South African Sign Language Dataset Development and Translation: A Glove-based Approach

Prepared by:

Ben McInnes

MCNBEN003

Department of Electrical Engineering
University of Cape Town

Prepared for:

Mohohlo Tsoeu

Electrical Engineering
University of Cape Town

August 2014

Submitted to the Department of Electrical Engineering at the University of Cape Town in partial fulfilment of the academic requirements for a Master of Science degree in Electrical Engineering

Declaration

1. I know that plagiarism is wrong. Plagiarism is to use another's work and pretend that it is one's own.
2. I have used the IEEE convention for citation and referencing. Each contribution to, and quotation in, this final year project report from the work(s) of other people, has been attributed and has been cited and referenced.
3. This Research Project is my own work.
4. I have not allowed, and will not allow, anyone to copy my work with the intention of passing it off as their own work or part thereof

Name: **Ben McInnes**

Signature: ____________________ **Date:** 13 November 2014

Terms of Reference

Title

"South African Sign Language Dataset Development and Translation: A Glove-based Approach"

Description

There is a definite breakdown in communication between the hearing and Deaf communities. Researchers have turned to technology to remedy this problem using Automatic Sign Language Translation. This is not possible in South Africa as there currently is no South African Sign Language database available.

The goal of this research is to develop a pair of low cost data gloves in order to develop a basic SASL database. The research is limited to creating a static gesture database and using available software in order to classify the gestures. The data glove system must be able to be extended to record dynamic gestures for future research and the further development of the SASL database.

Deliverables

Objectives

The main objectives of this research are:

i. Understand the requirements of the project
ii. Conduct a literature review of previous work and critically evaluate current technology/research
iii. Develop and evaluate a pair of data gloves with the purpose of creating a SASL database
iv. Create a static gesture database
v. Using machine learning to validate the database
vi. Discuss the performance of the data gloves, database and machine algorithms and make conclusions
vii. Make recommendations for future work

Outcomes

i. A dissertation document that reports the findings of the research
ii. A pair of fully functioning data gloves (left and right)
iii. An IEEE standard conference paper

Skills/Requirements

Strong mechatronic, electronic and programming skills.

Acknowledgements

First and foremost I would like to thank my parents, Janet and Phil McInnes. Their unwavering support and faith in my abilities kept me going during the most trying and stressful times during my Masters. Words cannot express my gratitude. I would also like to thank my brother Iain, for being interested, supportive and always willing to listen despite all of my Master's related rants.

Next I would like to thank my girlfriend, Adoné Kitching. She has been in the Master's trenches with me, staying positive and giving amazing support during the uncountable work weekends, while friends were out having fun. The post-masters activities list grew with each work weekend, giving us something to look forward to and keep our sanity in check.

I would like to thank all my friends for their support and for not giving up on me when I became a Master's hermit. They had to listen to all of the many tirades about the problems with my research and for being good fun when I could pull myself away from my research.

I would like to thank the other students in the UCT Control Lab, a constant source of information when stuck and procrastination when work was just not going to be done that day.

I would like to thank my supervisor Mohohlo Tsoeu for his guidance, knowledge and all of the feedback he has given me over the last 2 years.

I would like to thank Robyn Verrinder for being an amazing "unofficial supervisor". I really appreciate and am grateful for the amazing advice as well as all the time that she gave to me over the course of my Master's, despite the fact that I was one of her students.

Abstract

There has been a definite breakdown of communication between the hearing and the Deaf communities. This communication gap drastically effects many facets of a Deaf person's life, including education, job opportunities and quality of life. Researchers have turned to technology in order to remedy this issue using Automatic Sign Language. While there has been successful research around the world, this is not possible in South Africa as there is no South African Sign Language (SASL) database available. This research aims to develop a SASL static gesture database using a data glove as the first step towards developing a comprehