}_\mathrm{N})$ where $N \in \mathbb{N}$. From now onwards, we use i and j to represent an entry in time-series x and y, respectively. We define the Euclidean distance between any two samples in time-series x and y as [126],

$$d(i,j) = \sqrt{(x_i - y_j)(x_i - y_j)} \qquad 1 \le i \le M, \qquad 1 \le j \le N. \tag{3-36}$$

A two-dimensional N by M cost matrix D is created and every individual value of the $D(i,j)$ is determined as follows:

$$D(1,1) = d(1,1) \tag{3-37}$$

$$D(1,j) = D(1,j-1) + d(1,j) \qquad 2 \le j \le N \tag{3-38}$$

$$D(i,1) = D(i-1,1) + d(i,1) \qquad 2 \le i \le M \tag{3-39}$$

$$D(i,j) = d(i,j) + min\begin{Bmatrix} D(i,j-1) \\ D(i-1,j-1) \\ D(i-1,j) \end{Bmatrix} \quad 2 \le i \le M, \quad 2 \le j \le N. \tag{3-40}$$

Then a warping path $w = w_1, w_2, \cdots, w_r$ is an adjacent collection of some matrix constituents that are always close to the diagonal. These matrix elements when added together are equivalent to a minimum-distance of a warp path.

The minimum-distance of a warp path provides a mapping between *x* and *y* that fulfils the following requirements [124]:

- Boundary requirements: Given $w_1 = (1,1)$ and $w_r = (M,N)$ where *r* is the length of the warping path.
- Continuity conditions: Given $w_z = (a,b)$ and $w_{z-1} = (a',b')$, then $a - a' \le 1$ and $b - b' \le 1$ must be fulfilled.
- Monotonicity requirements: Given $w_z = (a,b)$ and $w_{z-1} = (a',b')$, then $a - a' \ge 0$ and $b - b' \ge 0$ must be satisfied.

We can now illustrate how to obtain a minimum-distance warp path with an example. Given $x = (1,2,4,3,5,3,2,3,2,5)$ that represents a query series and $y =$

$(1,1,2,4,3,5,3,2,3,2)$ that represents a reference series, we can construct a cost matrix D. The minimum-distance of a warp path is traced through the matrix constituents from $D(1,1)$ to $D(M,N)$, highlighted in pink, as shown in Figure 15. The minimum-distance of the warp path using Euclidean distance as a metric is three. The above DTW formulation can only be implemented if a framework considers a single feature alignment i.e. one dimensional measurements [126].

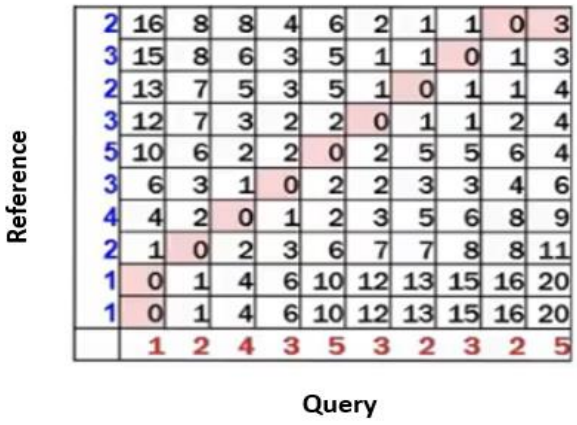

Figure 15: Cost matrix and the minimum-distance of the warp path.

c) Problem Formulation using MDTW

Since in our framework, we are considering up to 14 joint angles to describe a dynamic hand gesture at an instant, we employ MDTW. For MDTW, the two time-series X and Y have to be initially created as multi-dimensional matrices where each row represents the time-series of a single measured feature and each column represents all the measured features at a given instant. The matrices X and Y can now be written as shown below. M and N are samples of a dynamic query hand gesture and a dynamic reference hand gesture, respectively, and P is the number of measured features being considered.

$$\mathbf{X} = \begin{bmatrix} x_{1,1} & \cdots & x_{1,M} \\ \vdots & \ddots & \vdots \\ x_{P,1} & \cdots & x_{P,M} \end{bmatrix}$$

$$\mathbf{Y} = \begin{bmatrix} y_{1,1} & \cdots & y_{1,N} \\ \vdots & \ddots & \vdots \\ y_{P,1} & \cdots & y_{P,N} \end{bmatrix}$$

We now define the Euclidean distance between X and Y as

$$d(i,j) = \sqrt{(X_i - Y_j)^T (X_i - Y_j)}. \quad (3\text{-}41)$$

where X_i are all fourteen measured joint angles at ith frame of X and Y_j are the fourteen measured joint angles at jth frame of Y.

The entries in the cost matrix D can then be determined as shown in Equations (3.37), (3.38), (3.39) and (3.40). Then the MDTW algorithm searches for a minimum-distance warp path that runs close to the diagonal line from $D(1,1)$ to $D(M,N)$. The warping path must fulfil all three requirements as described in the case for DTW. If the minimum-distance of a warp path is of a lower value, then the two time-series, i.e. the two dynamic hand gestures being compared, are similar, otherwise the two dynamic hand gestures being compared are effectively different to a quantifiable extent.

3.5.5 Relevant Applications of DTW and MDTW in Areas Related to our Research

The following text provides examples where researchers have implemented DTW. Specifically, we are considering applications of DTW that are closely related to our research.

a) Using Adaptive DTW to Recognise Natural Gestures

The work describes a novel technique considered to recognise 3D dynamic composed gestures [114]. During experimental analysis, every gesture is illustrated by association of angle variations defined as a vector. A composed gesture can be defined as a sequence of two simple gestures or more carried out in a sequence. Examples of simple gestures included come, recede, point to the right, point to the left and stop. For the purpose of recognising all the composed gestures accurately, they combined DTW with an Adaptive Sliding Window. In addition, DTW was implemented to compare the reference gestures to sequences provided by the adaptive window. Finally, the reference gesture that

generates a lower distance metric is regarded as the source class of the tested gesture.

b) Implementing DTW in Post-Stroke

The objective of this research was to evaluate the performance of a real-time model of DTW in order to recognise motor exercises [21]. When provided with a potentially partial input stream of data and a reference time-series, the proposed and implemented DTW technique calculates both the size of the prefix of the reference that best matches the input, and the dissimilarity between the matched sections. In addition, the implemented technique was able to generate real-time feedback to neurological patients performing motor hand exercises. This real-time DTW approach is appropriate during the classification of condensed quantitative time-series, despite the existence of noise.

c) Implementing DTW for Recognising Static and Dynamic Hand Gesture

Research conducted by [127] proposes a natural hand gesture user interface that monitors and identifies hand gestures in real-time based on depth data obtained from a Kinect sensor. In particular, DTW is employed in selecting gesture candidates and recognising hand gestures by establishing a comparison of an observed gesture with various pre-recorded reference hand gestures.

The comparison of their results with state-of-the-art techniques such as in [128, 129] illustrates that the proposed approach performs better than most of the systems for the static hand gesture recognition of sign digits and is similar with regard to performance of the static and dynamic hand gesture recognition of popular signs used in the sign language alphabet. Their system achieves a recognition rate of 92.4% on average; however, to improve this recognition rate, a different version of DTW [130] needs to be taken into consideration when performing comparison tasks.

d) Implementing Weighted DTW for Time-series Classification

These researchers propose and implement a novel distance measure, referred to as a weighted DTW (WDTW), that is a penalty-based DTW [131]. Their technique penalises points having greater phase difference between a reference point and a query point with the purpose of preventing minimum distance distortion due to outliers. A novel weight

operation, referred to as the modified logistic weight function (MLWF), is also implemented to regularly allocate weights as a function of the phase difference for both reference point and query point.

By employing various weights to neighbouring points, the proposed and implemented algorithm can improve the estimation of similarity between any two specified time-series. They employ the proposed and implemented technique to other forms of DTW for example derivative dynamic time warping (DDTW). In addition, they propose and implement the weighted form of DDTW. The experimental results illustrate that the implemented technique can achieve improved outcomes for time-series classification and clustering data analysis.

e) Implementing DTW for Signature Verification

One of the limitations of DTW was reported when implementing signature verification system [132-134]. The implementation involves obtaining correspondence and the similarity of two planar curves. The limitation was that in portions of the curves at locations with sparse sampling, insufficient resolution in the matching procedure was experienced attributed to the fact that DTW matches only individual samples instead of continuous curves. The authors in [132] further proposed that a feasible solution to this limitation is to oversample the curves. Oversampling can be implemented by employing a spline interpolation prior to curve matching.

f) Applying DTW to Measure Time-series Similarity

In [135] DTW is implemented to measure time-series similarity, perform classification, and identify corresponding portions between two time-series [136-139]. One strategy that was implemented applies a multilevel technique which repeatedly predicts a measurement from a coarse resolution and improves the predicted measurement. The strategy possesses linear time and space complexity and partially solves the issues of quadratic time and space complexity experienced by DTW when implemented with large time-series datasets [135].

g) Applying Improved DTW to Recognise Dynamic Hand Gestures

In [140], the researchers implemented a dynamic hand gesture recognition system using an improved DTW algorithm. The 3D positions of a human hand are acquired after the analysis of the depth information, which is obtained through a Kinect sensor. Eight points are chosen as the hand motion characteristics, and the mathematical model of the hand is created by the approach of weighted distance. In order to improve performance of DTW, the distortion threshold is used, and the path constraints are applied in training templates. The implemented approach demonstrates that the improved DTW algorithm provides a substantial improvement in both speed and accuracy.

h) Implementing DTW to Recognise 3D Hand Gestures

In their research [141], the authors propose dynamic time warping to recognise 3D hand gestures. In their approach, they split the time-series curve of a 3D hand gesture into different finger combinations, referred to as fingerlets, that can be learned or be set manually to characterise each gesture and to obtain inter-class variations.

The implemented DTW approach finds the minimal path to warp two fingerlets, that are from one observation point and the specific class, respectively. In addition, the hand gesture recognition employs an ensemble of multiple DTW fingerlet distances to obtain better performance. Their approach was assessed on two 3D hand gesture datasets, and they demonstrated that the technique greatly improves hand gesture recognition.

i) Applying DTW to Recognise Hand Gestures Performing Air-written English Alphabet Capital Letters

DTW is implemented in a hand gesture system in [142]. Hand gestures are demonstrated as air-written English alphabet capital letters and the sensor component is a Leap Motion controller (LM). Through the application of DTW, the system can identify hand gestures that describe capital letters as two-dimensional values despite the input set and the reference template possessing a different number of points and, additionally, classify hand gestures by obtaining the lowest difference of the standard deviation between the input and templates. In their system, LM is employed to locate the position of the finger of the user, which acts as the "pen" in air-writing letters (hand gestures).

3.6 Summary

In this chapter, we discussed LM device, PCA, LDA, SVM, and DTW prior to describing how we have employed them in our own research. We first explained what an LM is, its interaction area, and its internal structure. We outlined the steps undertaken when calibrating the LM. We also discussed suggestions on how to make employing the LM feel natural and easy. Secondly, we described what PCA is, how it can be applied as a filtering mechanism and discuss how researchers have employed it in various applications. Thirdly, we illustrate how LDA is applicable to our research, and this is described in three steps in Section 3.3.1. Then, we describe applications of LDA in areas related to our research. We describe how SVM is relevant to our research problem in regard to hand gesture recognition in Section 3.4.1. Other applications of SVM in areas related to our research are also discussed. Next, we discuss DTW in detail where we demonstrate that it is mainly employed to make a comparison between time-dependent sequences. We also discuss warping constraints of DTW, i.e. boundary constraints, monotonicity conditions, local continuity constraints, and global path constraints. We also illustrate the problem formulation for comparing dynamic hand gestures using DTW and MDTW. Finally, we discuss some pertinent applications of DTW in regard to hand gesture recognition.

4 Characterising and Reducing Absolute Errors (AEs) when Measuring Finger Joint Angles in LM using PCA

We have configured four parameters in our designed experiments to characterise finger joint angle absolute errors. These four parameters are elevation relative to the LM, lateral (side-to-side) position relative to the LM, forward-backward position relative to the LM, and rotation relative to the LM. Definitions and further explanation of these parameters are given in Section 4.2.1.

4.1 Motion Data Collection Procedure

We conducted a series of experiments to assess the characteristics of the LM in terms of accuracy based on parameters like elevation, lateral (side-to-side) position, forward-backward position, and rotation. We used an "artist's hand" placed above the LM. The artist's hand is more accurate than a human hand in performing static hand gestures as it can maintain a fixed posture as long as is necessary. Figure 16 (a) shows the artist's hand when clamped on the stand and placed at the centre above the LM within its interaction area. The LM is connected to the PC using a USB cable. The LM software must be started in order to record the frames of data.

A joint angle θ can be obtained from the two vectors $\vec{u}$ and $\vec{v}$ representing the directions of the bones that form the joint as

$$\theta = \arccos \frac{(\vec{\mathbf{u}} \cdot \vec{\mathbf{v}})}{(\|\vec{\mathbf{u}}\| \cdot \|\vec{\mathbf{v}}\|)}. \qquad (4\text{-}1)$$

Equation (4-1) and Figure 16 (b) show how to obtain a joint angle of a finger. If θ is the Distal Inter-Phalangeal (DIP) joint angle of the middle finger for example, then $\vec{u}$ is the direction vector of the intermediate bone and $\vec{v}$ is the direction vector of the distal bone.

We customise a Java application from the LM Software Development Kit (SDK) to obtain frames of data whenever we run it. From the frames of data, we extract direction vectors of the metacarpal bone, the proximal bone, the intermediate bone, and the distal bone for all five fingers. Figure 17 illustrates an example of a single frame of data.

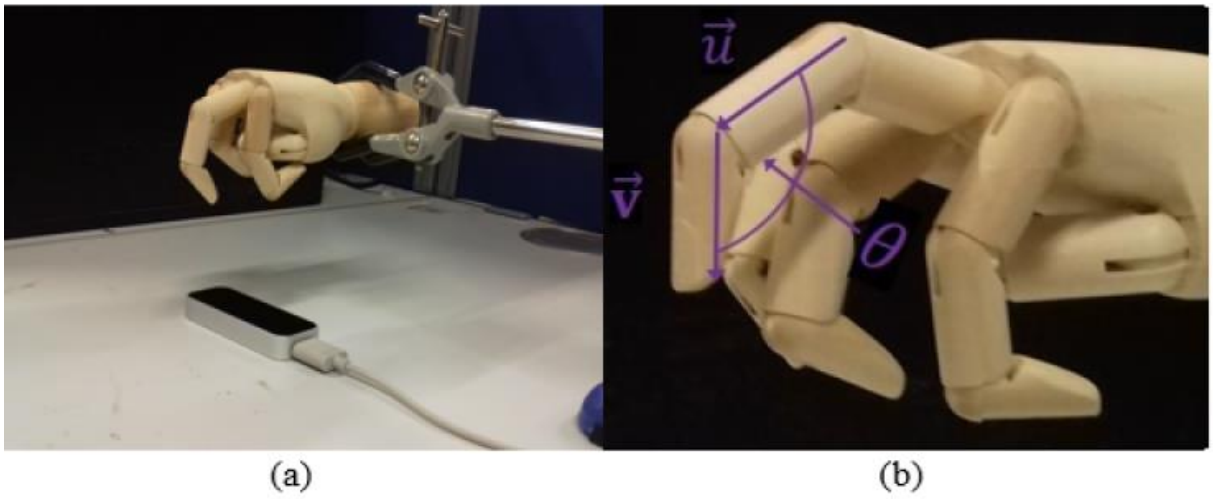

Figure 16: (a) Artist's hand at the centre of the LM. (b) How to obtain DIP joint θ.

For example, from the frame of data illustrated in Figure 17, we extract direction vectors of each bone. Lines 16 to 23 of the frame show the direction vectors of the middle finger. We show how we calculate joint angles of the middle finger from the frame. The direction vectors for metacarpal bone, proximal bone, intermediate bone, and distal bone are (0.169954, -0.148547, 0.974192), (0.176145, -0.269852, 0.946653), (0.352664, -0.637749, 0.685075), and (0.450808, -0.87293, 0.186452), respectively.

Using MATLAB software, we demonstrate how to obtain MCP, PIP and DIP joint angles from the four direction vectors. We name the vectors in the MATLAB command window as follows: a = [0.169954, -0.148547, 0.974192], b = [0.176145, -0.269852, 0.946653], c = [0.352664, -0.637749, 0.685075], and d = [0.450808, -0.87293, 0.186452]. We then calculate the joint angles in two steps as follows:

Step 1: In the command window of MATLAB, type, cosTheta = dot(a,b)/(norm(a)*norm(b));,

Step 2: In the command window of MATLAB, type, ThetaInDegrees = acosd(cosTheta);. This gives us 7.14^o for MCP joint. To obtain PIP and DIP joints we use vectors b and c, c and d, respectively, and follow the above two steps. We obtain 28.05^o and 32.52^o for the PIP and DIP joint respectively.

In addition, from Figure 17, the position and orientation of a hand can be obtained. The second line of the frame illustrates the palm position. From the third line up to the last line, orientations of the finger bones relative to the palm are illustrated.

We also calculate the Absolute Error (*AE*) between measured joint angles (x_i) and the default angle (d) for a given experiment of a parameter, e.g. 5 cm elevation is an example of an experiment of the elevation parameter. Other experiments of the elevation parameter are 10, 20, 30, 40 and 50 cm as illustrated in the first column of Table 1. Equation (4-2) illustrates how to obtain the *AE*.

$$AE = |d - \bar{x}| \tag{4-2}$$

where $\bar{x} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i}$ and n is the number of joint angles measured at a given experiment of a parameter. In our experiments the default angle was 90° and we used a protractor to measure it. We chose 90° as the default angle because it can be easily and accurately measured. We note that the value of n for all these the experiments is 5 since there are 5 measured finger joint angles for every experiment, i.e. the PIP joint of middle finger, MCP and PIP joint angles of ring finger, MCP and PIP joint angles of little finger. Five out of fourteen finger joint angles were sufficient for the purpose of absolute error related investigations since the errors are independent of the finger joints, hence there is no significant benefit considering all fourteen joint angles in this study. It is important to note that we compute the average, $\bar{x}$, before obtaining the absolute error to obtain a value that generalises well on how the set of the five joint angles perform as a whole for a given experiment of a parameter. In addition, the obtained average is more representative than a single value.

We also compute the 95% confidence interval (95% CI) which is defined as a range of values such that with a probability of 0.95, the range contains the true measured joint angle. The 95% CI is computed as

$$95\%\ CI = \bar{x} \pm t \cdot \frac{s}{\sqrt{n}} \tag{4-3}$$

where $\bar{x}$ is the mean, t is the student's t value obtained at the specific degrees of freedom,

n is the number of measured joint angles considered at a particular level of a parameter and S is the standard error defined as

$$S = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2}. \quad (4\text{-}4)$$

Frame id: 416130, timestamp: 210119142059, hands: 1, fingers: 5
Right hand, id: 190, palm position: (-1.20554, 96.1047, -19.9025)
TYPE_THUMB, id: 1900, length: 39.5858mm, width: 15.360008mm
TYPE_METACARPAL bone, start: (13.9199, 109.326, 23.9032), end: (13.9199, 109.326, 23.9032), direction: (0, 0, 0)
TYPE_PROXIMAL bone, start: (13.9199, 109.326, 23.9032), end: (17.4874, 136.235, -4.31132), direction: (-0.0911184, -0.687287, 0.720648)
TYPE_INTERMEDIATE bone, start: (17.4874, 136.235, -4.31132), end: (11.463, 151.811, -24.1047), direction: (0.232621, -0.601457, 0.764289)
TYPE_DISTAL bone, start: (11.463, 151.811, -24.1047), end: (0.502649, 158.628, -36.3281), direction: (0.616561, -0.383463, 0.687611)
TYPE_INDEX, id: 1901, length: 44.66816mm, width: 14.67188mm
TYPE_METACARPAL bone, start: (16.8458, 93.2922, 14.9589), end: (13.4702, 106.478, -41.1151), direction: (0.0584998, -0.22852, 0.97178)
TYPE_PROXIMAL bone, start: (13.4702, 106.478, -41.1151), end: (13.5721, 115.455, -72.4887), direction: (-0.00312056, -0.275093, 0.961413)
TYPE_INTERMEDIATE bone, start: (13.5721, 115.455, -72.4887), end: (9.04868, 126.194, -86.6755), direction: (0.246386, -0.584944, 0.772745)
TYPE_DISTAL bone, start: (9.04868, 126.194, -86.6755), end: (2.94657, 136.542, -91.5847), direction: (0.470202, -0.797379, 0.378282)
TYPE_MIDDLE, id: 1902, length: 50.895782mm, width: 14.409736mm
TYPE_METACARPAL bone, start: (9.83246, 86.8411, 13.3886), end: (0.532438, 94.9697, -39.9198), direction: (0.169954, -0.148547, 0.974192)
TYPE_PROXIMAL bone, start: (0.532438, 94.9697, -39.9198), end: (-5.91649, 104.849, -74.5781), direction: (0.176145, -0.269852, 0.946653)
TYPE_INTERMEDIATE bone, start: (-5.91649, 104.849, -74.5781), end: (-13.5208, 118.624, -89.3752), direction: (0.352064, -0.637749, 0.685075)
TYPE_DISTAL bone, start: (-13.5208, 118.624, -89.3752), end: (-19.9555, 131.084, -92.0366), direction: (0.450808, -0.87293, 0.186452
TYPE_RING, id: 1903, length: 48.937653mm, width: 13.711777mm
TYPE_METACARPAL bone, start: (1.55561, 81.9077, 13.4198), end: (-12.7265, 84.8525, -33.4961), direction: (0.290701, -0.0599402, 0.954935)
TYPE_PROXIMAL bone, start: (-12.7265, 84.8525, -33.4961), end: (-25.9679, 96.1778, -62.6188), direction: (0.390175, -0.333714, 0.858137)
TYPE_INTERMEDIATE bone, start: (-25.9679, 96.1778, -62.6188), end: (-35.9462, 112.748, -70.9019), direction: (0.474217, -0.787497, 0.393657)
TYPE_DISTAL bone, start: (-35.9462, 112.748, -70.9019), end: (-41.1778, 125.639, -68.0986), direction: (0.368641, -0.908342, -0.197533)
TYPE_PINKY, id: 1904, length: 38.366215mm, width: 12.179872mm
TYPE_METACARPAL bone, start: (-9.16808, 81.1627, 16.4287), end: (-26.2134, 78.6295, -25.6592), direction: (0.374794, 0.0556991, 0.925433)
TYPE_PROXIMAL bone, start: (-26.2134, 78.6295, -25.6592), end: (-40.4728, 74.4088, -48.0241), direction: (0.530925, 0.157152, 0.832719)
TYPE_INTERMEDIATE bone, start: (-40.4728, 74.4088, -48.0241), end: (-49.3378, 76.9763, -59.6657), direction: (0.59672, -0.172826, 0.783618)
TYPE_DISTAL bone, start: (-49.3378, 76.9763, -59.6657), end: (-57.1412, 83.0331, -68.2585), direction: (0.596019, -0.462614, 0.656315)

Figure 17: Illustration of a single frame of data.

4.2 Characterising and Reducing Absolute Errors (AEs) in Finger Joint Angle Measurements

In this section, we present the experimental setup and results for characterising and reducing absolute errors in finger joint measurements. All the measurements are obtained when a hand is placed above the LM in its area of interaction. The interaction area of the LM is described in Section 3.1.1.

4.2.1 Experimental Set-up to Characterise Absolute Errors (AEs)

We present an experimental set-up for exploring joint angle absolute errors. We arrange an experimental set-up in four different ways to test the accuracy of the LM based on elevation, lateral (side-to-side), forward-backward, and rotation movements of the hand relative to the LM. We have chosen four parameters because the first three parameters are the only possible situations where a hand performing gestures can move away from the field of view (interaction area) of LM. The fourth parameter, i.e. rotation movements of the hand, can cause finger occlusion depending on how the rotation is performed. Movements away from the field of view of the LM and rotation of the hand performing gestures cause absolute errors in the LM. We briefly explain the experimental set-up for all four parameters.

(a) Elevation Relative to the LM

By elevation relative to the LM, we mean the vertical distance from the surface of the LM to the palm of the artist's hand. Figure 18 illustrates an example of this type of positioning.

We varied elevation from 5 to 50 centimetres (cm). In all the elevation experiments, the artist's hand remained aligned with the centre of the LM, i.e. at 0 cm for both lateral (side-to-side) and forward-backward positions and was not rotated, it remained in the horizontal position with the palm facing downwards. The kinematic variables are joint angles measured in degrees and we set five known joint angles to 90^{o} for all the experiments.

(b) Lateral (side-to-side) Position Relative to the LM

By lateral (side-to-side) positioning, we mean static hand gestures were performed at the centre, to the left and to the right relative to the LM. Figure 19 illustrates some of these positions.

The artist's hand was held at 0, 5, 10, 15 and 20 cm to the right and left of the LM centreline. The elevation was maintained at 10 cm from the surface of the LM to the palm of the artist's hand and it was not rotated. It remained in the horizontal position with the palm facing downwards.

Figure 18: The artist's hand at an elevation of 40 cm from the LM.

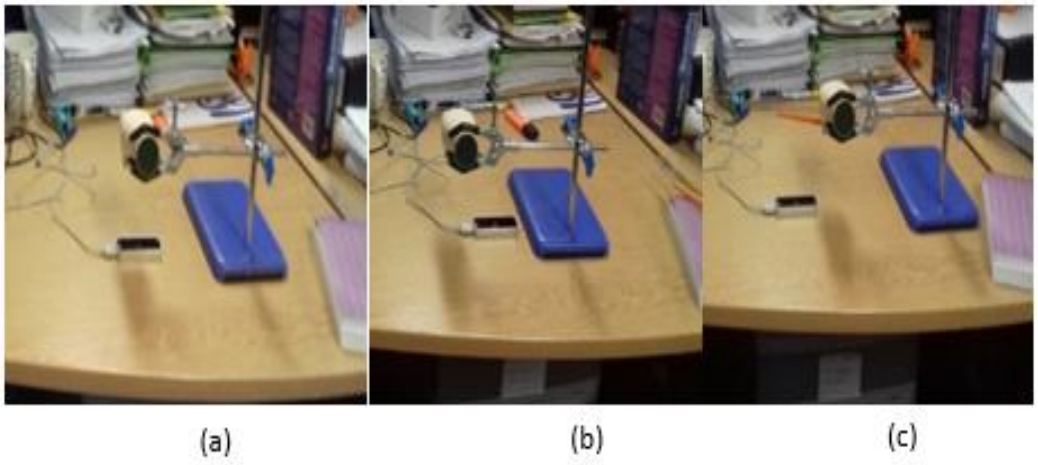

Figure 19: Lateral (side-to-side) position, (a) At the centre, (b) to the left, (c) to the right of the LM.

(c) Forward-Backward Position Relative to the LM

The forward position relative to the LM means the artist's hand was in front of the LM when performing a static hand gesture. A backward position relative to the LM means that the artist's hand was behind of the LM when performing a static hand gesture. These positions are illustrated in Figure 20.

The artist's hand was held at 0, 2, 4, 6, 8 and 10 cm to the front and behind the LM centreline. The elevation was maintained at 10 cm from the surface of the LM to the palm of the artist's hand and it was not rotated. It remained in the horizontal position with the palm was facing downwards.

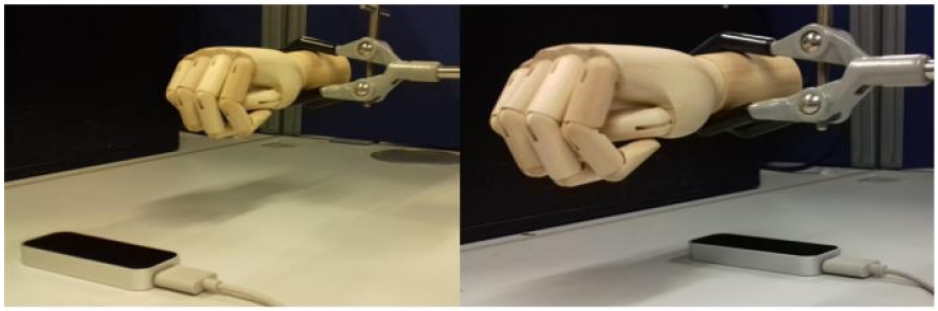

Figure 20: Forward-backward position relative to the LM: (a) Forward and (b)

(d) Rotation Relative to the LM

For the rotation, a static hand gesture using the artist's hand is configured where the hand is slightly altered as shown in Figure 21. All the alterations are made relative to the LM. Figure 21 shows rotation of the artist's hand relative to the LM as follows: (a) performing a static gesture at vertical position when the thumb is down, (b) at an angle of 45^{o} when the thumb is up, (c) at an angle of 45^{o} when the thumb is down, (d) at vertical position when the thumb is up, (e) when the palm faces upwards, and (f) when the palm faces downwards.

The elevation was maintained at 10 cm from the surface of the LM to the palm of the artist's hand and it was held along the centreline of the LM in all rotations. This meant that the artist's hand was held at 0 cm for both lateral (side-to-side) and forward-backward positions.

4.2.2 Results of Characterising Absolute Errors (AEs)

We performed 6 times (trials) for every experiment of a given parameter setting (the parameters are elevation, lateral (side-to-side), forward-backward, and rotation as

explained in Section 4.2.1) and obtained the average which was recorded. The recorded values in the second, third, fourth, fifth, and sixth columns of Tables 1, 2, 3, and 4 are the averages of these 6 trials. Experiments of the elevation parameter are considered for 5, 10, 20, 30, 40, and 50 cm as illustrated in the first column of Table 1. Experiments for lateral (side-to-side), forward-backward, and rotation are the first columns of Tables 2, 3, and 4, respectively. These 6 trials were performed on different days and on different times of a particular day i.e. day and night. It is important to notice that the results we obtained were achieved by carefully following the guidelines associated with the LM SDK. We have identified the four parameters of interest, i.e. elevation, lateral (side-to-side), forward-backward, and rotation, as described in Section 4.2.1. Since absolute errors (AEs) are experienced as a hand is moved away from the centreline of the LM and upwards above its interaction area, we investigated how absolute errors vary with these four parameters. For example for the elevation parameter, the absolute errors were investigated at 5, 10, 20, 30, 40 and 50 cm.

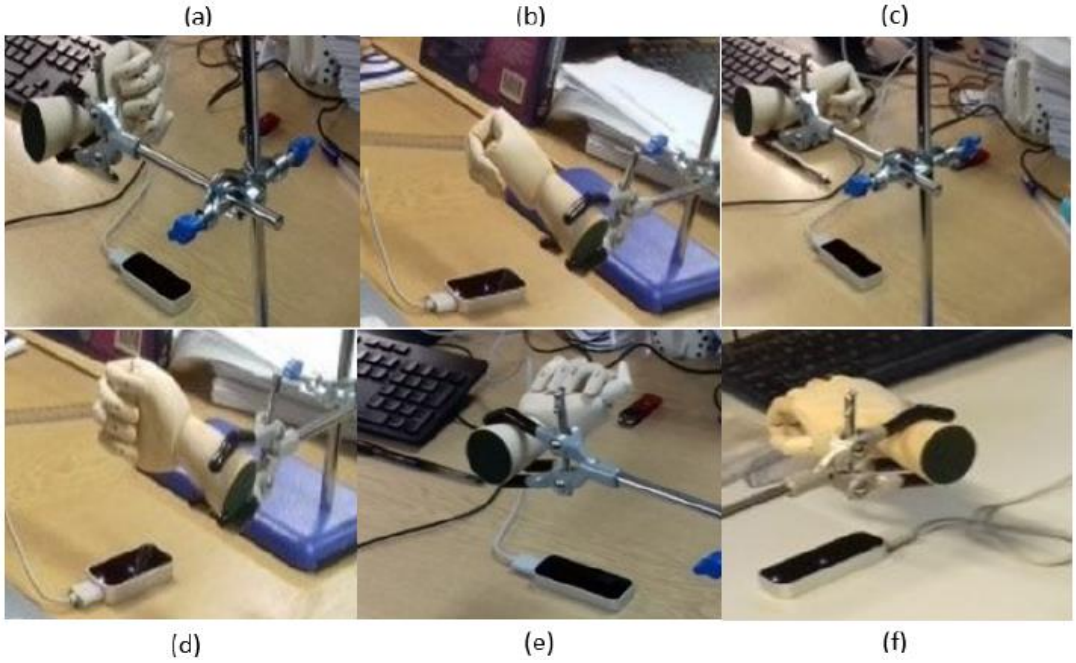

Figure 21: Rotation relative to the LM.

(a) Elevation Experimental Results

In all the elevation experiments, the artist's hand remained aligned with the centre of the LM i.e. at 0 cm for both lateral (side-to-side) and forward-backward positions and was not rotated, it remained in the horizontal position with the palm facing downwards.

Table 1 and Figure 22 show results obtained for the elevation experiment. Roughly, finger joint angles obtained at the lower elevations relative to the LM were closer to the default angle of 90^o. The finger joint angles at elevations of 5 cm, 10 cm, 20 cm and 30 cm registered absolute error of less than 10^o showing little variation. The finger joint angles measured at higher elevations experienced greater variation from the default angle of 90^o. This can be clearly seen from the elevations of 40 cm and 50 cm.

Generally, absolute errors increases as the elevation relative to the LM increases. However, there could be a better range of elevations where measured joint angles are obtained with smaller absolute errors. According to this experiment, we observe that this range is established when hand exercises are performed at an elevation of less than 30 cm relative to the LM.

The 95% confidence intervals reveal that elevations of 5, 10 and 20 cm illustrate an appropriate range which is close to the default angle of 90^o. On the other hand, elevations of 30, 40 and 50 cm reveal a much bigger range that shows that the lower limits of the respective confidence intervals deviate much more from the default angle of 90^o.

Measured joint angles at PIP joint of the middle finger experienced more variation from the default angle compared to MCP joint of the ring finger that experienced less variation. This could be attributed to occlusion likely to be experienced by a middle finger. In Figure 22, PIP(3) is PIP joint at the middle finger, MCP(4) and PIP(4) are MCP and PIP joints at the ring finger, and MCP(5) and PIP(5) are MCP and PIP joints at the little finger.

From Figure 22, we notice that elevations greater than 30 cm produce much variation from the default angle. In this experiment, elevations of 40 and 50 cm produced a variation of approximately 25.0^o in the worst case scenario. Thus for accurate measurements, users using LM for hand exercises, should try to avoid elevations that are

greater than 30 cm.

Table 1: Results for Varied Elevation.

Elevation (cm)	Middle	Ring		Little		$\bar{x}$ (deg)	AE (deg)	95% CI (deg)
	PIP (deg)	MCP (deg)	PIP (deg)	MCP (deg)	PIP (deg)			
5	84.73	85.95	81.35	81.06	89.93	84.60	5.40	[80.07, 89.14]
10	83.46	86.72	85.64	86.02	89.23	86.21	3.79	[83.63, 88.8]
20	84.36	88.19	83.69	82.81	86.28	85.07	4.94	[82.38, 87.76]
30	76.83	85.95	78.81	75.98	82.83	80.08	9.92	[74.85, 85.31]
40	64.57	75.96	83.95	75.34	72.01	74.37	15.63	[65.66, 83.08]
50	69.53	80.14	76.39	73.42	78.31	75.56	14.44	[70.36, 80.76]

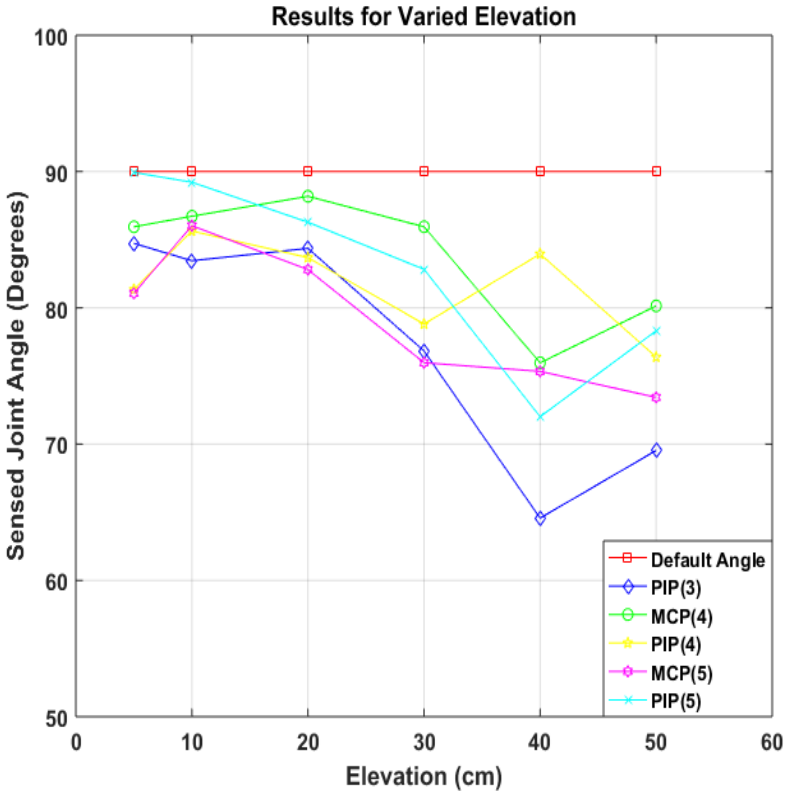

Figure 22: Sensed Joint Angle for Elevation.

(b) Lateral (side-to-side) Experimental Results

The artist's hand was held at 0, 5, 10, 15 and 20 cm to the right and left of the LM centreline. The elevation was maintained at 10 cm from the surface of the LM to the palm of the artist's hand and it was not rotated. It remained in the horizontal position with

the palm facing downwards.

Table 2 and Figure 23 show results obtained for the lateral (side-to-side) experiment. Lowest absolute error was obtained when the artist's hand was at the centre of the LM when performing static hand gestures and this was 4.33°. The highest absolute errors were obtained at lateral position of 20 cm to the right of the LM and 20 cm to the left of the LM which were 16.97° and 16.20°, respectively. Generally, the joint angles obtained at the positions further from the LM registered more variation of joint measured angles from the default angle than those that were close to the LM. This is evident from the joint measured angles obtained at 20 cm to the right and left of the LM.

Absolute errors roughly increases as the artist's hand is moved away from the centre of the LM. In this experiment, we have established that the best position to place the hand to perform hand gesture/exercises when using the LM is along the centreline or roughly up to 10 cm on either right or left of the device. This is because all the measured angles when it was at the centre of the LM or closer are fairly accurate. Measured joint angles at the PIP joint of the little finger experienced more variation from the default angle compared to the DIP joint of the same finger that experienced less variation.

Table 2: Results for Varied Lateral (side-to-side) Position.

Lateral Position (cm)	Index	Middle		Little		$\bar{x}$ (deg)	AE (deg)	95% CI (deg)
	DIP (deg)	PIP (deg)	DIP (deg)	PIP (deg)	DIP (deg)			
20 to the right of the LM	73.80	77.34	70.90	68.45	74.67	73.03	16.97	[68.76, 77.30]
15 to the right of the LM	78.67	76.56	83.25	74.46	75.92	77.77	12.23	[73.53, 82.01]
10 to the right of the LM	80.71	75.46	84.56	76.98	78.54	79.25	10.75	[74.85, 83.65]
5 to the right of the LM	79.46	79.42	80.68	78.96	82.88	80.28	9.72	[78.31, 82.25]
Centre of the LM	79.41	86.77	85.78	87.12	89.25	85.67	4.33	[81.05, 90.29]
5 to the left of the LM	78.77	80.55	81.35	79.61	80.45	80.15	9.85	[78.93, 81.37]
10 to the left of the LM	77.46	76.76	82.13	79.45	76.86	78.53	11.47	[75.70, 81.37]
15 to the left of the LM	81.26	74.94	73.50	73.46	75.57	75.75	14.25	[71.76, 79.74]
20 to the left of the LM	75.06	79.26	72.98	69.79	71.93	73.80	16.20	[69.34, 78.26]

The 95% confidence intervals that correspond to the centre of the LM and that correspond to 5 cm to the left of the LM illustrate an appropriate range since the intervals are close to the default angle of 90°. The 95% confidence intervals for 5, 10, and 15 cm to the right of the LM and that for 10 cm to the left of the LM indicate somewhat good intervals though the lower limits for these intervals are not as close to the default angle of

90° as the upper limits. In contrast, confidence intervals that correspond to 20 cm to the right of the LM, 15, and 20 cm to the left of the LM exhibit an inappropriate range of values since both lower and upper limits of these intervals are not close to the default angle of 90°.

In Figure 23, 25 and 26, DIP(2) is DIP joint at the index finger, PIP(3) and DIP(3) are PIP and DIP joints at the middle finger, and PIP(5) and DIP(5) are PIP and DIP joints at the little finger. Negative distances are distances moved to the left of the LM, 0 is the position at the centre of the LM, and positive distances are distances moved to the right of the LM.

From this experiment, we see it is appropriate for users performing hand exercise using LM to avoid moving their hands more than 10 cm on either the left or the right of the LM. It is preferred for the users to perform hand exercises/gestures at the centreline of the LM or near to it.

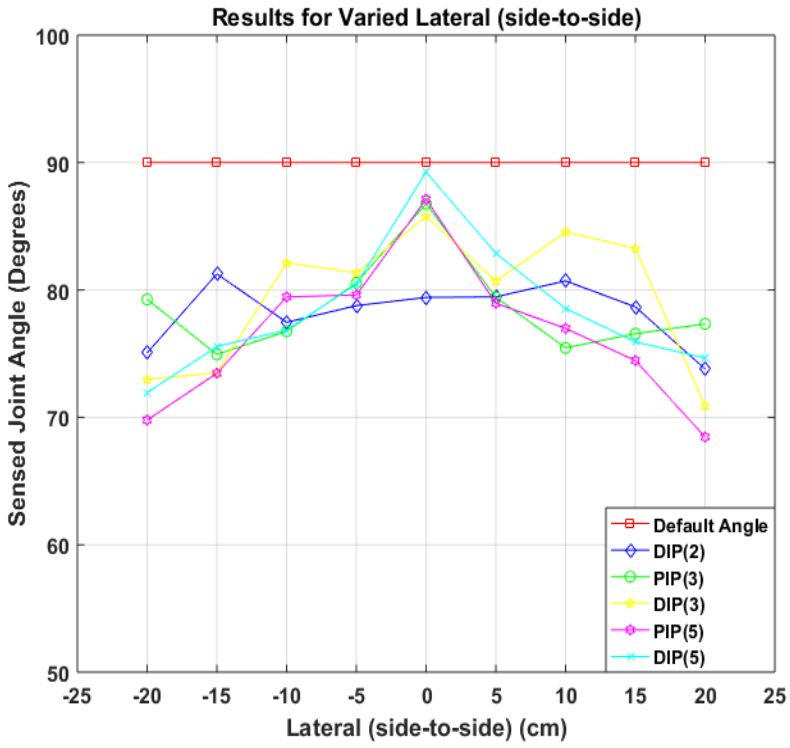

Figure 23: Sensed Joint Angle for Lateral (side-to-side).

(c) Forward-Backward Experimental Results

The artist's hand was held at 0, 2, 4, 6, 8 and 10 cm to the front and behind the LM centreline. The elevation was maintained at 10 cm from the surface of the LM to the palm of the artist's hand and it was not rotated. It remained in the horizontal position with the palm facing downwards. Negative distances are distances moved behind of the LM, 0 is the position at the centre of the LM, and positive distances are distances moved in front of the LM, closer to the user.

Table 3 and Figure 24 show results obtained for the forward-backward experiment. From the results, the lowest absolute error was obtained when the artist's hand was at the centre of the LM field of view when performing static hand gestures and this was 6^{o}. Highest absolute errors were obtained at forward-backward positions of 8 cm behind of the LM, 10 cm in front, and behind of the LM which were 17.76^{o}, 18.02^{o} and 19.24^{o}, respectively. Generally, the joint angles obtained for the forward-backward positions further from the LM registered more variation from the default angle than those that were close to the LM, as expected. This is evident from the measured joint angles obtained at 10 cm in front, and behind of the LM, 8 cm in front, and behind of the LM, 6 cm in front, and behind of the LM. Compared with all the five measured joints, the PIP joint at the middle finger registered more variation from the default angle and DIP joint of the little finger registered less variation. This could be caused by occlusion of the middle finger by neighbouring fingers.

Generally, absolute errors increase as the artist's hand is moved away from the centre of the LM. In this experiment, we have established that the best position to place the hand to perform hand gestures/exercises when using LM is along the centreline of the device or up to 4 cm from it. This is because all the measured joint angles when the artist's hand was at the centre of the LM field of view or closer are reasonably accurate.

The 95% confidence intervals that correspond to the centre of the LM, 2 and 4 cm in front of the LM, and 2 cm behind of the LM illustrate an appropriate range since the intervals are close to the default angle of 90^{o}. The 95% confidence interval that corresponds to 4 cm behind the LM indicates a somewhat good interval though the lower limit for this

interval is not as close to the default angle of 90° as the upper limit. On the other hand, the 95% confidence intervals that correspond to 6, 8, and 10 cm in front of the LM and those that correspond to 6, 8, 10 cm behind the LM reveal unsuitable ranges since both lower and upper limits of these intervals are not close to the default angle of 90°.

Table 3: Results for Forward-Backward Position.

Forward-Backward Position (cm)	Index	Middle		Little		$\bar{x}$ (deg)	AE (deg)	95% CI (deg)
	DIP (deg)	PIP (deg)	DIP (deg)	PIP (deg)	DIP (deg)			
10 in front of the LM	73.46	69.95	75.91	69.51	71.08	71.98	18.02	[68.66, 75.30]
8 in front of the LM	78.46	70.13	71.26	72.55	77.76	74.03	15.97	[69.28, 78.78]
6 in front of the LM	74.78	78.46	74.46	74.46	78.23	76.08	13.92	[73.50, 78.66]
4 in front of the LM	80.45	78.65	79.76	79.25	81.92	80.01	9.99	[78.45, 81.57]
2 in front of the LM	86.33	78.97	83.43	84.86	80.44	82.81	7.19	[79.02, 86.60]
Centre of the LM	81.63	85.44	86.25	78.94	87.69	83.99	6.00	[79.52, 88.47]
2 behind of the LM	78.45	77.78	80.25	81.70	83.56	80.35	9.65	[77.42, 83.29]
4 behind of the LM	75.57	74.46	75.61	82.35	81.76	77.95	12.05	[73.26, 82.64]
6 behind of the LM	72.87	75.24	76.87	74.67	77.94	75.52	14.48	[73.08, 77.96]
8 behind of the LM	71.91	76.05	69.23	74.46	69.56	72.24	17.76	[68.53, 75.95]
10 behind of the LM	75.49	70.98	68.91	69.53	68.90	70.76	19.24	[67.31, 74.21]

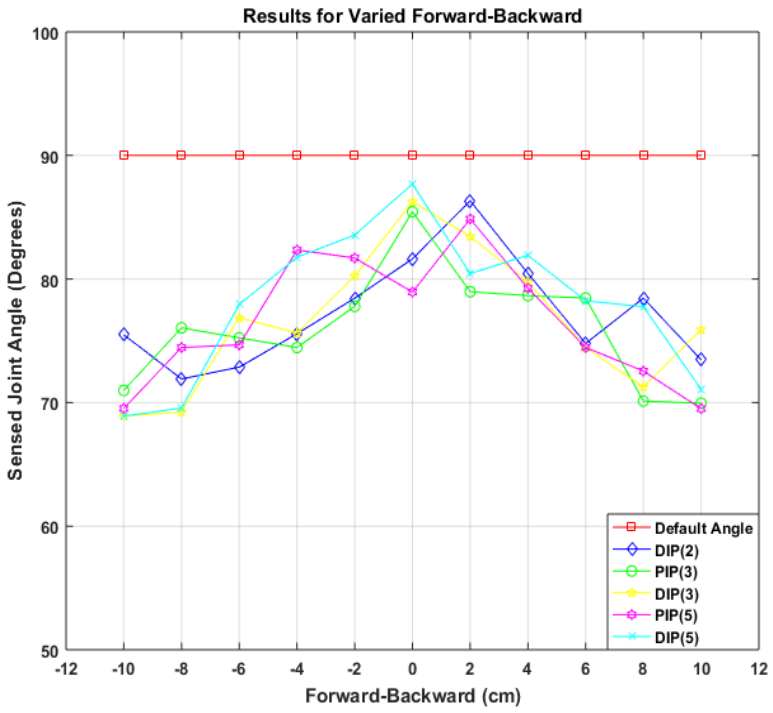

Figure 24: Sensed Joint Angle for Forward-Backward.

(d) Rotation Experimental Results

For rotation experiments, the elevation was maintained at 10 cm from the surface of the LM to the palm of the artist's hand and it was held along the centreline of the LM in all of the rotations. This meant that the artist's hand was held at 0 cm for both lateral (side-to-side) and forward-backward positions.

Table 4 and Figure 25 show results obtained for the rotation experiment. The lowest absolute error was obtained when the palm of the artist's hand was facing down when performing static hand gestures and this was 5.42°. The highest absolute errors were obtained at rotations when the artist's hand was at 45° with the thumb down, when the artist's hand was at the vertical position with the thumb down, and when the palm of the artist's was facing up and these were 12.83°, 13.08° and 18.36°, respectively. Compared with all five measured joints, the PIP joint at the middle finger registered more variation from the default angle and DIP joint of the little finger registered less variation. This could be attributed to occlusion likely to be experienced by the middle finger caused by neighbouring fingers.

Table 4: Results for Rotation Relative to the LM.

Rotation of the Hand Relative to the LM	Index	Middle		Little		$\bar{x}$ (deg)	AE (deg)	95% CI (deg)
	DIP (deg)	PIP (deg)	DIP (deg)	PIP (deg)	DIP (deg)			
Palm faces down	79.56	83.85	84.44	85.78	89.25	84.58	5.42	[80.23, 88.93]
At 45° and thumb up	84.69	77.57	85.05	83.45	87.70	83.69	6.31	[79.03, 88.35]
At Vertical position and thumb up	89.95	77.25	79.26	78.16	86.75	82.27	7.73	[75.19, 89.35]
Palm faces up	70.10	65.44	77.91	64.78	79.97	71.64	18.36	[62.94, 80.34]
At 45° and thumb down	83.46	70.66	74.26	77.46	80.01	77.17	12.83	[71.01, 83.33]
At Vertical position and thumb down	69.59	74.44	77.84	80.06	82.68	76.92	13.08	[70.60, 83.24]

The 95% confidence intervals that were obtained when palm faces down and at 45° when thumb up illustrate an appropriate range since the intervals are close to the default angle of 90°. The 95% confidence intervals obtained at the vertical position when the thumb is up, when palm faces up, at 45° when thumb is down, and at the vertical position when thumb down indicate somewhat good intervals though the lower limits for these intervals are not as close to the default angle of 90° as the upper limits.

Small alterations such as placing a hand above the LM so that it makes an angle of 45° or less do not trigger much variation from the default angle of 90°. However,

the same alterations may trigger much variation if the palm of the hand is facing up while the thumb pointed down. Therefore, users operating the LM for hand exercise purposes should avoid rotations that make their palms face upwards, and rotations that make the thumbs face downwards.

The above results confirm that the magnitude of the absolute errors varies with displacement from a "sweet spot". This implies some form of set-up protocol may be needed to ensure exercises only take place when the hand is within a limited range of this ideal location. Periodically, the LM should be able to tell if the hand is in this sweet spot or has drifted a few centimetres. If this is the case, the user must stop hand exercises and try to locate the sweet spot before starting the exercises again.

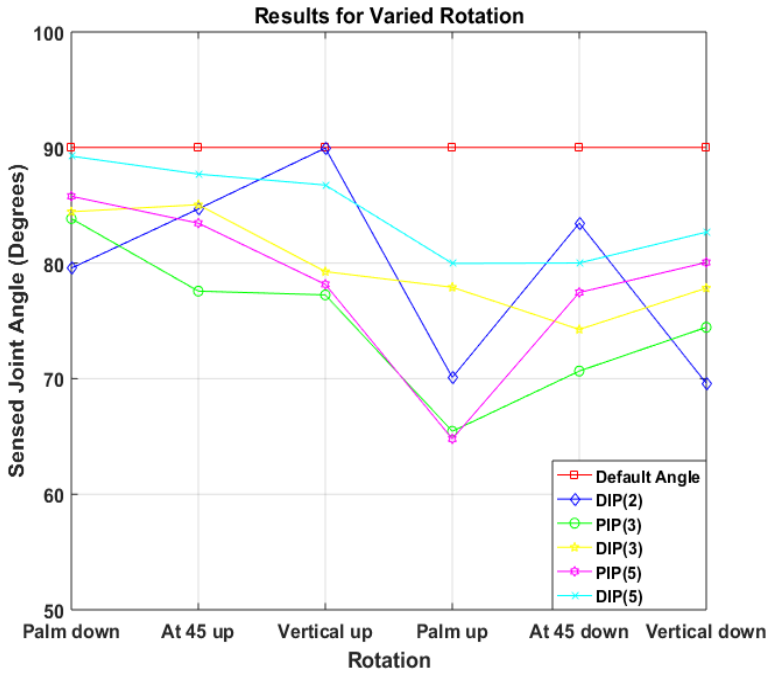

Figure 25: Sensed Joint Angle for Rotation.

4.2.3 Experimental Set-up to Reduce Absolute Errors (AEs) using PCA

Absolute Error (AE) between a measured joint angle x_i by the LM and the default joint angle d measured by a protractor has been defined in Equation (4-2), Section

4.1. It is important to note that PCA is capable of reducing absolute errors that are largely due to a hand moving away from the centreline of the LM, moving a hand outside the range of the interaction area of the LM (as described in Section 3.1.1), and unreasonable rotations of a hand relative to the LM when performing hand gestures/exercises.

We have applied PCA to the raw data to reduce absolute errors in LM measurements. Figure 26 shows the proposed PCA error reduction pipeline. Data is transferred to the PCA domain using a PCA transform. We then select PCs and transfer data back by projecting onto selected PCs that form a feature vector.

If no absolute errors (AEs) and noise were experienced, our dataset would have 0.0^{o} as variance for each of the five variables however this was not the case as each variable experienced variance. This motivated us to devise techniques that reduce variance in a dataset. In particular, we devised a filtering technique that is able to reduce variance and hence AEs in a dataset. Some of the known techniques that can be employed include moving average filter, Savitzky-Golay filter, local regression filter [143], PCA [34] etc. Compared to other filtering techniques, PCA is more efficient and easier-to-use in relation to variance reduction in a dataset. PCA computes a new set of variables, i.e. principal components (PCs), that expresses a dataset in order of higher variance attached to the first PCs. In order to reduce absolute errors (AEs) and noise, we employ PCA on our dataset. PCA reduces variance and hence AEs as illustrated in Equation (4-5). We delete the first PC that contains the highest variance and retain the remainder PCs that form the feature vector. According to our dataset, deleting the first PC while retaining the remaining PCs in the feature vector is the best means of minimising absolute errors. The steps involved in absolute error reduction are described below.

Step 1: We systematically arrange our data in a matrix form X e.g. data from one of the sets of experiments is arranged in such a way that it has 6 rows and 5 columns (see matrix X on the next page, derived from Table 1). The 6 rows correspond to elevations of 5, 10, 20, 30, 40, and 50 cm. The 5 columns correspond to the PIP joint of the middle finger, PIP and MCP joints of the ring finger, and the MCP and PIP joints of the little finger. It is important to note that the matrix X contains only five data variables, all measured in

degrees, and therefore there no need to perform normalisation of the data variables.

$$\mathbf{X} = \begin{bmatrix} 84.73 & 85.95 & 81.35 & 81.06 & 89.93 \\ 83.46 & 86.72 & 85.64 & 86.02 & 89.23 \\ 84.36 & 88.19 & 83.69 & 82.81 & 86.28 \\ 76.83 & 85.95 & 78.81 & 75.98 & 82.83 \\ 64.57 & 76.96 & 83.95 & 75.34 & 72.01 \\ 69.53 & 80.14 & 76.39 & 73.42 & 78.31 \end{bmatrix}$$

Step 2: We input X in the R statistical package using 'prcomp()' function and eigenvectors (PCs) together with their respective standard deviations are obtained as output. From the obtained standard deviations, we compute the variance that corresponds to each of the PCs.

Step 3: We select PCs and form the feature vector v. In our experiments, we delete the first PC that contributes much variation and retain the rest to form V.

Step 4: We derive the absolute error reduced data matrix Y as

$$\mathbf{Y} = \mathbf{XVV}^T \tag{4-5}$$

where V^T is the transpose of V.

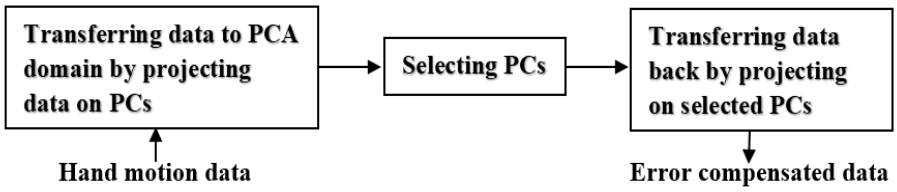

Figure 26: The proposed PCA error compensation technique pipeline.

4.2.4 Results of Absolute Error Compensation using PCA

We present results after applying PCA on all the four parameters. All the results show that the measured joint angles are fairly close to the default joint angle after applying PCA.

(a) Elevation Experimental Results

Figure 27 (b) shows the results after applying PCA on the elevation experimental results. There is a considerable improvement after applying PCA, i.e. the highest absolute error is reduced by 37.5%. Most of the joint angles are relatively closer to the default angle (90^o). This shows that PCA is an effective and efficient technique to compensate for the absolute errors arising from inadvertent hand misalignment from the central position of the LM.

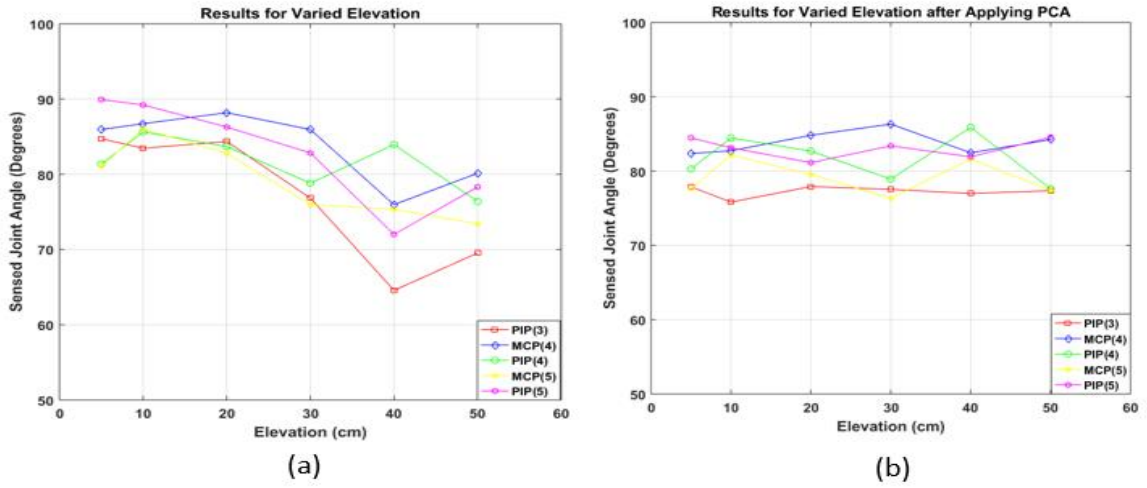

Figure 27: Varied Elevation. (a) Before Applying PCA. (b) After Applying PCA.

(b) Lateral (side-to-side) Experimental Results

Figure 28 (b) shows the results after applying PCA on lateral (side-to-side) experimental results. There is a significant improvement after applying PCA, i.e. the highest absolute error is reduced by 28.3%. Almost all the joint angles are closer to the default angle (90^o) than before applying PCA.

(c) Forward-Backward Experimental Results

Figure 29 (b) shows the results after applying PCA on forward-backward experimental results. We notice a considerable improvement after applying PCA, i.e. the highest absolute error is reduced by 33.0%. More joint angles are closer to the default angle (90^o) than before applying PCA.

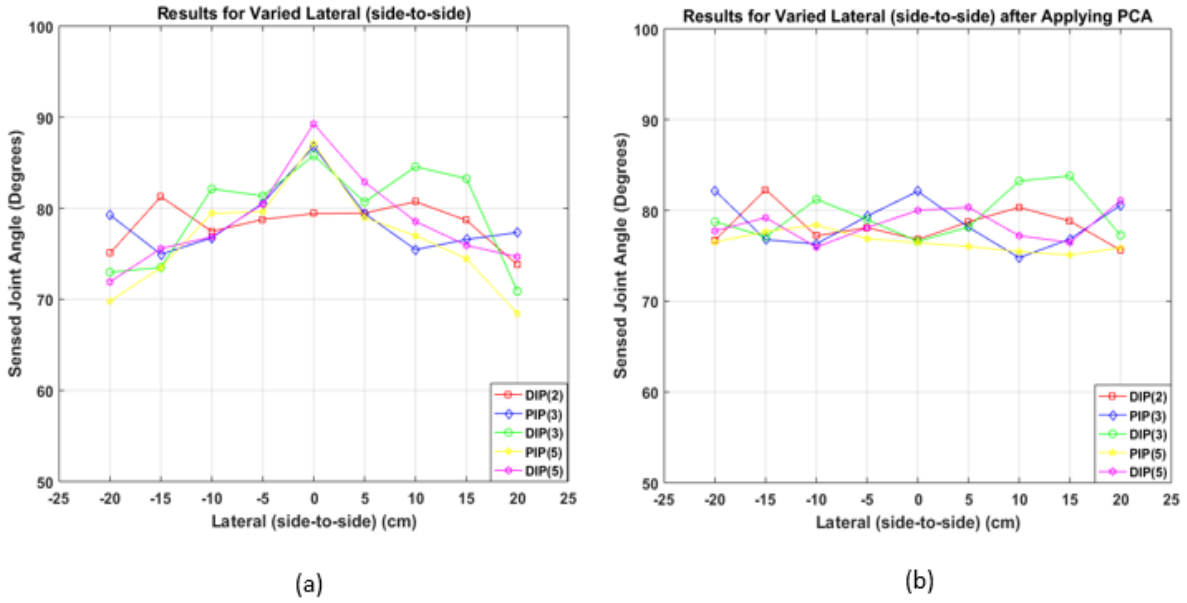

Figure 28: Varied Lateral (side-to-side). (a) Before Applying PCA (b) After Applying PCA.

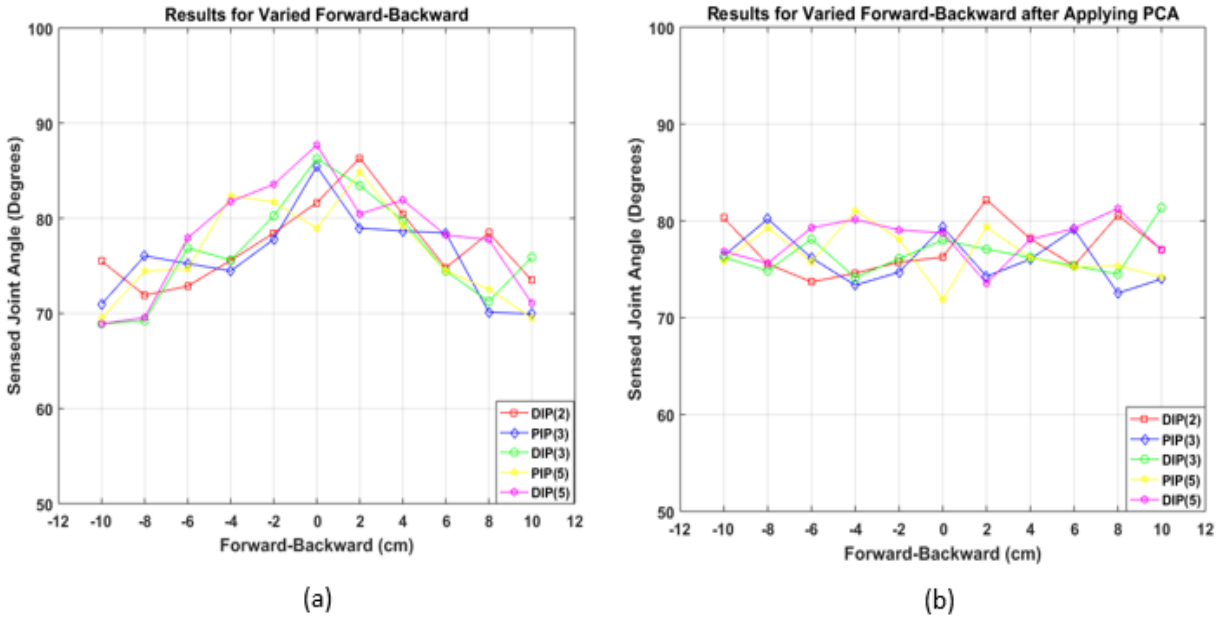

Figure 29: Varied Forward-Backward. (a) Before Applying PCA. (b) After Applying PCA.

(d) Rotation Experimental Results

Figure 30 (b) shows the results after applying PCA on rotation experimental results. There is a noticeable improvement after applying PCA, i.e. the highest absolute error is reduced by 22.4%. Most of the joint angles are above 75° which clearly shows a reasonable improvement than before applying PCA.

PCA has been applied to reduce joint angle absolute errors. It can be observed that it is not perfect. One would have expected most measured joint angles to have been close to 90°. In some experiments like the elevation experiment, PCA performed reasonably well. For lateral (side-to-side) and forward-backward, PCA performed fairly, and poorly on rotation experimental results.

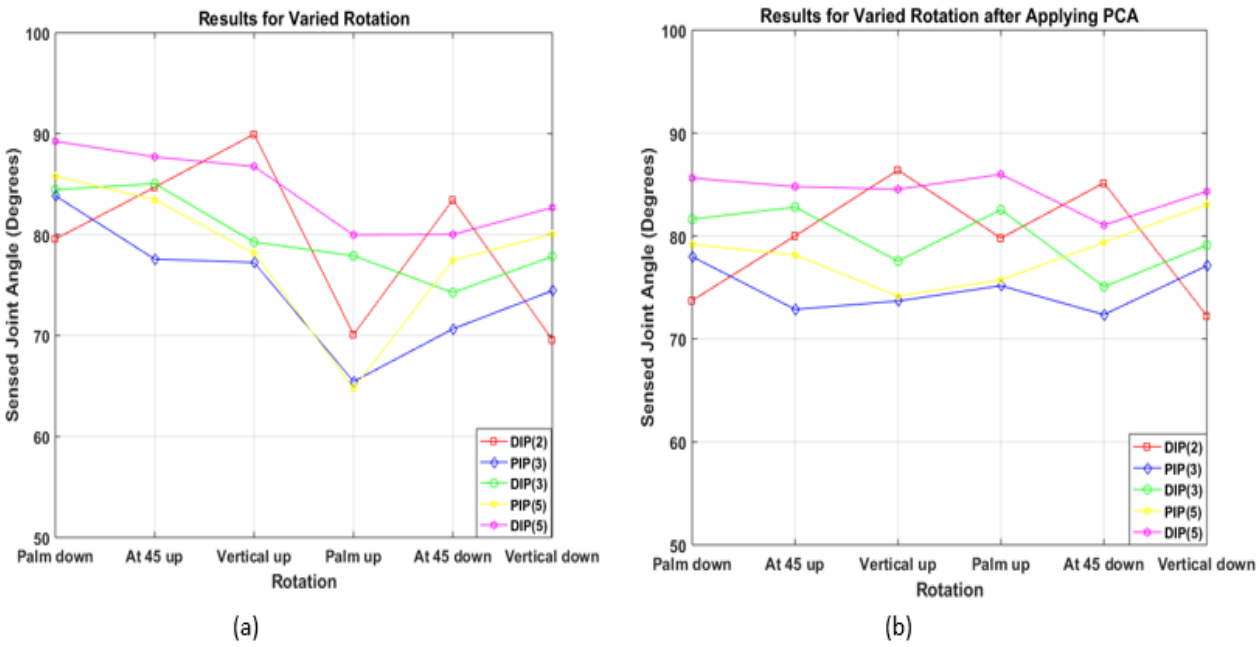

Figure 28: Varied Rotation. (a) Before Applying PCA (b) After Applying PCA.

4.3 Variation of Errors with Finger Joint Angles

We performed a set of experiments to study variation of errors with the size of the measured finger joint angles. For these experiments, the elevation was maintained at 10 cm from the surface of the LM to the palm of the artist's hand and it was held along the

centreline of the LM. This meant that the artist's hand was held at 0 cm for both lateral (side-to-side) and forward-backward positions. The artist's hand was not rotated. It remained in the horizontal position with the palm facing downwards.

4.3.1 Experimental Set-up to Investigate Variation of Errors with Finger Joint Angles

In this experimental set-up and the results Section 4.3.2, we define error as

$$Error = |d - x_i| \quad (4\text{-}6)$$

where d is the default joint angle measured by a protractor, x_i is the measured joint angle by the LM. The default varied angles are 0°, 10°, 30°, 50°, 70°, and 90°.

4.3.2 Results of Variation of Errors with Finger Joint Angles

We measured the four finger joints i.e. PIP(2), DIP(2), PIP(5), and DIP(5) of each default joint angle 6 times using the LM device. The recorded values in Figure 31 are the averages for these 6 measurements. The default joint angles are 0°, 10°, 30°, 50°, 70°, and 90°. PIP(2) and DIP(2) are PIP and DIP joints at the index finger, PIP(5) and DIP(5) are PIP and DIP joints at the little finger. These 6 times were performed on different days and on different times of a particular day, i.e. day and night. Figure 31 shows the results. The elevation was maintained at 10 cm from the surface of the LM to the palm of the artist's hand and it was held along the centreline of the LM. This meant that the artist's hand was held at 0 cm for both lateral (side-to-side) and forward-backward positions. The artist's hand was not rotated. It remained in the horizontal position with the palm facing downwards.

From Figure 31, the lowest error was registered when the default angle was 0° and this was 0°. Compared with all the default angles, the highest error was registered when the default angle was 50° and this was 4.7°. The errors are much less compared to other results, e.g. experimental results in Section 4.2.2 simply because we maintained suitable conditions for all the four parameters of elevation, lateral (side-to-side), forward-backward, and rotation. Generally, the default angles of 0°, 10°, and 70° registered smaller errors. DIP and PIP joints of the index and little fingers registered greater errors compared to PIP and DIP joints of the index and little fingers.

We can conclude that measured joint angle errors are independent of the size of the joint angle though the LM was able to register 0° of error when the angle was set to 0°. Regardless of the size of the angle, errors will be encountered in joint angle measurement.

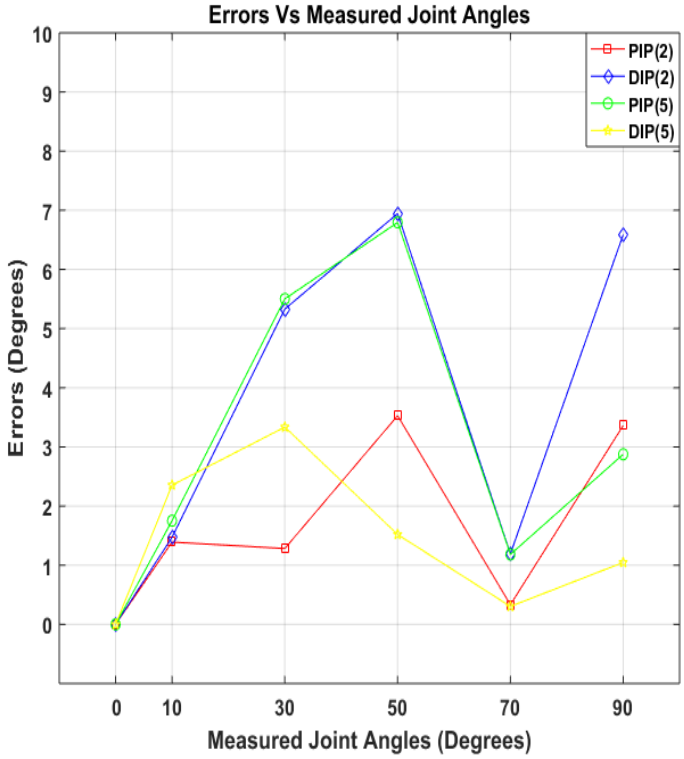

Figure 29: Errors in Measured Joint Angles.

4.4 Determining whether Absolute Errors (AEs) are Consistent in Leap Motion Devices (LMs)

We carry out this study to determine whether absolute errors (AEs) are consistent where we have used two LMs i.e. first Leap Motion (LM1) and second Leap Motion (LM2). In the following Section 4.4.1, we describe the experimental setup and the results are discussed in Section 4.4.2.

4.4.1 Experimental Set-up for Determining whether AEs are Consistent in LMs

As for the case of characterising AEs in finger joint angle measurements, all experimental set-ups are similar and we use four parameters i.e. elevation, lateral (side-to-

side), forward-backward and rotation movements of the hand relative to the LM. We use one LM at a time to test errors for the four parameters and then use the second LM and repeat the experiments.

4.4.2 Results for Determining whether Absolute Errors are Consistent in LMs

We performed each experiment 6 times (trials) for each given parameter (the parameters are elevation, lateral (side-to-side), forward-backward, and rotation as explained in Section 4.2.1) for the two LMs and obtained the average which was recorded. The recorded values in the second, third, fourth, and fifth columns of Tables 5, 6, 7, and 8 are the averages of these 6 trials. Experiments for elevations of 5, 10, 20, 30, 40, and 50 cm are provided in the first column of Table 5. Experiments for lateral (side-to-side), forward-backward, and rotation are recorded in the first columns of Tables 6, 7, and 8, respectively. These 6 trials were performed on different days and at different times of a particular day, i.e. day and night.

(a) Elevation Experimental Results

In all the elevation experiments, the artist's hand remained aligned with the centre of the LM i.e. at 0 cm for both lateral (side-to-side) and forward-backward positions and was not rotated, it remained in the horizontal position with the palm facing downwards.

In all the four parameters, we set 90^o at the DIP joints of the middle and little fingers. We chose 90^o for convenience, but this could be any angle between 0^o and 90^o. Table 5 and Figure 32 show the results obtained for LM1 and LM2. $\bar{x}_1$ and $\bar{x}_2$ are averages, AE_1 and AE_2 are absolute errors that correspond to LM1 and LM2 respectively. LM1-DIP(3) represents the DIP joint measured at middle finger using LM1 and LM2-DIP(5) represents the DIP joint measured at little finger using LM2.

AE_1 and AE_2 at same elevations generally have little variation, i.e. they are almost the same. As in the case of characterising errors in Section 4.2.2, higher absolute errors were registered at higher elevations of 30, 40, and 50 cm and lower absolute errors were registered at lower elevations of 5, 10, and 20 cm. From Figure 32, measured joint angles at the DIP joint of the little finger registered somewhat similar readings for LM1 and LM2.

Absolute errors are more when the artist's hand is moved at higher elevations.

From Figure 32, the two LMs behave similarly. For both LMs, little variation is registered for lower elevation and much variation is registered when static gestures are performed at higher elevations, e.g. at 40 and 50 cm. There is no clear distinction between the performance of the two LMs in regard to elevation experiments.

Table 5: Results for Varied Elevation.

Elevation (cm)	LM1		LM2		$\bar{x}_1$ (deg)	$\bar{x}_2$ (deg)	AE_1 (deg)	AE_2 (deg)
	Middle DIP (deg)	Little DIP (deg)	Middle DIP (deg)	Little DIP (deg)				
5	83.43	85.33	80.93	87.34	84.38	84.14	5.62	5.86
10	84.46	85.94	87.01	89.15	85.20	88.08	4.80	1.92
20	83.11	89.93	85.55	87.19	86.52	86.37	3.48	3.63
30	78.15	79.00	81.12	79.20	78.57	80.15	11.43	9.85
40	74.19	76.14	75.52	77.19	75.17	76.36	14.83	13.64
50	73.14	75.34	74.00	74.52	74.24	74.26	15.76	15.74

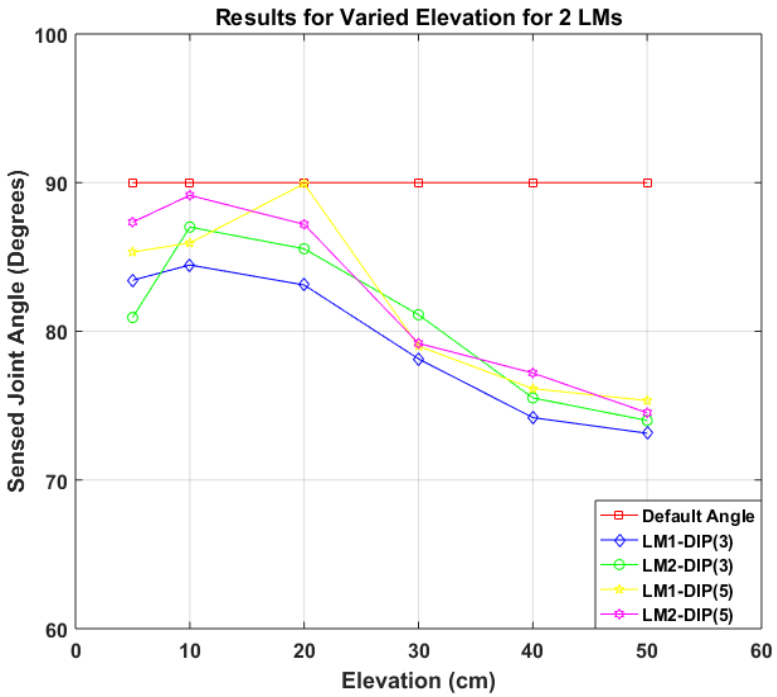

Figure 30: Varied Elevation for 2 LMs.

(b) Lateral (side-to-side) Experimental Results

The artist's hand was held at 0, 5, 10, 15 and 20 cm to the right and left of the

LM centreline. The elevation was maintained at 10 cm from the surface of the LM to the palm of the artist's hand and it was not rotated. It remained in the horizontal position with the palm facing downwards. Table 6 and Figure 33 show the results obtained.

AE_1 and AE_2 at the same lateral (side-to-side) positions are almost the same. For example, at 5 cm to the right of the LM, AE_1 is 8.25° and AE_2 is 8.43°. This also applies to other lateral positions in this experiment. As in the case of characterising absolute errors in Section 4.2.2, smaller absolute errors are registered when the artist's hand performs gestures near the centreline of the LM and greater absolute errors are registered when static gestures are performed far away from the centreline for example at 20 cm to the right and left of the LM. Figure 33 confirms that the performance of LM1 and LM2 are almost identical.

Table 6: Results for Varied Lateral (side-to-side).

Lateral Position (cm)	LM1		LM2		$\bar{x}_1$ (deg)	$\bar{x}_2$ (deg)	AE_1 (deg)	AE_2 (deg)
	Middle DIP (deg)	Little DIP (deg)	Middle DIP (deg)	Little DIP (deg)				
20 to the right of the LM	74.81	77.01	74.68	76.11	75.91	75.40	14.09	14.60
15 to the right of the LM	77.15	77.90	80.15	79.66	77.53	79.90	12.47	10.10
10 to the right of the LM	80.34	79.45	79.00	80.45	79.89	79.73	10.11	10.27
5 to the right of the LM	81.45	82.04	83.05	80.09	81.75	81.57	8.25	8.43
Centre of the LM	85.04	84.24	86.41	84.01	84.64	85.21	5.36	4.79
5 to the left of the LM	83.00	84.01	84.99	82.45	83.51	83.72	6.50	6.28
10 to the left of the LM	79.41	81.41	80.45	83.41	80.41	81.93	9.59	8.07
15 to the left of the LM	81.41	79.00	78.41	79.44	80.21	78.93	9.79	11.07
20 to the left of the LM	75.11	76.42	77.02	76.82	75.77	76.92	14.23	13.08

(c) Forward-Backward Experimental Results

The artist's hand was held at 0, 2, 4, 6, 8 and 10 cm to the front and behind the LM centreline. The elevation was maintained at 10 cm from the surface of the LM to the palm of the artist's hand and it was not rotated. It remained in the horizontal position with the palm facing downwards. Table 7 and Figure 34 show the results obtained.

AE_1 and AE_2 are almost identical for the same forward-backward position. For example, at 2 cm in front of the LM, AE_1 is 3.74° and AE_2 is 3.44°. Much variation from the default angle (90°) is experienced when static hand gestures are performed relatively far away from the centreline of the LMs. Less variation is registered when static hand gestures are performed at the centre or near the centreline of the LMs. From Figure 30, LM1 and

LM2 perform relatively the same, i.e. the difference in performance is negligible.

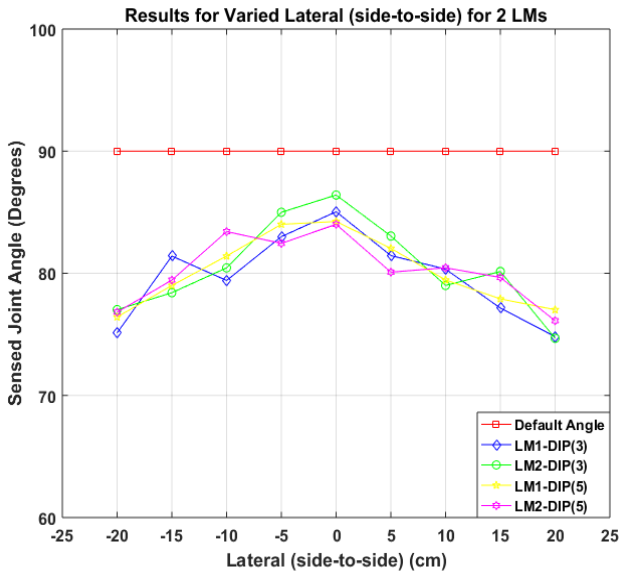

Figure 31: Varied Lateral (side-to-side) for 2 LMs.

Table 7: Results for Forward-Backward.

Forward-Backward Position (cm)	LM1		LM2		$\bar{x}_1$ (deg)	$\bar{x}_2$ (deg)	AE_1 (deg)	AE_2 (deg)
	Middle DIP (deg)	Little DIP (deg)	Middle DIP (deg)	Little DIP (deg)				
10 in front of the LM	74.11	69.19	75.24	71.24	71.65	73.24	18.35	16.76
8 in front of the LM	77.41	72.41	75.01	71.08	74.91	73.04	15.09	16.96
6 in front of the LM	77.50	75.04	74.35	78.42	76.27	76.39	13.73	13.61
4 in front of the LM	81.44	79.61	80.41	80.00	80.53	80.21	9.47	9.79
2 in front of the LM	87.41	85.11	83.12	90.00	86.26	86.56	3.74	3.44
Centre of the LM	86.44	85.40	86.00	84.09	85.92	85.05	4.08	4.95
2 behind of the LM	79.44	82.09	83.11	80.09	80.77	81.60	9.23	8.40
4 behind of the LM	78.00	84.11	82.00	82.95	81.06	82.48	8.94	7.52
6 behind of the LM	79.44	81.01	78.46	83.00	80.22	80.73	9.78	9.27
8 behind of the LM	77.00	76.45	79.11	75.10	76.72	77.11	13.28	12.89
10 behind of the LM	75.41	74.15	73.49	74.13	74.78	74.46	15.22	15.54

(d) Rotation Experimental Results

For rotational experiments, the elevation was maintained at 10 cm from the surface of the LM to the palm of the artist's hand and it was held along the centreline of the LM in all the rotations. This meant that the artist's hand was held at 0 cm for both lateral (side-to-side) and forward-backward positions. Table 8 and Figure 35 show the results obtained.

Generally, AE_1 and AE_2 are almost identical for the same rotation, e.g. when static gestures are performed with the palm of the artist's hand facing down, the AE_1 is 6.87° and AE_2 is 5.60°. As in the case of characterising errors in Section 4.2.2, larger errors are registered when the palm faces up. This can be attributed to the LM being unable to detect the direction vectors of the finger bones as well. From Figure 35, there is no clear distinction in the performance of LM1 and LM2.

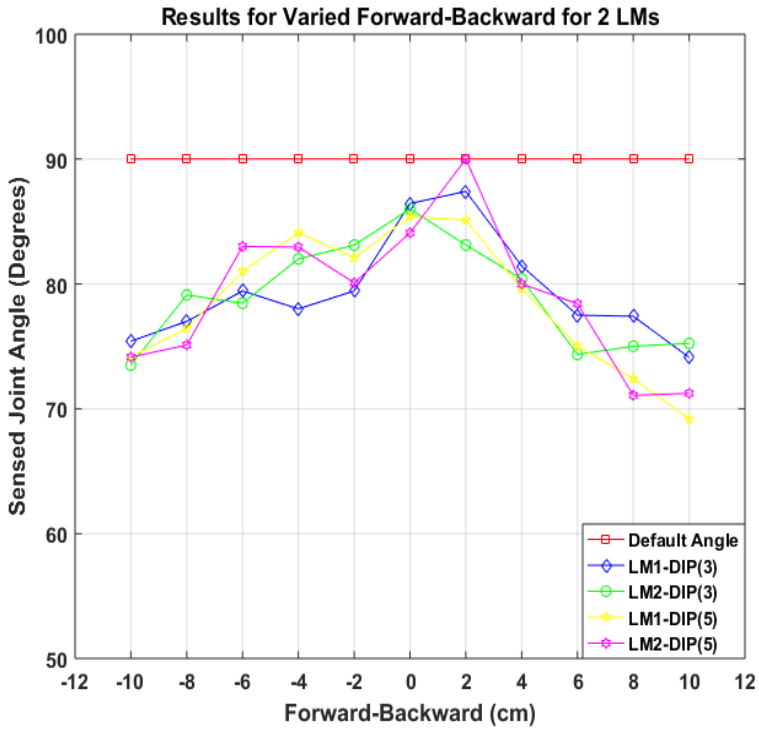

Figure 32: Varied Forward-Backward for 2 LMs.

Table 8: Results for Varied Rotation.

Rotation of the Hand Relative to the LM	LM1		LM2		$\bar{x}_1$ (deg)	$\bar{x}_2$ (deg)	AE_1 (deg)	AE_2 (deg)
	Middle DIP (deg)	Little DIP (deg)	Middle DIP (deg)	Little DIP (deg)				
Palm faces down	81.41	84.85	85.14	83.65	83.13	84.40	6.87	5.60
At 45° and thumb up	83.05	80.15	83.45	82.68	81.60	83.06	8.40	6.94
At Vertical position and thumb up	79.45	81.09	80.15	80.61	80.27	80.38	9.73	9.62
Palm faces up	74.09	75.18	76.64	75.74	74.64	76.19	15.36	13.81
At 45° and thumb down	83.77	80.80	82.14	81.98	82.28	82.06	7.72	7.94
At Vertical position and thumb down	78.45	79.68	80.66	79.86	79.06	80.26	10.94	9.74

In this experiment, our aim was to determine whether absolute errors are consistent across LMs. We have used two LMs to ascertain that worse absolute errors are experienced by movements away from the centreline of the LM. This was actually the case for characterising absolute errors in Section 4.2.2 but we wanted to confirm this by using a second LM. We can therefore conclude that whichever LM you use, substantial absolute errors are likely to be encountered when hand gestures are performed away from the centreline of the LM.

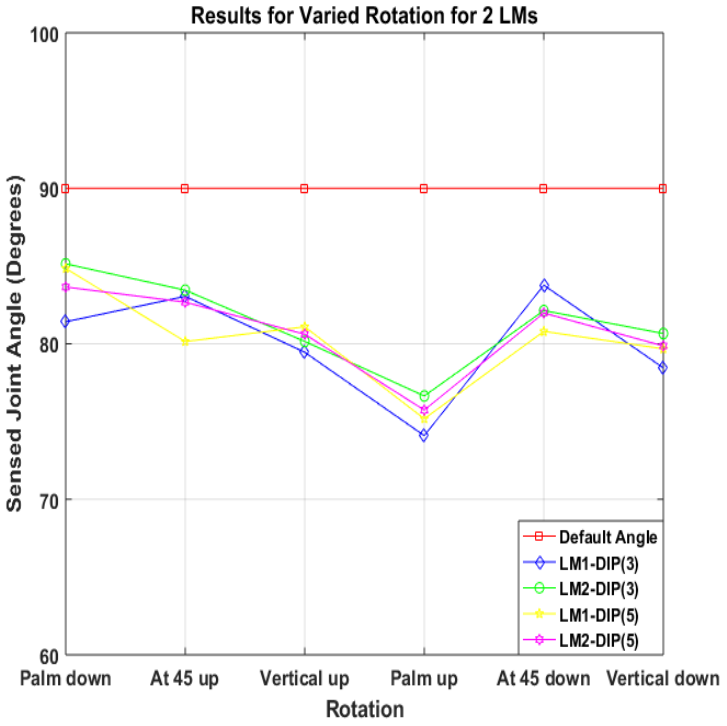

Figure 33: Varied Rotation for 2 LMs.

4.5 Summary

We have described the motion data collection procedure where we illustrated how a joint angle is calculated from the frames of the data that are in form of direction vectors of the finger bones. We have also defined joint angle θ, absolute error and 95% CI. We have illustrated our designed experiments to characterise finger joint angle absolute errors based on four parameters i.e. elevation, lateral (side-to-side), forward-backward, and

rotation movements of the hand relative to the LM. We have also described the experimental set-up to study variation of joint angles with absolute errors, to determine whether absolute errors are consistent with LMs, and to reduce absolute errors using PCA.

We have presented our results and discussion of characterising finger joint angle absolute errors. Absolute errors were characterised based on four parameters i.e. elevation movements, lateral (side-to-side) movements, forward-backward movements and rotation. We have discussed that absolute errors are mainly experienced when these movements are away from the centreline of the LM.

We also applied PCA on the raw data in order to reduce absolute errors. Absolute errors were reduced by 37.5%, 28.3%, 33.0%, and 22.4% for the experimental results of elevation, lateral (side-to-side), forward-backward, and rotation, respectively. In addition to restraining users from movements away and far above the centreline of the LM, PCA can be applied to improve the accuracy of measured joint angles.

We have also presented results on the variation of errors with joint angles where we observed that lowest errors were registered when the default angle was 0^{o} and highest errors were registered when default angle was 50^{o}. Furthermore, we discovered that whichever size of joint angle you measure, errors will be experienced. We also presented results to determine whether absolute errors are consistent using two LMs.

During experimentation, systematic errors that are mainly associated with a fault in a measuring device were minimised by performing an initial simple calibration procedure when the LM was being used for the first time. If the simple calibration procedure is not performed, it may affect LM measurements since the sensors in the LM may have been knocked out of their initial alignment. Absolute errors are largely due to moving a hand away from the centreline and outside the range of the interaction area of the LM. Unreasonable rotations of a hand relative to the LM when performing hand gestures/exercises also exacerbate absolute errors.

5 Hand Gesture Recognition in LM Using LDA and SVM

In Chapter 4 we implemented PCA to reduce absolute errors in LM measurements, however these kinds of errors can also be reduced by implementing machine learning techniques such as LDA and SVM. In this chapter, we implement LDA and SVM in order to recognise and classify hand gestures. In addition, LDA and SVM can learn about measurement errors in the LM and hence compensate for them. LDA performs relatively well compared to other models such as a logistic regression model when the classes are separated relatively well [35]. SVM has been shown to perform well in various settings and is normally known to be a superior classifier [35]. Machine learning techniques are able to learn parameters that describe the hand gestures [4]. After training the LDA and SVM models, the models can accurately recognise and classify the hand gestures in given test samples and therefore measurement errors experienced during dataset creation are reduced.

5.1 Methodology

We use a markerless LM sensor that can be easily acquired by a user at low cost. The LM can detect palm and finger movements. The tracking data, in the form of frames, can be accessed using its Software Development Kit (SDK).

5.1.1 Architecture of the Proposed System

In our proposed framework, an input signal is acquired by use of the LM. The customised application that operates in conjunction with LM software on a computing device produces frames of data when it is run. After obtaining these frames of data, extraction of direction vectors of all the finger bones is performed. We then obtain joint angles for all the finger joints, and this forms the feature vector.

Once we obtain feature vectors for our dataset, we partition the dataset into training and test samples. We then apply LDA and SVM on the training samples using RStudio. After training, we then apply our trained model on test samples to observe to what extent the model accurately classifies the gestures. Figure 36 illustrates the architecture of our proposed framework.

5.1.2 Signal Acquisition

To obtain meaningful recognition and classification of the performed hand gestures, accurate knowledge concerning hand location and orientation is a prerequisite [144-148]. LM is used to record finger joint angles for each of the performed gestures. In order to obtain frames of data in form of direction vectors of the bones of the fingers, we run the Java customised application with the LM SDK application simultaneously. We then compute joint angles using direction vectors of the bones of the specified fingers. Details are explained in Section 4.1, Chapter 4.

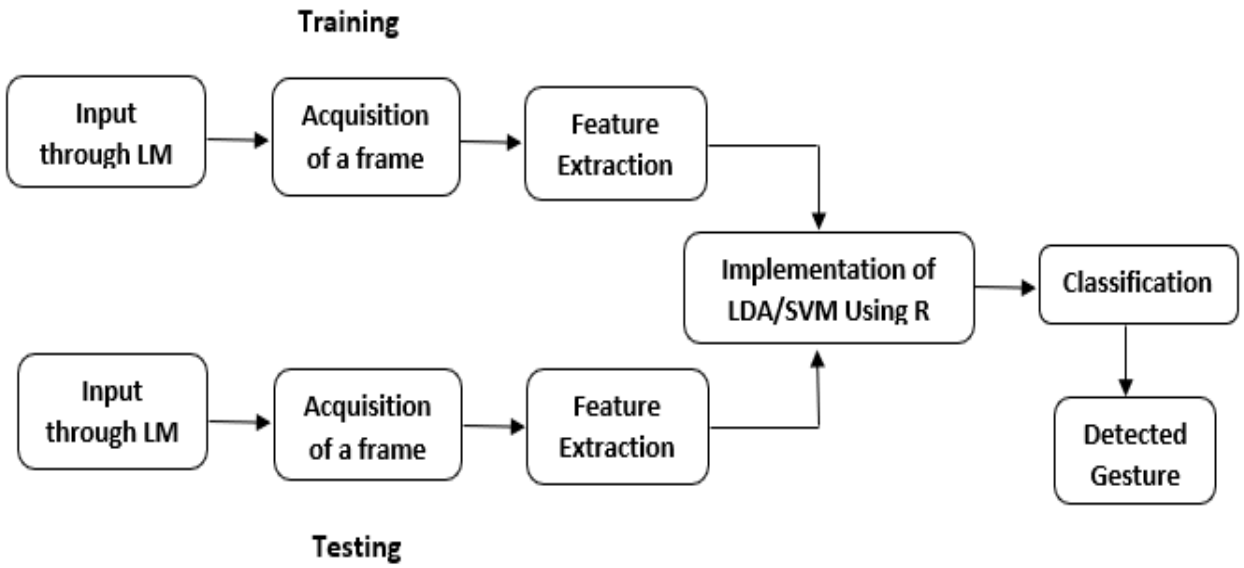

Figure 34: Architecture of the proposed system [53].

5.1.3 Description of the Feature Vector

The feature vector consists of all fourteen finger joint angles i.e. $P = 14$ as described in Table 9. Other parameters such as the angles between the fingers and palm position can be considered as features. However, our experimental trials suggest that including such features does not have a significant impact on classification accuracy. Because of this, we limit the feature vector to a collection of 14 measured features, all joint angles.

5.1.4 Gestures Under Consideration

We have provisionally limited the number of gestures to 4 as depicted in Figure 37. In this figure, (a) is g_1 when the hand is open and the palm facing down, (b) is g_2 when the fist is closed, (c) is g_3 when the middle, ring, and little finger are closed, and (d) is g_4 when the thumb and index finger are closed. This set of gestures can be updated as necessary.

Table 9: Feature Vector.

Finger	MCP Joint	PIP Joint	DIP/IP Joint
Thumb	y_1		y_2
Index	y_3	y_4	y_5
Middle	y_6	y_7	y_8
Ring	y_9	y_{10}	y_{11}
Little	y_{12}	y_{13}	y_{14}

Figure 35: A set of gestures to facilitate gesture recognition using LDA and SVM.

5.2 Results and Discussion

The machine learning techniques described in Sections 3.3.1 and 3.4.1 have been tested with a dataset of 600 samples. We begin this section with a description of how we collected our dataset. Later, results of experiments are presented and analysed.

5.2.1 Dataset for Experiments

We created a dataset using an artist's hand fixed above the LM. This is because artist's hand gives more accurate measurements than a human hand since it is good at maintaining a required posture for as long as necessary.

We collected our dataset taking into consideration all four parameters as described in Section 4.2.1. These parameters were elevation, lateral (side-ways) movements, forward-backward positioning, and rotation; all relative to the LM. For elevation, the artist's hand was moved roughly between 5 to 30 cm. For lateral positioning, it was moved approximately between 0 to 15 cm to the right and left of the LM middle position. For forward-backward positioning, it was moved roughly between 0 to 10 cm to the front and behind of the LM middle position. For rotation, the hand was in some cases not rotated, ensuring that the palm was facing down while performing various gestures. In other cases, it was rotated at slightly less or equal to 45^{o}. The specified movements relative to the LM are in the field of view (interaction area) of the LM and this reduces absolute errors (AEs) as described Section 4.2.2. In addition, sensible rotations of a hand relative to the LM reduce absolute errors also as discussed in Section in 4.2.2.

We collected 150 samples from each gesture, g_1, g_2, g_3, and g_4. Therefore, a total of 600 samples were collected. We used 62.5% of 600 samples in training and the remaining 37.5% in testing. Usually when implementing machine learning techniques, data scientists divide any dataset into a bigger proportion as a training dataset and a smaller one as a test dataset in order to avoid overfitting and to ensure the trained models generalise well on new observations (data patterns) that are not seen in a training dataset [34].

We employed two popular machine learning techniques, i.e. LDA and SVM, so that our framework can learn about the errors in order to compensate for them. In our previous work [34], we discovered that hand displacements away from the middle position and significantly above the screen surface of the LM give rise to serious errors hence resulting in inaccurate measures of finger joint angles. We therefore maintain the hand close to the centreline of the LM.

5.2.2 Results of Application of LDA to Recognise Hand Gestures

We present and analyse results obtained after applying LDA on our dataset. Table 10 depicts the confusion matrix obtained for the LDA model on a test data samples. The confusion matrix is a table layout that illustrates the performance of the two applied machine learning techniques i.e. LDA and SVM on a test dataset. Our confusion matrices are demonstrated in Tables 10, 11, 12 and 13. For example in Table 10, the total number of gestures in category g_1 is 58 (53+0+2+3) obtained after adding the four elements in the first row of the table. The same addition operations are applied to obtain the total number of gestures in category g_2, g_3 and g_4 in the second, third and fourth rows, respectively. Out of 58 gestures in category g_1 in a test dataset, 53 were classified correctly and assigned g_1 by the LDA model. It misclassified 2 gestures and assigned them to gesture g_3 and misclassified 3 gestures and assigned them to category g_4. Out of 59 gestures that are in category g_2, 55 are classified correctly and assigned g_2 by the LDA model. One gesture is misclassified and assigned to category g_1, and 3 gestures are further misclassified and assigned to category g_3. Out of 51 gestures that belong to category g_3, 45 are classified correctly and assigned to g_3 by the model. Six gestures are misclassified and assigned to category g_2. Out of 57 gestures that belong to category g_4, 46 gestures are classified correctly, and 11 gestures are misclassified and assigned to category g_1. The model recognised 11 gestures of class g_4 as those that belonged to class of g_1. This could be attributed to noise related issues during data collection.

To compute the accuracy of each category of the four gestures, the number of gestures classified correctly in a category is divided by the total number of gestures in the same category. From Table 10, accuracy of g_1 is computed as 53/58=91.4%, the accuracy of g_2 is 55/59=93.2%, the accuracy of g_3 is 45/51=88.2%, and the accuracy of g_4=46/57=80.7%. The overall accuracy of the LDA model is computed as a percentage of gestures classified correctly in a test sample divided by the total number of gestures in the test sample. From Table 10, the total number of gestures in the test sample is 58+59+51+57=225. Gestures that are classified correctly are 53+55+45+46=199. Therefore, the overall accuracy of the LDA model is 199/225=88.4%. A similar computational procedure is applied in Tables 11, 12

and 13 to obtain accuracy of SVM models.

Figure 38 shows how the LDA model performed on each of the 4 gestures. The model performed better on classifying g_2 with an accuracy of 93.2% while it performed poorly on g4 with an accuracy of 80.7%. This can be attributed to noise and measurement errors in the LM.

Table 10: Confusion Matrix for LDA Model.

		Predicted			
		g_1	g_2	g_3	g_4
Actual	g_1	53	0	2	3
	g_2	1	55	3	0
	g_3	0	6	45	0
	g_4	11	0	0	46

5.2.3 Results of Application of SVM to Recognise Hand Gestures

We present and analyse results obtained after applying SVM models on our dataset. We varied the SVM models using kernels. In Section 3.4.1, we described that SVM enlarges the feature space efficiently using kernels. These kernels are linear, polynomial, radial, and sigmoid. We shall use K when referring to kernel in our experimental results. We first present confusion matrices for all our SVM models varied with four kernels in Tables 11-13. We note that the confusion matrix tables for SVM radial and sigmoid are exactly the same; hence we present one table for both. All the four gestures are predicted in the same way by the SVM radial and the SVM sigmoid model.

For all the confusion matrix Tables 11-13, the four SVM models almost agree on the prediction of g_2 and g_3. Only 6 gestures that were labelled as g_3 were classified as g_2 and only 3 gestures that belonged to g_2 were recognised as g_3. Furthermore, for the SVM linear and SVM polynomial models, 9 gestures belonging to g_4 were recognised as g_1 and for SVM radial and SVM sigmoid models, 11 gestures that belong to g_4 were classified as g_1. This could be attributed to the fact that g_1 and g_4 have nine identical variables out of fourteen in

their respective feature vectors and this applies to g_2 and g_3 as well.

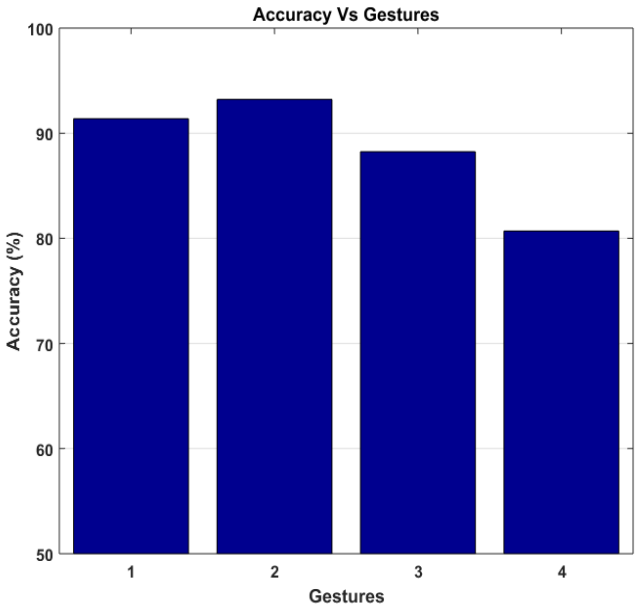

Figure 36: Accuracy of each gesture by LDA model.

Table 11: Confusion Matrix for SVM, K = Linear.

		Predicted			
		g_1	g_2	g_3	g_4
Actual	g_1	54	1	0	3
	g_2	1	55	3	0
	g_3	0	6	45	0
	g_4	9	0	0	48

From Figure 39, linear and polynomial kernels perform relatively better especially in recognising gesture g_4 than the radial and sigmoid kernels on our dataset. All 4 kernels recognise and classify gestures 1 and 2 more accurately than gestures 3 and 4. This could be because, when performing gesture 3 and 4, finger occlusion is likely to be experienced. This further contributes to measurement errors in the LM.

We finally compare the accuracy of the two machine learning algorithms with a baseline. The baseline is the accuracy obtained when we directly observe the test samples and assign all the samples to the gestures they supposedly represent. We use a threshold of joint angles for straightening and bending fingers. For example, for all straightened fingers, all joint angles were less or equal to 30^o and for all bent fingers, the joint angles were greater or equal 60^o. For example, a test sample with all the fourteen joint angles that are equal or less than 30^o is identified as g_1 by the baseline. A test sample with the joint angles for the middle, ring, and little fingers that are equal to or greater than 60^o and the joint angles for the thumb and index that are equal to or less than 30^o is identified as g_3 by the baseline. This is because all the fingers are straightened for g_1. For g_3, the thumb and index are straightened whereas the middle, ring, and little fingers are bent (see Section 5.1.4). This similarly applies to g_2 and g_4. Out of 225 test samples, 151 satisfied the above baseline criteria and this resulted in a baseline performance of 67.1%. It is challenging for the human eye to obtain a greater accuracy since the four different gestures possess some similar finger joint angles in their respective feature vectors. For example, g_1 and g_4 have nine similar joint angles for little, ring and middle finger that stay straight when both gestures are performed. The same applies for g_2 and g_3 where the two gestures have nine similar joint angles for the little, ring and middle finger that are bent when both gestures are performed.

Table 12: Confusion Matrix for SVM, K = Polynomial.

		Predicted			
		g_1	**g_2**	**g_3**	**g_4**
Actual	**g_1**	53	2	0	3
	g_2	1	55	3	0
	g_3	0	6	45	0
	g_4	9	0	0	48

Table 13: Confusion Matrix for SVM, K = Radial/Sigmoid.

		Predicted			
		g_1	g_2	g_3	g_4
Actual	g_1	53	0	2	3
	g_2	1	55	3	0
	g_3	0	6	45	0
	g_4	11	0	0	46

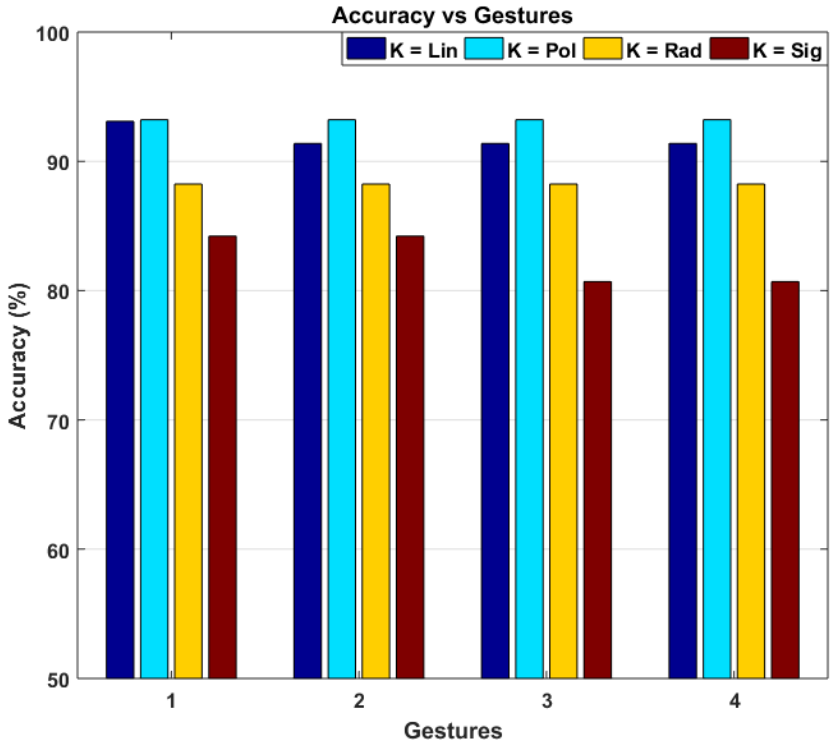

Figure 37: Accuracy of SVM for 4 Kernels on 4 gestures.

Figure 40 shows a comparison between the performances of all machine learning models used on our dataset. We notice that all the models perform better than the baseline approach, showing the benefit of applying these techniques on our dataset. LDA and SVM radial and sigmoid models perform similarly. However, the linear and polynomial

SVM models perform somewhat better than the others.

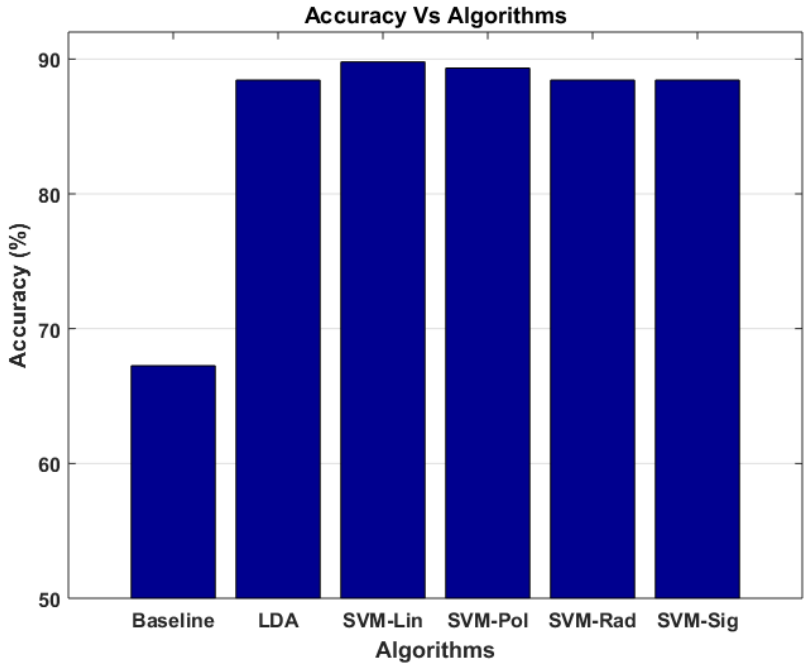

Figure 38: Accuracy of applied techniques compared to a baseline.

5.3 Summary

In this chapter, we illustrate our methodology for hand gesture recognition using LDA and SVM. In the chapter, we describe architecture of the system, signal acquisition, and the feature vector. In addition, we introduce the four different static hand gestures that we consider during data collection. We implement LDA and SVM techniques so that our framework can learn more about measurement errors in order to achieve better compensation for these errors. All these techniques achieve an accuracy above 88.0% compared to the baseline performance of 67.1%. This demonstrates the significant benefit of employing machine learning techniques in this context.

6 Comparing Dynamic Hand Gestures in LM Using MDTW

We explore the efficacy of performing a comparison between a dynamic hand gesture that would be performed by a patient with a reference dynamic hand gesture that is recommended by a physiotherapist using Multi-dimensional Dynamic Time Warping (MDTW). MDTW quantifies how similar or different two dynamic hand gestures are from each other [20]. In our approach, factors such as difference in duration of hand gestures, rotation of hand, reasonable distance from LM sensor etc. do not have significant impact on the performance of hand gestures (See Sections 6.2.2, 6.2.3, 6.2.4, 6.2.5, and 6.2.6 for further information).

DTW can be applied to establish to what extent any two time-series are similar or different. Both DTW and MDTW techniques utilise a distance metric between a query time-series and a reference one and create a discriminating value: a low distance value when the two time-series are similar and a high distance value when the two time-series are different [22]. DTW is employed to measure the similarity between hand gesture orientation trajectories [149], increase recognition accuracy by categorising hand movements [150], compare the hand movement direction and estimate the difference, and hence reflects the characteristics of the hand gesture direction [151], etc.

In this chapter we implement both DTW and MDTW, however there exist other possible dissimilarity measures such as the Longest Common SubSequence (LCSS) [20, 37], Edit distance with Real Penalty (ERP) [20], Edit Distance on Real sequences (EDR) [20, 38], and Time Warp Edit Distance (TWED) [20, 39]. However DTW and MDTW have an advantage since their implementation is simple and efficient. In addition, DTW and MDTW are superior because it is not essential that both time-series being compared are of equal length as required by typical distances and this behaviour is termed elasticity [20].

During MDTW implementation, we selected four dynamic gestures (See Section 6.1.4) that are similar to the hand gestures/exercises recommended by NHS (National Health Service) physiotherapists [152-154] for patients recovering from rheumatology related complications and those performing hand therapy related exercises. It is important

to note that in [152-154], MDTW or any signal processing technique was not implemented as done in this thesis. We have referenced these hand gestures to illustrate that this study could be extended to a hand rehabilitation setting.

6.1 Methodology

6.1.1 Architecture of the Proposed System

We obtain an input signal when a hand is placed above the LM that is connected to a computing device using a USB cable. On a computing device, the LM application and a Java customised application are started simultaneously and frames of data that represent palm and finger movements are displayed. We then calculate joint angles of the fingers from the frames of data and, consequently, obtain feature vectors that represent various dynamic hand gestures.

After obtaining feature vectors that represent both the dynamic query hand gesture and the dynamic reference hand gesture, we apply MDTW on both feature vectors and the minimum-distance value of a warp path is obtained. The minimum-distance value signifies how similar or different the two hand gestures are. A low minimum-distance value implies the two hand gestures that are compared are similar. On the other hand, a high minimum-distance value implies the two hand gestures vary to a certain extent. Figure 41 illustrates the architecture of the MDTW comparison process.

6.1.2 Description of the Feature Vector

A dynamic hand gesture is represented using a set of fourteen measured finger joint angles that evolve over time. The fourteen finger joint angles that constitute the feature vector are illustrated in Table 14. Other features, e.g. the angles between the finger and palm position, are sufficient to be included in the feature vector. However, we choose to limit the feature vector to only finger joint angles since including other features does not contribute a considerable change to the experimental results regarding our application of MDTW.

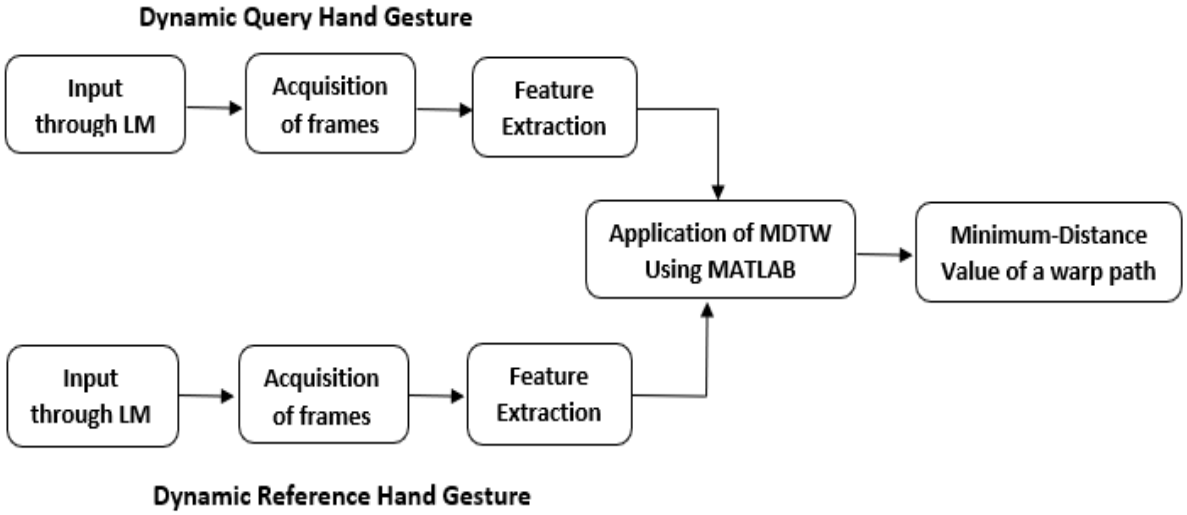

Figure 39: Architecture of the proposed MDTW implementation set-up.

6.1.3 Robotic Hand

In order to simulate the repetitive nature of dynamic hand gestures and to provide carefully controlled experimental conditions, we used a robotic hand during data collection. In addition, compared to a human hand, a robotic hand permits gestures to be performed frequently in a short time provided the movements of all the necessary servo controllers are suitably programmed for a specific hand gesture. Furthermore, the dynamic hand gestures are performed with minimum positional or temporal errors and this provides a suitable working environment to reduce errors during experimentation. The robotic hand was fitted with a glove so that it can be easily recognised by the LM. Figure 42 illustrates the robotic hand performing a dynamic hand gesture.

6.1.4 Gestures Under Consideration

We have considered 4 different dynamic hand gestures. Dynamic hand **gesture 1** is when a hand is performing a full fist gesture where all the five fingers move close to the palm. Dynamic hand **gesture 2** is where only the thumb and index move close to the palm whereas the rest of the 3 fingers remain stationary. Dynamic hand **gesture 3** is performed when the middle, ring and little fingers move close to the palm whereas the thumb and

index do not move. Lastly, dynamic hand **gesture 4** involves motion of the index, middle, ring and little fingers moving close to the palm whereas the thumb remains stationary. Figure 43 illustrates two of these dynamic hand gestures.

Table 14: Feature Vector in the Experimental Set-up.

Finger	MCP Joint	PIP Joint	DIP/IP Joint
Thumb	θ_1		θ_2
Index	θ_3	θ_4	θ_5
Middle	θ_6	θ_7	θ_8
Ring	θ_9	θ_{10}	θ_{11}
Little	θ_{12}	θ_{13}	θ_{14}

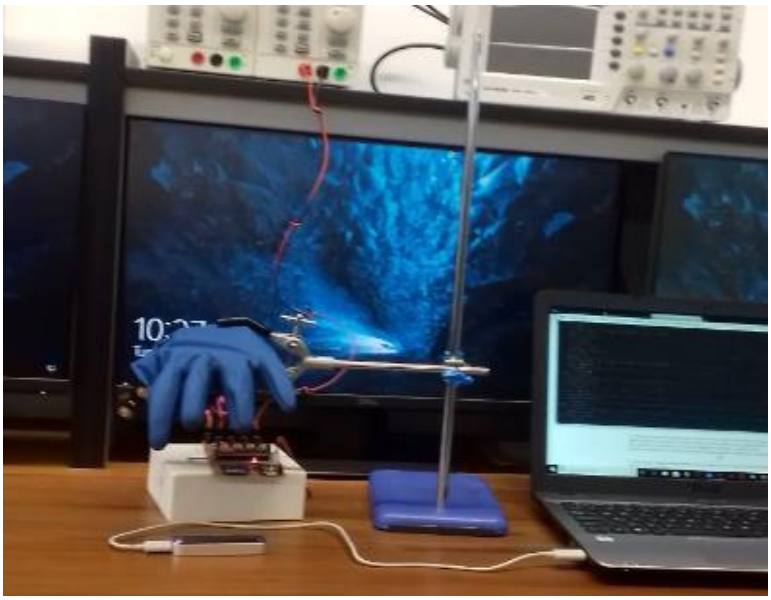

Figure 40: An illustration on how the robotic hand performs a dynamic gesture.

These selected hand gestures or exercises are similar to those recommended by Salford Royal NHS hospital [152] and Milton Keynes University NHS hospital [153] for patients performing rheumatology related hand exercises. In addition, Derbyshire

Community NHS Health Services [154] recommends similar exercises for hand therapy.

Figure 41: All the four hand gestures to facilitate implementation of MDTW.

6.2 Results and Discussion

6.2.1 Evaluation of MDTW

Here, we compare a particular hand gesture with itself, meaning the same vector describing how the fourteen finger joints evolve over successive time frames is compared to itself. When the comparison is completed, the minimum-distance value of a warping path obtained is **0.0º**. This is a perfect result since the similarity is **100.0%**.

6.2.2 Comparing Different Gestures to a Reference Gesture

Here we designed 4 different hand gesture comparisons where in comparison 1 i.e. **cf. #1**, hand gestures 2, 3 and 4 are compared to hand gesture 1. In **cf. #2**, hand gestures 1, 3 and 4 are compared to hand gesture 2. In **cf. #3**, hand gestures 1, 2 and 4 are compared to hand gesture 3 and finally in **cf. #4**, hand gestures 1, 2 and 3 are compared to hand gesture 4.

From Figure 44, hand gesture 1 and hand gesture 4 are similar since a minimum-distance value of **35.9º** is obtained. Further, hand gesture 2 and hand gesture 3 experience

the largest variation since a high value of minimum-distance of **248.5º** is obtained. From our experimental design, this makes sense since hand gesture 1 is nearly identical to hand gesture 4 whereas hand gesture 2 can be regarded as the direct opposite to hand gesture 3.

6.2.3 Comparing Gestures at Varied Distances from the LM

In this experiment we considered four different configurations i.e. **D1** is 5 cm to the right of the centreline of the LM from the middle of the palm of the robotic hand whereas **D2** is 5 cm to the left. **D3** is 5 cm in front of the centreline of the LM from the middle of the palm of the robotic hand whereas **D4** is 5 cm behind. During the execution of a dynamic hand gesture, we maintained a moderate speed, a vertical distance of 15 cm from the surface of the LM to the palm of the robotic hand, and the robotic hand was fixed in a horizontal position with its palm facing downwards.

For comparison purposes, **D1D2** implies the dynamic hand gesture performance at **D1** is compared with the dynamic hand gesture performance at **D2**. This notation is applied to the remaining cases as well, for example **D3D4** implies dynamic hand gesture performance at **D3** is compared with dynamic hand gesture performance at **D4.**

We choose a distance parameter as one of the parameters where a hand performing gestures relative to the LM can experience measurement errors as recorded by the LM. However, when the distances are relatively close to the LM, these measurement errors are inconsequential. This has been discussed in Section 4.2.2 (b), Chapter 4. This section considers experimental results where measurement errors in the LM are investigated and characterised with respect to lateral distances from the LM.

From Figure 45, the minimum-distances range is from around 5.0 to 12.5^{o}. These are quite low values compared to minimum-distance values obtained in Figure 44 where we compared dynamic hand gestures to a specific gesture. Hence positioning a reasonable distance from the centreline of the LM to the middle of the palm of a hand does not affect gesture performance significantly.

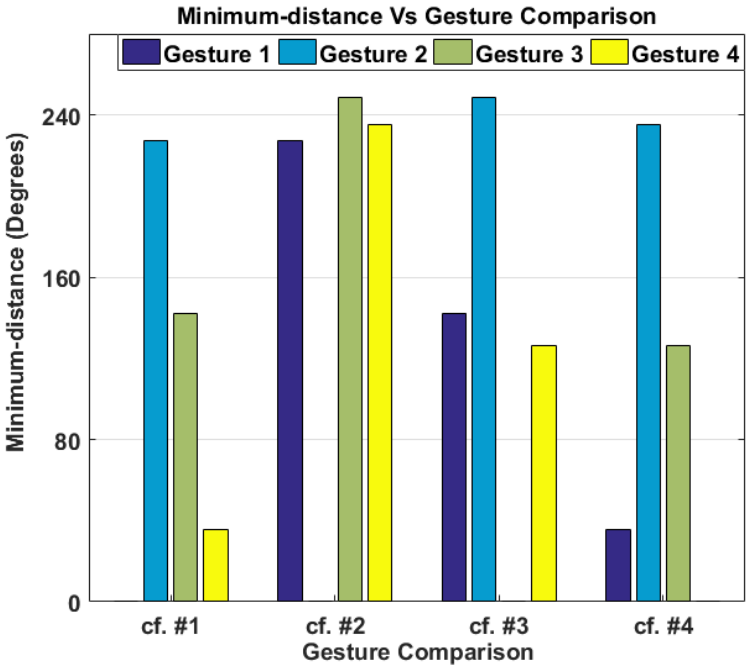

Figure 42: Minimum-distance against gesture comparison.

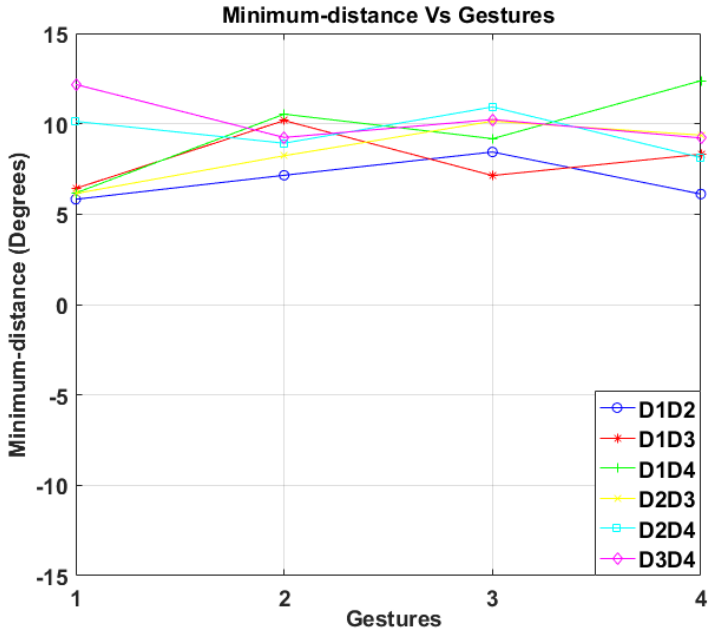

Figure 43: Varied distances for hand gesture comparisons.

6.2.4 Comparing Gestures at Varied Speeds

In this experiment we arranged four different configurations, i.e. **S1** is when a dynamic hand gesture is performed at a moderate speed, **S2** is when a dynamic hand gesture is performed at a speed twice as fast as a moderate one, **S3** is when a dynamic hand gesture is performed at a speed three times faster than the moderate case and **S4** is when a dynamic hand gesture is performed at a speed four times faster than the moderate one.

For comparison purposes, **S1S2** implies the dynamic hand gesture performance at **S1** is compared with the dynamic hand gesture performance at **S2**. This notation applies to the remainder of the results as well, for example **S3S4** implies dynamic hand gesture performance at **S3** is compared with dynamic hand gesture performance at **S4**. During a dynamic hand gesture evaluation we placed the robotic hand at its centreline, maintaining a vertical distance of 15 cm from the surface of the LM to the palm of the robotic hand, and the robotic hand was fixed in a horizontal position with its palm facing downwards.

As illustrated in Figure 46, the minimum-distances range from approximately 3.0 to 11.0^{o}. It is important to note that these values are in the same range as those obtained when comparisons are made based on distances from the LM to the robotic hand (Figure 45). This implies variable speed of hand gestures does not significantly affect gesture comparison performance.

6.2.5 Comparing Gestures when the Robotic Hand is Rotated

In this experiment we rotate the robotic hand while performing dynamic hand gestures for four different scenarios, i.e. **R1**, **R2**, **R3**, and **R4**. **R1** is when the palm of the robotic hand is in horizontal position and facing downwards. **R2** is when the robotic hand is roughly rotated at 30^{o} and its thumb facing upwards. **R3** is when the robotic hand is roughly rotated at 60^{o} and its thumb facing upwards. **R4** is when the robotic hand is in a vertical position and its thumb facing upwards. During this dynamic hand gesture evaluation we placed the robotic hand at its centreline, maintained a vertical distance of 15 cm from the surface of the LM to the palm of the robotic hand, and maintained a moderate speed.

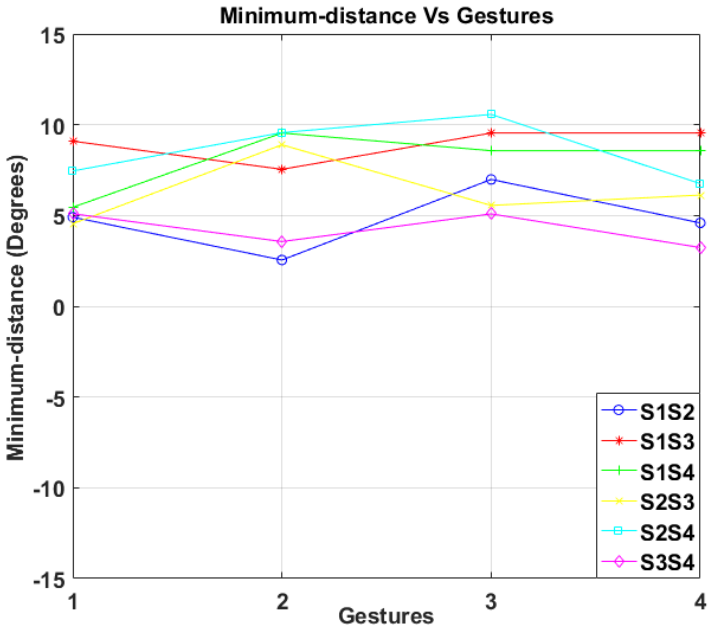

Figure 44: Varied speeds for hand gesture comparisons.

For comparison purposes, **R1R2** implies the dynamic hand gesture performance at **R1** is compared with the dynamic hand gesture performance at **R2**. This notation applies to the remaining results as well, for example **R3R4** implies dynamic hand gesture performance at **R3** is compared with dynamic hand gesture performance at **R4**.

We choose rotation of a robotic hand as one of the parameters when performing hand gestures relative to the LM to explore measurement errors as recorded by the LM. However sensible rotations of hand gestures relative to the LM lead to insignificant measurement errors. This has been discussed in Section 4.2.2 (d), Chapter 4 where experimental results provide the measurement errors as the hand is rotated.

As shown in Figure 47, the minimum-distances range from approximately 4 to 14^{o}. These minimum-distance values are in the same range as those in Figures 45 and 46. However, it is important to note that the **RIR2** comparison registers the lowest minimum-distance values. On the other hand, **RIR4** registers the highest minimum-distance values. This may suggest that a physiotherapist should encourage patients to always avoid unnecessary rotations of their hands.

From the preceding analysis of the results, in regard to hand gesture comparisons with the reference gestures, there is no significant concern about operating with a reasonable horizontal distance from the LM to the hand of a user, sensible rotations of the hand, and varying speed of gestures performed by different users as these factors do not significantly affect the comparison performance. However, it is important to note that we maintained a vertical distance of roughly 15 cm from the screen of the LM to the palm of the robotic hand in all our experiments simply because, in our previous work [34], we established that accurate LM readings are obtained around this height.

6.2.6 Detailed Comparison for Hand Gestures

Here we undertake a comprehensive comparison of dynamic hand gestures performed in all the scenarios we have explained in Sections 6.3.3 through to 6.3.5 i.e. at various distances (D1, D2, D3 and D4), at various speeds (S1, S2, S3 and S4), and at various rotations (R1, R2, R3, and R4). Tables 15 to 18 record these comparisons for all of the four dynamic hand gestures. All the minimum-distance values in these tables are measured in degrees.

As illustrated in Table 15, the lowest minimum-distance value is 0.0^{o} and the highest value is 12.8^{o}. From Table 16, it can be observed that minimum-distance values range from 0.0^{o} to 18.2^{o}. In Table 17, the minimum-distance values range from 0.0^{o} to 17.4^{o} and in Table 18, the minimum-distance values range from 0.0^{o} to 18.8^{o}. Zero degrees was obtained when a dynamic hand gesture was compared to itself, and this is expected since the similarity is 100.0%. On the other hand, other minimum-distance values are lower compared to those obtained when comparison is made with different hand gestures. This signifies the acceptable distance from the LM to the hand and indifference to speed of hand gestures during performance of hand exercises. Sensible rotations of the hand also have little impact on hand gesture comparison performance.

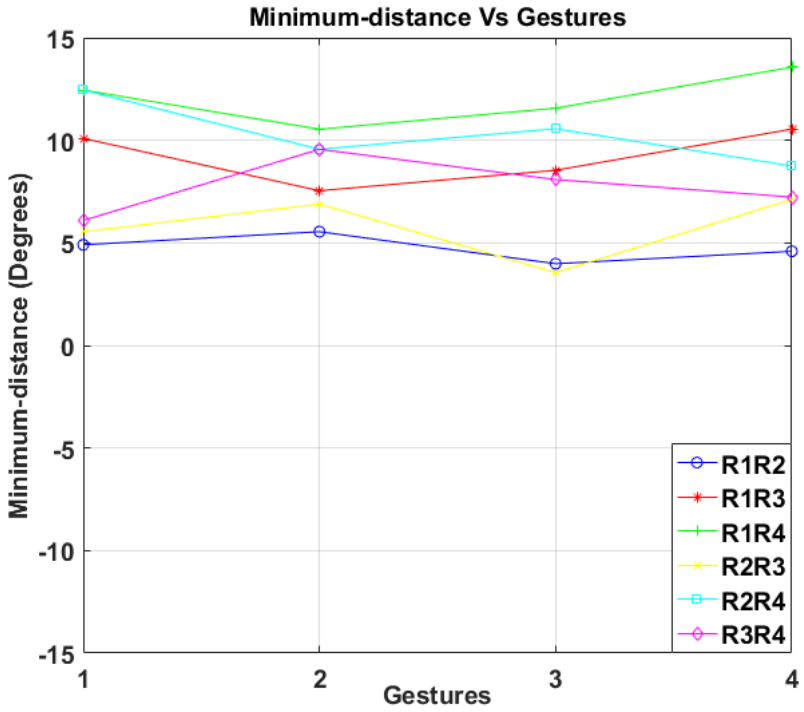

Figure 45: Varied rotations for hand gesture comparisons.

6.3 Summary

In this chapter, we proposed, implemented and evaluated Multi-Dimensional Dynamic Time Warping (MDTW) to establish how similar or different a query dynamic hand gesture is in relation to a reference dynamic hand gesture. We described our methodology where we presented the architecture of the proposed system, described the feature vector, showing the robotic hand we used during experimentation, and described the gestures we considered during experimentation.

Experimental results show that MDTW is robust in the way that it can distinguish quite different hand dynamic gestures by yielding a high value of a minimum-distance metric. Furthermore, the method is capable of producing a low value of a minimum-distance if the dynamic hand gestures under comparison are similar. In addition, the results show that when we compare a specific hand gesture with itself, we obtain a minimum-distance value of 0.0^{o}. This is a perfect result since the similarity is 100.0%. Furthermore, when we compare two closely similar hand gestures, i.e. gesture 1 and gesture 4, a

minimum-distance value of 35.9^o is obtained. However, when we compare two quite different gestures, i.e. gesture 2 and gesture 3, a minimum-distance value of 248.5^o is obtained.

In Section 6.2.6, a detailed comparison of the same hand gestures at various reasonable distances, at various speeds, and at sensible hand rotations is performed. The highest minimum-distance obtained is 18.8^o. This implies that 35.9^o is not acceptable when the same hand gestures are compared. However, 35.9^o is acceptable for different hand gestures and this indicates that the two hand gestures being compared are similar as illustrated in Section 6.2.2.

Table 15: Detailed Comparison for Hand Gesture 1.

					Q	U	E	R	Y				
		D1	D2	D3	D4	S1	S2	S3	S4	R1	R2	R3	R4
	D1	0.0	5.8	6.4	6.2	6.4	7.3	5.9	6.1	7.4	7.1	11.3	9.9
R	D2	5.8	0.0	6.1	10.1	8.7	12.3	7.7	8.3	10.3	5.6	9.4	10.5
E	D3	6.4	6.1	0.0	12.2	6.4	8.8	11.2	9.3	10.2	11.6	7.8	9.3
F	D4	6.2	10.1	12.2	0.0	8.4	12.3	9.5	10.4	7.9	4.8	12.2	7.8
E	S1	6.4	8.7	6.4	8.4	0.0	4.9	9.1	5.5	10.3	12.4	9.4	12.6
R	S2	7.3	12.3	8.8	12.3	4.9	0.0	4.5	7.5	7.9	12.4	7.9	12.8
E	S3	5.9	7.7	11.2	9.5	9.1	4.5	0.0	5.1	6.8	10.7	12.3	8.6
N	S4	6.1	8.3	9.3	10.4	5.5	7.5	5.1	0.0	7.5	9.1	3.9	11.8
C	R1	7.4	10.3	10.2	7.9	10.3	7.9	6.8	7.5	0.0	4.9	10.1	12.5
E	R2	7.1	5.6	11.6	4.8	12.4	12.4	10.7	9.1	4.9	0.0	5.5	12.5
	R3	11.3	9.4	7.8	12.2	9.4	7.9	12.3	3.9	10.1	5.5	0.0	6.1
	R4	9.9	10.5	9.3	7.8	12.6	12.8	8.6	11.8	12.5	12.5	6.1	0.0

Table 16: Detailed Comparison for Hand Gesture 2.

					Q	U	E	R	Y			
		D1	D2	D3	D4	S1	S2	S3	S4	R1	R2	R3	R4
	D1	0.0	7.1	10.2	10.5	9.7	12.1	8.1	7.9	10.2	12.4	11.6	9.6
R	D2	7.1	0.0	8.2	8.9	5.9	6.4	11.0	9.3	5.6	10.4	13.4	14.5
E	D3	10.2	8.2	0.0	9.2	7.3	9.1	12.4	8.0	11.5	13.6	12.4	7.4
F	D4	10.5	8.9	9.2	0.0	15.2	14.8	7.9	12.4	16.4	14.7	11.3	14.9
E	S1	9.7	5.9	7.3	15.2	0.0	2.5	7.5	9.5	15.9	6.8	17.1	10.3
R	S2	12.1	6.4	9.1	14.8	2.5	0.0	8.9	9.6	12.1	7.0	6.2	14.0
E	S3	8.1	11.0	12.4	7.9	7.5	8.9	0.0	3.6	7.3	11.4	8.4	16.3
N	S4	7.9	9.3	8.0	12.4	9.5	9.6	3.6	0.0	18.2	9.2	7.5	13.4
C	R1	10.2	5.6	11.5	16.4	15.9	12.1	7.3	18.2	0.0	5.5	7.5	10.5
E	R2	12.4	10.4	13.6	14.7	6.8	7.0	11.4	9.2	5.5	0.0	6.9	9.6
	R3	11.6	13.4	12.4	11.3	17.1	6.2	8.4	7.5	7.5	6.9	0.0	9.6
	R4	9.6	14.5	7.4	14.9	10.3	14.0	16.3	13.4	10.5	9.6	9.6	0.0

Table 17: Detailed Comparison for Hand Gesture 3.

					Q	U	E	R	Y			
		D1	D2	D3	D4	S1	S2	S3	S4	R1	R2	R3	R4
	D1	0.0	8.4	7.1	9.2	9.3	10.4	5.5	16.3	7.4	13.5	14.4	8.0
R	D2	8.4	0.0	10.1	10.9	5.9	10.3	7.7	9.3	5.7	7.6	13.2	4.9
E	D3	7.1	10.1	0.0	10.2	11.4	6.6	8.5	9.5	7.9	6.4	11.6	9.3
F	D4	9.2	10.9	10.2	0.0	8.4	9.5	12.2	10.4	4.8	16.4	17.4	8.0
E	S1	9.3	5.9	11.4	8.4	0.0	7.0	9.5	8.6	12.3	7.0	6.1	13.6
R	S2	10.4	10.3	6.6	9.5	7.0	0.0	5.6	10.6	14.9	15.7	6.3	12.8
E	S3	5.5	7.7	8.5	12.2	9.5	5.6	0.0	5.1	7.1	9.0	16.7	15.7
N	S4	16.3	9.3	9.5	10.4	8.6	10.6	5.1	0.0	5.0	11.7	15.3	14.5
C	R1	7.4	5.7	7.9	4.8	12.3	14.9	7.1	5.0	0.0	4.0	8.5	11.6
E	R2	13.5	7.6	6.4	16.4	7.0	15.7	9.0	11.7	4.0	0.0	3.6	10.6
	R3	14.4	13.2	11.6	17.4	6.1	6.3	16.7	15.3	8.5	3.6	0.0	8.1
	R4	8.0	4.9	9.3	8.0	13.6	12.8	15.7	14.5	11.6	10.6	8.1	0.0

.

Table 18: Detailed Comparison for Hand Gesture 4.

						Q	U	E	R	Y			
		D1	D2	D3	D4	S1	S2	S3	S4	R1	R2	R3	R4
	D1	0.0	6.1	8.3	12.4	7.4	13.7	16.7	10.4	9.6	18.4	7.2	14.5
R	D2	6.1	0.0	9.4	8.1	6.9	14.4	16.3	13.7	10.7	7.5	18.5	15.0
E	D3	8.3	9.4	0.0	9.2	7.2	10.5	6.2	14.6	12.3	14.6	9.9	9.2
F	D4	12.4	8.1	9.2	0.0	5.3	15.7	12.0	13.1	11.4	9.1	7.3	11.6
E	S1	7.4	6.9	7.2	5.3	0.0	4.6	9.5	8.6	7.7	13.3	9.4	4.5
R	S2	13.7	14.4	10.5	15.7	4.6	0.0	6.1	6.8	11.3	17.6	9.7	12.5
E	S3	16.7	16.3	6.2	12.0	9.5	6.1	0.0	3.2	7.8	14.7	12.9	10.6
N	S4	10.4	13.7	14.6	13.1	8.6	6.8	3.2	0.0	18.8	13.4	14.6	12.1
C	R1	9.6	10.7	12.3	11.4	7.7	11.3	7.8	18.8	0.0	4.6	10.5	13.6
E	R2	18.4	7.5	14.6	9.1	13.3	17.6	14.7	13.4	4.6	0.0	7.1	8.8
	R3	7.2	18.5	9.9	7.3	9.4	9.7	12.9	14.6	10.5	7.1	0.0	7.2
	R4	14.5	15.0	9.2	11.6	4.5	12.5	10.6	12.1	13.6	8.8	7.2	0.0

7 Applying PCA to Improve the Performance of LDA, SVM and MDTW

In Chapter 4, we have implemented PCA to reduce absolute errors in LM finger joint measurements. However in this chapter PCA is implemented on datasets of LDA, SVM and MDTW to improve the performance of LDA, SVM, and MDTW by selecting the best combination of principal components (PCs) when forming feature vectors during the PCA implementation.

7.1 Implementing PCA to Improve the Performance of LDA and SVM

We have used the same dataset as illustrated in Section 5.2.1 that was employed to recognise static hand gestures using two popular machine learning techniques i.e. LDA and SVM.

We apply PCA on the whole dataset and then employ LDA and SVM to recognise static hand gestures. As described in Section 5.2.1, out of a total of 600 samples collected, we used 62.5% of the 600 samples in training and the remaining 37.5% in testing.

7.1.1 Experimental Set-up to Improve the Performance of LDA and SVM Using PCA

We apply PCA to the raw dataset to see whether the performance of LDA and SVM could be improved. The steps involved are described below.

Step 1: Our dataset is a matrix X that comprises of 600 rows (observations) and 14 columns (variables of finger joint angles).

Step 2: We input X in the R statistical package using 'prcomp()' function and eigenvectors or principal components (PCs) together with their respective standard deviations are obtained as output. From the obtained standard deviations, we compute the variance that corresponds to each of the PCs.

Step 3: In our experiments, the R statistical package generated 14 PCs. From these PCs, we formed three feature vectors as follows. The feature vector V_1 constituted 13 PCs after discarding PC1, V_2 constituted 13 PCs after discarding PC2, and V_3 constituted 12 PCs after discarding PC1 and PC2. This implies PCA was implemented in three different ways. PC1 had 68.0% of the total variability whereas PC2 had 25.8%.

Step 4: We derive the PCA integrated dataset Y as

$$\mathbf{Y} = \mathbf{XVV}^T \quad (7\text{-}1)$$

where v^T is the transpose of V.

It is important to note that we had 3 different feature vectors i.e. V_1, V_2 and V_3, hence giving us 3 different PCA implementations. This resulted in three different PCA results i.e. PCA(1), PCA(2), and PCA(3).

7.1.2 Results and Discussion Regarding the Performance of LDA and SVM Using PCA

Here we analyse how PCA performs regarding the accuracy of LDA and SVM techniques. Accuracy has been defined and described in Section 5.2.2, Chapter 5. We considered the overall performance of LDA and SVM on recognising the four static hand gestures. As shown in Figure 48, there is an improvement in accuracy from 88.4% to 91.4% after PCA(1) has been implemented on LDA. For SVM Linear, PCA(1) registered an improvement from 89.7% to 93.6%, for SVM Polynomial, the improvement was from 89.3% to 93.9%, for SVM Radial, the improvement was from 88.4% to 92.5% and for SVM Sigmoid, the improvement was from 88.4% to 90.1%.

We registered a desired outcome regarding the improvement in accuracy when selecting the feature vector used in implementing PCA discarded PC1 that contained 68.0% variability. However when the feature vector selected discarded both PC1 and PC2 that both contained 93.8% variability, the accuracy of LDA and SVM reduced greatly and from Figure 48, this is shown as the PCA(3) performance. PCA can only perform well if we select the right feature vector i.e. when we discard the first principal component (PC1) and the rest form the feature vector. Since PCA(1) registers a significant improvement compared to PCA(2) and PCA(3), PCA can be employed to improve the accuracy of LDA and SVM in recognising the static hand gestures.

7.2 Implementing PCA to Improve the Performance of MDTW

We have used the same dataset that was used in Section 6.3.2 to compare dynamic hand gestures using MDTW. It is important to note here that we had four different matrices each representing one of the four dynamic gestures. A description of this type of

matrix can be found in Section 6.1.3. We then apply PCA on each matrix followed by implementing MDTW as explained in the next section.

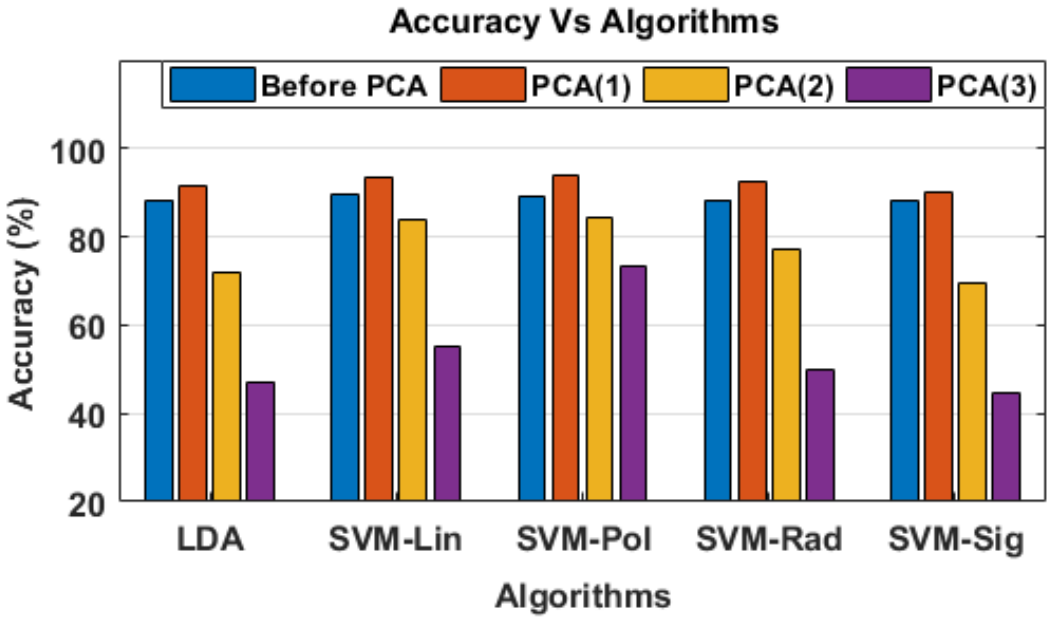

Figure 46: PCA Performance on LDA and SVM.

7.2.1 Experimental Set-up to Improve the Performance of MDTW Using PCA

We apply PCA to the raw dataset to see whether the performance of MDTW could be improved. The steps involved are described below.

Our dataset contains four matrices, each matrix representing one of the four dynamic hand gestures. It is important to note that rows represent the joint angles, and columns represent successive time frames. Since the steps are exactly the same as described in Section 7.1.1 we shall only highlight the differences.

The first difference was that we had to transpose the matrices because originally the variables (joint angles) were represented in rows. We also had to transpose other mathematical terms in Equation (7.1) so that they suit the implementation of PCA with MDTW accordingly. Such terms include V and μ.

As in Section 7.1, we had 3 different feature vectors i.e. V_1, V_2 and V_3 for each of the dynamic hand gesture giving us 3 different PCA implementations. This resulted in three different PCA results i.e. PCA(1), PCA(2), and PCA(3). In particular, V_1 was obtained after discarding PC1, while retaining the remaining 13 principal components. V_2 and V_3 were also obtained after discarding PC2 and PC3, respectively. Table 19 shows the percentage variability of PC1, PC2, and PC3 for each of matrix i.e. each of the dynamic hand gesture.

Table 19: Percentage Principal Components Variability for each Gesture.

Gesture	Percentage Variability for PC1	Percentage Variability for PC2	Percentage Variability for PC3
1	76.4	16.4	6.8
2	75.3	16.4	6.8
3	76.7	16.4	6.9
4	76.7	16.4	6.9

7.2.2 Results and Discussion Regarding the Improved Performance of MDTW Using PCA

The lower the minimum-distance value, the better the MDTW performance is in terms of establishing the extent of similarity of the two dynamic hand gestures. Overall, PCA(1) registers more significant results compared to PCA(2) and PCA(3). This is because all the minimum-distance values obtained by PCA(1) are lower compared to those obtained before PCA is implemented. PCA(2) performs fairly well in obtaining lower values in comparison to those obtained before implementing PCA. The only situation where a higher value is obtained is during gesture comparison of G1 with G2 where a value of 39.7^o is obtained whereas the original value was 35.9^o. PCA(3) registers three values out of six that are higher than the original minimum-distance values. These three values are obtained when gesture comparison was done between G1 with G2, G1 with G4 and G2 with G4. This means the implementation of PCA(3) does not give us the desired results. Since PCA(1) is capable of lowering all the minimum-distance values in comparison to those obtained

before PCA implementation, it is beneficial to implement PCA on the raw dataset of dynamic hand gestures before employing MDTW to improve accuracy.

7.3 Summary

In this chapter, we implemented PCA to see if it could improve on the performance of LDA, SVM and MDTW. We have noted that the feature vector should be carefully selected after discarding one or more PCs. In particular, when we discard the first PC and retain the remainder as the feature vector, the PCA implementation give us significant results regarding an improvement in the performance of LDA, SVM and MDTW. However, when we try to alter the feature vector by discarding other PCs, such as the second and third one, the results were not as good as those when we discarded the first PC. Therefore, for our experimental datasets, discarding the first PC in the feature vector proved to be more beneficial than discarding the second, third or both.

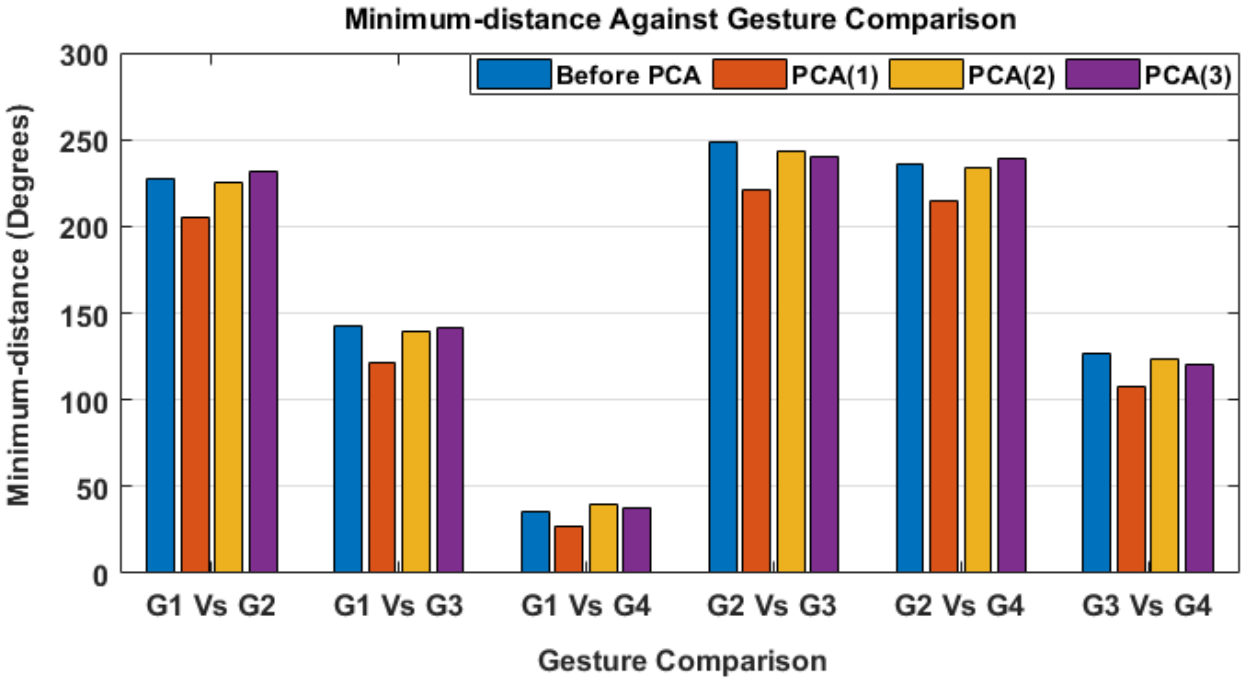

Figure 47: PCA Performance on MDTW.

8 Conclusion

In this thesis, we have examined a low-cost device that can be used in a hand gesture recognition framework although users with missing fingers or shaky hands are not considered in this research. During the initial stages of the research, we noticed that absolute errors are triggered by four parameters, i.e. elevation, lateral (side-to-side), forward-backward, and rotation movements of the hand relative to the LM. We used an artist's hand placed above the LM. The artist's hand is more accurate than a human hand in performing static hand gestures as it can maintain a fixed position as long as is necessary. We further describe the experimental set-up to study variation of finger joint angles with errors, to determine whether absolute errors are consistent, and then to compensate for absolute errors using PCA. We discovered that absolute errors are mainly experienced when hand movements are away from the centreline of the LM. We proposed and implemented PCA on the raw data in order to compensate for these absolute errors (AEs). AEs were reduced by 37.5%, 28.3%, 33.0%, and 22.4% for the experimental results of elevation, lateral (side-to-side), forward-backward, and rotation, respectively. The experimental results have demonstrated that using PCA is feasible, allowing meaningful benefits to be obtained.

We then performed additional research and formulated a mathematical problem for recognising static hand gestures. This problem formulation was further extended to suit the application of LDA and SVM machine learning techniques. We implemented LDA and SVM techniques so that our framework can learn more about measurement errors in order to achieve better compensation for these errors. All these techniques achieved an accuracy above 88.0% compared to the baseline performance of 67.1%. This demonstrates the significant benefit of employing machine learning techniques in the context of performing hand gestures.

In a further research contribution, we proposed, implemented and evaluated Multi-dimensional Dynamic Time Warping (MDTW) to establish how similar or different a query dynamic hand gesture could be in relation to a reference dynamic hand gesture. The approach is robust in the way that it can distinguish quite different hand dynamic gestures by yielding a high value of a minimum-distance metric. Furthermore, the method can

produce a low value of minimum-distance if the hand dynamic gestures under comparison are very similar. Therefore, in the context of dynamic hand gesture comparison, a clear distinction can be obtained between a query hand gesture and a reference hand gesture.

In the final part of our research, we applied PCA to investigate whether it could improve the performance of LDA, SVM and MDTW. This was achieved on condition that the feature vector was carefully selected during PCA implementation. This was executed in such a way that, the first PC was discarded while retaining the remaining PCs to form a feature vector. We further investigated altering the feature vector by discarding other PCs like the second and third PC and the results obtained were not as good as those obtained when only the first PC was discarded. Therefore, to obtain significant results in regard to improving the performance of LDA, SVM and MDTW, careful PCs selection to form a feature vector proved to be relevant when implementing PCA.

8.1 Future Work

In the future we would like to focus on our part of research where we implemented Multi-dimensional Dynamic Time Warping (MDTW) to compare dynamic hand rehabilitation gestures that would be performed by a patient relative to hand gestures prepared by a physiotherapist. Although the four different hand dynamic gestures were sufficient for the implementation and evaluation of MDTW, it would be desirable to extend the number of gestures. In this research we were constrained by the robotic hand performing only flexing and extending finger gestures. It was unable to execute complex hand gestures like abduction and adduction of fingers. With the appropriate resources, we could extend on the types of hand gesture examined by possibly employing a different robotic hand that can perform abduction and adduction of fingers gestures. The benefit of using a robot hand is two-fold. First it avoids the needs for ethical approval. Second it is able to repeat gestures in a controlled manner.

Owing to limited time, we were unable to extend our research to a hand rehabilitation setting. If we had extended our research, we could have recruited patients with hand related injuries like those recovering from stroke. However, it would be interesting to recruit both patients with hand related injuries and healthy individuals and

evaluate how the two groups perform hand rehabilitation gestures. The obtained results could provide significant benefits in the field of hand therapy where a physiotherapist can generate more realistic feedback to the patient. However, this would require the implementation of a suitable feedback mechanism so the patients could assess how well they were performing the exercise and adapt their actions accordingly.

Finally, since the recognition rate in LM varies slightly depending on the speed of hand gestures, this could also be an interesting area for future research. In their work [155], it is reported that the LM is capable of recognising the user hand gesture precisely by obtaining the fingertip position and direction details of the hand. Experimental results demonstrate that implemented scheme can effectively solve the issues of reduction in the recognition rate of the LM owing to the measurement errors attributable to finger overlap [155]. In their research [156] they proposed and implemented a scheme to improve the recognition rate of the LM. The recognition rate is hampered owing to the fact that fingertips covered some part of the palm [156]. This made the part of the palm undetectable to the LM sensor. Their approach was centred on data correction of the fingertip direction by employing a Hidden Markov Model (HMM). By applying the Viterbi algorithm, the palm can be more easily detected by the LM [134] hence improving the recognition rate. The research work [155-157] could form the basis for further research regarding variability in the recognition rate of LM.

9 References

[1] A. Hadjidj et al., “Wireless sensor networks for rehabilitation applications: challenges and opportunities”, Journal of Networks and Computer Applications, January 2013, pp. 1-15, 2013.

[2] World population ageing 2013, United Nations, New York, NY, USA, pp. 8-10, 2013.

[3] S. Patel et al., “A review of wearable sensors and systems with application in rehabilitation”, Journal of Neuroengineering and Rehabilitation, pp. 1-17, April 2012.

[4] K. Vamsikrishna et al., “Computer-vision-assisted palm rehabilitation with supervised learning”, IEEE Transactions on Biomedical Engineering, vol. 63, no. 5, pp. 991-1001, May 2016.

[5] C. Tan et al., “Game-based human computer interaction using gesture recognition for rehabilitation,” in Proc. IEEE International Conference on Control System, Computing and Engineering (ICCSCE), pp. 344–349, Nov. 2013.

[6] C. Redd and S. Bamberg, “A wireless sensory feedback device for real-time gait feedback and training”, IEEE/ASME Transactions on Mechatronics, vol. 17, no. 3, pp. 425-433, 2012.

[7] N. Alberto et al., “An intelligent game engine for the at-home rehabilitation of stroke patients”, IEEE Second International Conference on Serious Games and Applications for Health (SeGAH), May 2013, pp. 1-8, 2013.

[8] N. Hu et al., “Experimental validation of a contactless finger displacement measurement system using electrical near-field sensing”, IEEE Transactions on Instrumentation and Measurement, vol. 70, pp. 1-9, 2021.

[9] L. Sucar et al., “Gesture therapy: a vision-based system for upper extremity stroke rehabilitation,” in Proc. IEEE 32nd International Engineering in Medicine and Biology Conference pp. 3690–3693, Buenos Aires, Argentina, 2010.

[10] J. Zariffa and J. Steeves, “Computer vision-based classification of hand grip variations in neurorehabilitation,” in Proc. IEEE International Conference on Rehabilitation Robotics, pp. 1–4, Zurich, Switzerland, June 2011.

[11] T. D. White et al., “Modelling hand gestures to test Leap Motion controlled applications,” 2018 IEEE International Conference on Software Testing, Verification and Validation Workshops, pp. 204-213,Vasteras, Sweden, April 2018.

[12] C. Cifuentes et al., “Development of a wearable zigbee sensor system for upper limb rehabilitation robotics,” IEEE RAS EMBS International Conference on Biomedical Robotics and Biomechatronics, pp. 1989-1994, 2012.

[13] S. Salim et al., “Integration of tilt sensors as a device for monitoring rehabilitation process,” IEEE International Conference on Control System, Computing and Engineering, pp. 232-235, Batu Ferringhi, Malaysia, Nov. 2014.

[14] J. Guna et al., “An analysis of the precision and reliability of the leap motion sensor and its suitability for static and dynamic tracking,” IEEE Sensors, vol. 14, no. 2, pp. 3702–3720, Feb. 2014.

[15] L.B. Amor et al., “PCA-Based multivariate anomaly detection in mobile healthcare applications,” IEEE International Symposium on Distributed Simulation and Real Time Applications, pp. 172-179, Rome, Italy, 2017.

[16] N.F. Troje, “Decomposing biological motion: a framework for analysis and synthesis of human gait pattern,” Journal of Vision, vol. 2, pp. 371-387, 2002.

[17] M. Lee et al., “An application of principal component analysis for lower body kinematics loaded and unloaded walking,” Journal of Biomechanics, vol. 42, no. 14, pp. 2226-2230, 2009.

[18] M. K. Ahuja and A. Sing, “Static vision based hand gesture recognition using Principal Component Analysis,” IEEE International Conference on MOOCs, Innovation and Technology in Education (MITE), pp.402 - 406 ,Punjab, India, 2015.

[19] A. Mannini and A. Sabatini, “Machine learning methods for classifying human physical activity from on-body accelerometers,” Sensors, vol. 10, no. 2, pp. 1154-1175, Feb. 2010.

[20] A. Uribe-Hurtado et al., “Dynamic time warping dissimilarity matrices,” IEEE Journals and Magazines, vol. 37, no. 5, pp. 34-42, Oct. 2018.

[21] P. Tormene et al., “Matching incomplete time-series with dynamic time warping: an algorithm and an application to post-stroke rehabilitation,” Artificial Intelligence in Medicine, vol. 45, no. 1, pp. 11-34, Jan. 2009.

[22] K. Mendhurwar et al., “The discriminative power of shape an empirical study in time series matching,” IEEE Transactions on Visualization and Computer Graphics, vol. 24, no. 5, pp. 1799-1813, May 2018.

[23] F.A. Dwiputra et al., “Accelerometer-based recorder of fingers dynamic movements for post-stroke rehabilitation,” International Journal on Advanced Science Engineering Information Technology, vol. 7, no. 1, pp. 299-304, 2017.

[24] D. Biswas et al., “Recognition of elementary arm movements using orientation of a tri-axial accelerometer located near the wrist,” Physiological Measurement, vol. 35, no. 9, 1751-68, Aug. 2014.

[25] D. Wade and B. Jong, “Recent advances in rehabilitation,” Biomedical Journal, vol. 320, pp. 1385-1388, 2000.

[26] M. Ghassemi, J. M. Ochoa, N. Yuan, D. Tsoupikova, and D. Kamper, "Development of an integrated actuated hand orthosis and virtual reality system for home-based rehabilitation," 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, pp. 1689-1692, 2018.

[27] Y. Xue, Z. Ju, K. Xiang, J. Chen, and H. Liu, "Multimodal human hand motion sensing and analysis-A review," IEEE Transactions on Cognitive and Developmental Systems, vol. 11, no. 2, pp. 162-175, June 2019.

[28] J. Nassour et al., “A robust data-driven soft sensory glove for human hand motions identifications and replications,” IEEE Sensors Journal, vol. 20, no. 21, pp. 12972-12979, Nov. 2020.

[29] A. Phinyomark, E. Campbell, and E. Scheme, "Surface electromyography (EMG) signal processing, classification, and practical considerations, Biomedical Signal Processing, pp. 3-29, Nov. 2019.

[30] A. A. Tarar, U. Mohammad, and S. K. Srivastava, "Wearable skin sensors and their challenges: A review of transdermal, optical, and mechanical sensors," Biosensors, vol. 10, no. 56, pp. 1-22, May 2020.

[31] C. Mizera et al., "Evaluation of hand-tracking systems in teleoperation and virtual dexterous manipulation," IEEE Sensors Journal, vol. 20, no.3, pp. 1642-1655, Feb. 2020.

[32] H. Walugembe, C. Phillips, J. Requena-Carrion, and T. Timotijevic, “Comparing dynamic hand rehabilitation gestures in Leap Motion using multi-dimensional dynamic time warping,” IEEE Sensors Journal, Vol. 21, No. 6, pp. 8002-8010, March 2021.

[33] W. Wettayaprasit, N. Laosen, and S. Chevakidagarn, "Data filtering technique for neural networks forecasting", Proceedings of the 7th WSEAS International Conference on Simulation, Modelling and Optimization, pp. 225-230, Sept. 2007.

[34] H. Walugembe et al., “Characterizing and compensating for errors in a Leap Motion using PCA, ” In proceedings of IEEE International Conference on Signal Processing and Information Security (ICSPIS), pp. 1-4, Nov. 2018.

[35] G. James et al., “An introduction to statistical learning with applications in R,” Department of Information and Operations management, University of Southern California Los Angeles, CA, USA, 2013.

[36] Z. Jiang, “Support vector machines for multi-class pattern recognition based on improved voting strategy,” In proceedings of IEEE Chinese Control and Decision Conference, pp. 517-520, May 2010.

[37] M. Lu and H. Lin, “Parallel algorithms for the longest common subsequence problem,” IEEE Transactions on Parallel and Distributed Systems, vol. 5, no. 8, pp. 835-848, Aug. 1994.

[38] J. Herranz et al., “Optimal symbol alignment distance: a new distance for sequences of symbols,” IEEE Transactions on Knowledge and Data Engineering, vol. 23, no. 10, pp. 1541-1554, Oct. 2011.

[39] P. Marteau, “Time warp edit distance with stiffness adjustment for time series matching,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 31, no. 2,

pp. 306-318, Feb. 2009.

[40] H. Sveistrup, "Motor rehabilitation using virtual reality," Journal of NeuroEngineering and Rehabilitation, vol. 1, pp. 1-8, Dec. 2004.

[41] R.V. Kenyon et al., "Considerations for the future development of virtual technology as a rehabilitation tool," Journal of NeuroEngineering and Rehabilitation, vol. 1, pp. 1-10, Dec. 2004.

[42] M. Agius, C. Seguna, J. Attard, K. Scicluna and J. Scerri, "A wearable wireless sensing system for capturing human arm motion," IEEE Latin America Symposium on Circuits and System (LASCAS), pp. 1-4, 2021.

[43] Q. Fu, J. Fu, J. Guo, S. Guo, and X. Li, "Gesture recognition based on BP neural network and data glove," IEEE International Conference on Mechatronics and Automation, pp. 1918-1922, Oct. 2020.

[44] J. Nassour et al., "A robust data-driven soft sensory glove for human hand motions identification and replication," IEEE Sensors Journal, pp. 12972-12979, vol. 20, no. 21, Nov. 2020.

[45] X. Chen et al., "A wearable hand rehabilitation system with soft gloves," IEEE Transactions on Industrial informatics, pp. 943-952, vol. 17, no. 2, Feb. 2021.

[46] Y. Kang, Z. Li, X. Cao, and D. Zhai, "Robust control of motion/force for robotic manipulators with random time delays," IEEE Transactions on control systems technology, vol. 21, no. 5, pp. 1708-1718, Sept. 2013.

[47] C. J. D. Luca, "The use of surface electromyograph in biomechanics, pp. 1-29, 1997.

[48] A. H. Timemy, G. Bugmann, J. Escudero, and N. Outram, "Classification of finger movements for the dexterous hand prosthesis control with surface electromyograph," IEEE Journal of Biomedical and Health Informatics, vol. 17, no. 3, May 2013.

[49] Y. Xue, X. Ji, D. Zhou, J. Li and Z. Ju, "SEMG-based human In-hand motion recognition using nonlinear time series analysis and random forest," IEEE Access, vol. 7, pp. 176448-176457, 2019.

[50] S. Alexanderson, "Robust online motion capture labeling of finger markers," Proceedings of the 9th International Conference on Motion in Games, pp. 7-13, Burlingame, California October 2016.

[51] S. Alexanderson and J. Beskow, "Towards fully automated motion capture of signs development and evaluation of a key word signing avatar," ACM Transactions on Accessible Computing (TACCESS), Issue 2, vol. 7, pp. 1-17, July 2015.

[52] G. M. Lim, P. Jatesiktat, C. W. Keong Kuah, and W. Tech Ang, "Camera-based hand tracking using a mirror-based multi-view setup," 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC), pp. 5789-5793, Aug. 2020.

[53] C. Metcalf et al., "Markerless motion capture and measurement of hand kinematics: validation and application to home-based upper limb rehabilitation," IEEE Transactions on Biomedical Engineering, vol. 60, no. 8, pp. 2184-2192, August 2013.

[54] S. Sridhar, "FullHand: markerless skeleton-based tracking for free-hand interaction," (MPI-I-2016-4-002), 2016.

[55] C. Keskin, "Real time hand pose estimation using Depth sensors," In 2011 IEEE International Conference on Computer Vision, pp. 1228 –1234, Barcelona Spain November 2011.

[56] G. Marin et al., "Hand gesture recognition with leap motion and Kinect," IEEE International Conference on Image Processing (ICIP), pp. 1565-1569, Paris France, 2014.

[57] L. Shao, “Hand movement and gesture recognition using Leap Motion controller, Stanford EE 267, Virtual Reality Course Report pp. 1-5, 2016.

[58] American society for surgery of the hand, http://www.assh.org/handcare/Anatomy/Joints, accessed 21/09/2018.

[59] H. Walugembe et al., "Gesture recognition in Leap Motion using LDA and SVM," IEEE International Conference on Computing, Electronics & Communications Engineering (iCCECE), pp. 56-61, London UK, August 2019.

[60] G. Ponraj and H. Ren, “Sensor fusion of Leap Motion controller and flex sensors using

Kalman filter for human finger tracking ," IEEE Sensor Journal, vol. 18, no. 5, pp. 2042-2049, Jan. 2018.

[61] K. Li et al., "Development of finger-motion capturing device based on optical linear encoder," Journal of Rehabilitation Research and Development, 2011, vol. 48, no. 1, pp. 69-82, 2011.

[62] Y. Wu and T. Huang, "Human hand modeling, analysis and animation in the context of HCI," Proceedings of the International Conference on Image Processing, October 1999, Kobe Japan, pp. 6-10, 1999.

[63] T. Rhee et al., "Human hand modeling from surface anatomy," Proceedings of the 2006 Symposium on Interactive 3D Graphics and Games, March 2006, Redwood City California, pp. 27-34, 2006.

[64] S. Cobos et al., "Efficient human hand kinematics for manipulation tasks," IEEE/RSJ International Conference on Intelligent Robots and Systems, Nice France, pp. 2246-2251, Sept. 2008.

[65] F. Chen et al., "Constraint study for a hand exoskeleton: human hand kinematics and dynamics," Journal of Robotics, vol. 2013, pp. 1-18, July 2013.

[66] T. Guzsvinecz, V. Szucs, and C. Sik-Lanyi, "Suitability of the Kinect Sensor and Leap Motion controller-A literature review," Sensors, Vol. 19, no. 5, pp. 1-25, March 2019.

[67] Leap Motion, [Online]. Available: https://www.leapmotion.com/ accessed 19/03/2019.

[68] W. Lu et al., "Dynamic hand gesture recognition with leap motion controller," IEEE Signal Processing Letters, vol. 23, no. 9, pp. 1188-1192, Sept. 2016.

[69] K. M. Ribeiro, R. A. Braga Jr, G. W. Horgan, D. D. Ferreira, and T. Safadi, "Principal component analysis in the spectral analysis of the dynamic laser speckle patterns," vol. 9, Journal of the European Optical Society-Rapid publications 9, 2014.

[70] H. Abdi, and L. J. Williams, "Principal Component Analysis," Wiley Interdisciplinary Reviews, Computational Statistics vol. 2, no. 4, pp. 433-459, 2010.

[71] L. Zhang, W. Dong, D. Zhang, and G. Shi, "Two-stage image denoising by principal

component analysis with local pixel grouping," Pattern Recognition, vol. 43, pp. 1531-1549, 2010.

[72] S. Jung, A. Sen, and J. S. Marron, "Boundary behaviour in high dimension, low sample size asymptotics of PCA," Journal of Multivariate Analysis, vol. 109, pp. 190-203, Aug. 2012.

[73] J. D. Hadfield, "MCMC methods for multi-response generalized linear models: The MCMCglmm R package," Journal of Statistical Software, vol. 33, no. 2, pp. 1-22, 2010.

[74] L. Batina, J. Hogenboom, and J. G. J. Van Woudenberg, "Getting more from PCA: First results of using principal component analysis for extensive power analysis," Proceedings of the Cryptographers' Track at the RSA Conference, pp. 383-397, March 2012.

[75] J. Qi, G. Jiang, G. Li, Y. Sun, and B. Tao, "Surface EMG hand gesture recognition system based on PCA and GRNN, Neural Computing and Applications, vol. 32, pp. 6343-6351, 2020.

[76] M. Welling, “Fisher linear discriminant analysis,” Dept. Comp. Sci., Univ. Toronto, Toronto, ON, Canada, vol. 3, 2005.

[77] G. Tharwat et al, “Linear discriminant analysis: a detailed tutorial,” University of Salford, Manchester, pp. 1-23, 2017.

[78] H. Yu and J. Yang, “A direct LDA algorithm for high-dimensional data with application to face recognition,” Pattern Recognition, vol. 34, no. 10, pp. 2067-2070, 2001.

[79] J. Lu et al., “Face recognition using LDA-based algorithm,” IEEE Transactions on Neural Networks, vol. 14, no. 1, pp. 195-200, 2003.

[80] J. Ye et al., “Two-dimensional linear discriminant analysis,” In Proceedings of 17th Advances in Neural Information Processing Systems (NIPS), pp. 1569–1576, 2004.

[81] G. Strang, “Introduction to linear algebra,” Wellesley-Cambridge Press, Massachusetts, Fourth Edition, 2003.

[82] T. Li et al., “Using discriminant analysis for multi-class classification: an experimental investigation,” Knowledge and information systems, vol. 10, no. 4, pp. 453–472, 2006.

[83] Y. Wang, and L. Zhang, "3D hand gesture recognition based on polar rotation feature

and linear discriminant analysis," International Conference on Intelligent Control and Information Processing (ICICP), pp. 215-219, 2013.

[84] H. M. S. P. B. Herath, M. P. B. Ekanayake, G. M. R. I. Godaliyadda and J. V. Wijayakulasooriya, "Multi-feature based hand-gesture recognition," 2015 Fifteenth International Conference on Advances in ICT for Emerging Regions (ICTer), pp. 63-68, 2015.

[85] K. Hu, L. Yin and T. Wang, "Temporal Interframe Pattern Analysis for Static and Dynamic Hand Gesture Recognition," 2019 IEEE International Conference on Image Processing (ICIP), pp. 3422-3426, 2019.

[86] G. Luo, P. Yang, M. Chen and P. Li, "HCI on the Table: Robust Gesture Recognition Using Acoustic Sensing in Your Hand," in IEEE Access, vol. 8, pp. 31481-31498, 2020.

[87] C. Shen, Y. Chen, G. Yang and X. Guan, "Toward Hand-Dominated Activity Recognition Systems With Wristband-Interaction Behaviour Analysis," in IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 50, no. 7, pp. 2501-2511, July 2020.

[88] H. Zeng, K. Li, X. Tian, N. Wei, R. Song and L. Zhou, "Classification of hand motions using linear discriminant analysis and support vector machine," 2017 Chinese Automation Congress (CAC), pp. 2353-2356, 2017.

[89] A. Adenike, J. Levi, and A. Todd, "An analysis of intrinsic and extrinsic hand muscle EMG for improved pattern recognition control, IEEE Transactions on Neural Systems and Rehabilitation Engineering, vol. 24, pp. 485-494, 2016.

[90] R. N. Khushaba, A. Al-Timemy, S. Kodagoda, and K. Nazarpour, "Combined influence of forearm orientation and muscular contraction on EMG on EMG pattern recognition," Journal of Expert Systems with Applications, vol. 61, pp. 154-161, 2016.

[91] M. Liu, Y. Lin, Z. Qiu, C. Kuo and C. Wu, "Hand Gesture Recognition by a MMG-Based Wearable Device," IEEE Sensors Journal, vol. 20, no. 24, pp. 14703-14712, Dec.15, 2020.

[92] Y. Zhu, S. Jiang and P. B. Shull, "Wrist-worn hand gesture recognition based on barometric pressure sensing," 2018 IEEE 15th International Conference on Wearable and Implantable Body Sensor Networks (BSN), pp. 181-184, 2018.

[93] P. Kaczmarek, T. Mankowski and J. Tomczyiski, "Towards sensor position-invariant hand gesture recognition using a mechanomyographic interface," 2017 Signal Processing: Algorithms, Architectures, Arrangements, and Applications (SPA), pp. 53-58, 2017.

[94] H. Ding, Q. He, Y. Zhou, G. Dan, and S. Cui, “An individual finger gesture recognition system based on motion-intent analysis using mechanomyogram signal,” Frontiers in Neurology., vol. 8, pp. 1-11, Nov. 2017.

[95] R. Booth and P. Goldsmith, “A wrist-worn piezoelectric sensor array for gesture input,” Journal of Medical and Biological Engineering, vol. 38, no. 2, pp. 284–295, Apr. 2018.

[96] F. Duan, X. Ren, and Y. Yang, "A gesture recognition system based on time domain features and linear discriminant analysis," IEEE Transactions on Cognitive and Developmental Systems, vol. 13, no. 1, pp. 200-208, March 2021.

[97] J. Yousefi and A. Hamilton-Wright, “Characterizing EMG data using machine-learning tools,” Computer in Biology and Medicine, vol. 51, pp. 1–13, Aug. 2014.

[98] X. Zhang, X. Chen, Y. Li, V. Lantz, K. Wang, and J. Yang, “A framework for hand gesture recognition based on accelerometer and EMG sensors,” IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans, vol. 41, no. 6, pp. 1064–1076, Nov. 2011.

[99] A. Alkan and M. Gunay, “Identification of EMG signals using discriminant analysis and SVM classifier,” Expert Systems with Applications, vol. 39, no. 1, pp. 44–47, Jan. 2012.

[100] K. Xing, P. Yang, J. Huang, Y. Wang, and Q. Zhu, “A real-time EMG pattern recognition method for virtual myoelectric hand control,” Neurocomputing, vol. 136, pp. 345–355, Jul. 2014.

[101] F. Duan et al., “sEMG-based identification of hand motion commands using wavelet neural network combined with discrete wavelet transform,” IEEE Transactions on Industrial Electronics, vol. 63, no. 3, pp. 1923–1934, Mar. 2016.

[102] D. Martens et al., “Decompositional rule extraction from support vector machines by active learning,” IEEE Transactions on Knowledge and Data Engineering, vol. 21, no. 2, pp. 178-191, Feb. 2009.

[103] V.N. Vapnik, “The nature of statistical learning theory,” Springer-Verlag Inc., 1995.

[104] S. Abe, “Support Vector Machine for Pattern Classification,” vol. 53, New York, USA: Springer- Verlag, 2005.

[105] N. Cristianini and J. Shawe-Taylor, “An introduction to support vector machines and other kernel-based learning methods,” Cambridge University Press, 2000.

[106] D.K. Ghosh and S. Ari, “Static hand gesture recognition using mixture of features and SVM classifier,” IEEE International Conference on Communication Systems and Network Technologies, pp. 1094-1099, April 2015.

[107] H. Dardas and D. Georganas, “Real-time hand gesture detection and recognition using bag-of-features and support vector machine techniques,” IEEE Transactions on instrumentation and measurement, vol. 60, no. 11, Nov. 2011.

[108] J. Sahoo et al., “Hand gesture recognition using PCA based deep CNN reduced features and SVM classifier,” IEEE International symposium on smart electronic systems, pp. 221-224, 2019.

[109] W. K. Wong, F. H. Juwone, and B. T. T. Khoo, "Multi-features capacitive hand gesture recognition sensor: a machine learning approach," IEEE Sensors Journal, vol. 21, no. 6, pp. 8441-8450, 2021.

[110] X. Zhang, Z. Yang, T. Chen, D. Chen, and M. C. Huang, “Cooperative sensing and wearable computing for sequential hand gesture recognition,” IEEE Sensors Journal, vol. 19, no. 14, pp. 5775–5783, Jul. 2019.

[111] F. Chen et al., “WristCam: A wearable sensor for hand trajectory gesture recognition and intelligent human–robot interaction,” IEEE Sensors Journal, vol. 19, no. 19, pp. 8441 - 8451, Oct. 2019.

[112] K. S. Abhishek, L. C. F. Qubeley, and D. Ho, “Glove-based hand gesture recognition sign language translator using capacitive touch sensor,” IEEE International Conference on Electron Devices and Solid-State Circuits (EDSSC), pp. 334–337, Aug. 2016.

[113] V. Nguyen, S. Rupavatharam, L. Liu, R. Howard, and M. Gruteser, “HandSense:

Capacitive coupling-based dynamic, micro finger gesture recognition," Proceedings of the 17th Conference on Embedded Networked Sensor Systems, pp. 285-297, 2019.

[114] M. Hirsch, J. Cheng, A. Reiss, M. Sundholm, P. Lukowicz, and O. Amft, "Hands-free gesture control with a capacitive textile neckband," Proceedings of the ACM International Symposium on Wearable Computers, pp. 55-58, 2014.

[115] A. Devaraj and A. K. Nair, "Hand gesture signal classification using machine learning," IEEE International Conference on Communication and Signal Processing (ICCSP), pp. 390-394, 2020.

[116] P. Senin, "Dynamic time warping algorithm review," technical report. http://seninp.github.io/assets/pubs/senin_dtw_litreview_2008.pdf , Accessed on 16/10/2018.

[117] C. Andrea, "Dynamic time warping for offline recognition of a small gesture vocabulary," Proceedings of the IEEE ICCV Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems, August, 2001.

[118] D. Gavrila and L. Davis, "Towards 3-d model-based tracking and recognition of human movement: multi-view approach" IEEE International Workshop on Automatic Face and Recognition, IEEE Computer Society, Zurich, Switzerland, pp. 272-277, 1995.

[119] K. Eamonn and M. Pazzani, "Derivative dynamic time warping," First international SIAM International Conference on Data Mining, Chicago, IL USA, 2001.

[120] S. Stan and C. Philip, "FastDTW: toward accurate dynamic time warping in linear time and space," KDD Workshop on Mining Temporal and Sequential Data, 2004.

[121] E. Keogh and C. Ratanamahatana, "Exact indexing of dynamic time warping," Knowledge and Information Systems, vol. 7, Issue 3, pp. 358-386, 2005.

[122] T. Raihan, "Predicting US recessions: a dynamic time warping exercise in economics," University of California, Riverside (UCR), College of Humanities, Department of Economics, pp. 1-41, Oct 2017.

[123] H. Hiyadi et al., "Adaptive dynamic time warping for recognition of natural gestures,"

IEEE Sixth International Conference on Image Processing Theory, Tools and Applications, pp. 1-6, 2016.

[124] S. Elsworth, “Tutorial on dynamic time warping,” School of Mathematics, University of Manchester, Sept. 2017.

[125] H. Sakoe and S. Chiba, “Dynamic programming algorithm optimization for spoken word recognition,” IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 26, no. 1, Feb. 1978.

[126] M. Ibrahim and D. Mulvaney, “Geometrical-based lip-reading using template probabilistic multi-dimensional dynamic time warping,” Journal of Visual Communication and Image Representation, vol. 30, pp. 219-233, July 2015.

[127] G. Plouffe and A. Cretu, "Static and dynamic hand gesture recognition in depth data using dynamic time warping," IEEE Transactions on Instrumentation and Measurement, vol. 65, no. 2, pp. 305-316, Feb. 2016.

[128] Y. Li et al., "A Dynamic hand gesture recognition model based on the improved dynamic time warping algorithm," IEEE International Conference on Automation and Computing (ICAC), 2019, pp. 1-6, 2019.

[129] J. J. David, J. C. L. Genavia, T. L. Laplana, L. F. O. Rodrigo, D. A. Rodriguez and R. E. Tolentino, "Hand gesture recognition model using standard deviation-based dynamic time warping technique," IEEE International Conference on Computing Methodologies and Communication (ICCMC), 2021, pp. 1043-1050, 2021.

[130] G. Al-Naymat, S. Chawla, and J. Taheri, "SpareDTW: A novel approach to speed up dynamic time warping," in Proceedings of Australasian Data Mining Conference, vol. 101, pp. 117-127, 2012.

[131] Y. Jeong et al., “Weighted dynamic time warping for time series classification,” Pattern Recognition, vol. 44, pp. 2231-2240, 2011.

[132] M. E. Munich and P. Perona, “Continuous dynamic time warping for translation-invariant curve alignment with applications to signature verification,” In proceedings of the

7th IEEE International Conference on Computer Vision, pp. 1-8, Aug. 2002.

[133] M. Parizeau and R. Plamondon, “A comparative analysis of regional correlation, dynamical time warping and skeletal tree matching for signature verification,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 12, no. 7, pp. 710-717, July 1990.

[134] B. Serra and M. Berthod, “Subpixel contour matching using continuous dynamic programming,” In proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 202-207, June 1994.

[135] D. Liu et al., “Downsizing without downgrading: Approximated dynamic time warping on non-volatile memories,” IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 39, no. 1, pp. 131-144, Jan. 2020.

[136] X. Li et al., “Downsampling of time-series data for approximated dynamic time warping on non-volatile memories,” In proceedings of 6th Non-volatile Memory Systems and Applications Symposium (NVMSA), pp. 1-6, Aug. 2017.

[137] S. Salvador and P. Chan, “Toward accurate dynamic time warping in linear time and space,” Intelligent Data Analysis, vol. 11, no. 5, pp. 561-580, Oct. 2007.

[138] L. Rabiner and C. Schmidt, “Application of dynamic time warping to connected digit recognition,” IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 28, no. 4, pp. 377-388, Aug. 1980.

[139] C. Myers and L. Rabiner, “A level building dynamic time warping algorithm for connected word recognition,” IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 29, no. 2, pp. 284-297, April 1981.

[140] X. Ruan, and C. Tian, "Dynamic gesture recognition based on improved DTW algorithm," IEEE International Conference on Mechatronics and Automation, pp. 2134-2138, Aug. 2015.

[141] H. Cheng, Z. Dai, and Z. Liu, "Image-to-Class dynamic time warping for 3D hand gesture recognition," IEEE International Conference on Multimedia and Expo (ICME), pp. 1-6, 2013.

[142] J. J. David et al., "Hand gesture recognition model using standard deviation-based dynamic time warping," International Conference on Computing Methodologies and Communication (ICCMC), pp. 1043-1050, 2021.

[143] W. Wettayaprasit, N. Laosen, and S. Chevakidagarn, "Data filtering technique for neural networks forecasting", Proceedings of the 7th WSEAS International Conference on Simulation, Modelling and Optimization, pp. 225-230, Sept. 2007.

[144] S. Rautaray and A. Agrawal, “Interaction with virtual game through hand gesture recognition,” In proceedings of IEEE International Conference on Multimedia, Signal Processing and Communications Technologies, pp. 244-247, Dec. 2011.

[145] Z. Ren et al., “Robust part-based hand gesture recognition using Kinect sensor,” IEEE Transactions on Multimedia, vol. 15, no. 5, pp. 1110-1120, Aug. 2013.

[146] A. Truong et al., “Laban descriptors for gesture recognition and emotional analysis,” The Visual Computer, vol. 32, no. 1, pp. 83-98, Jan. 2015.

[147] E. Ohn-Bar and M. Trivedi, “Hand gesture recognition in real time for automatic interfaces: A multimodal vision-based approach and evaluations,” IEEE Transactions on Intelligent Transportation Systems, vol. 15, no. 6, pp. 2368-2377, Dec. 2014.

[148] J. Palacios et al., “Human-computer interaction based on hand gestures using RGB-D sensors,” Sensors, vol. 13, no. 9, pp. 11842-11860, Sept. 2013.

[149] Y. Liu, Y. Zhang, and M. Zeng, "Novel algorithm for hand gesture recognition utilizing a wrist-worn inertial sensor," IEEE Sensors Journal, vol. 18, no. 24, pp. 10085-10095, Dec 2018.

[150] Y. Wang and H. Ma, "Real-time continuous gesture recognition with wireless wearable IMU sensors," IEEE International Conference on e-Health Networking, Applications and Services (Healthcom), pp. 1-6, 2018.

[151] H. Choi and T. Kim, "Directional dynamic time warping for gesture recognition," Proceedings of the International Conference on Multimedia Systems and Signal Processing, pp. 22-25, Aug. 2017.

[152] Salford Royal NHS Foundation, "Rheumatology hand exercises," 2019.

[153] Milton Keynes University NHS Hospital, "Rheumatology Hand and Wrist Exercise Programme," [Online]. Available: https://www.mkuh.nhs.uk/patient-information leaflet/rheumatology-hand-and-wrist-exercise-programme accessed 17/06/2021

[154] Derbyshire Community NHS Health Services, "Hand and Finger Exercises, April 2018.

[155] J. Jia, G. Tu, X. Deng, C. Zhao, and W. Yi, "Real-time hand gestures system based on leap motion," Wiley Online Library Journal, pp. 1-11, Nov. 2018.

[156] L. Huang and T. Sun, "The HMM-based sensing correction method for leap motion finger tracking," IEEE International Conference on Innovation, Communication and Engineering (ICICE), pp. 78-81, 2019.

[157] H. Jin et al., “Multi-leap motion sensor based demonstration for robotic refine tabletop object manipulation task,” CAAI Transactions on Intelligence Technology, vol. 1, no. 1, pp. 104-113, Jan. 2016.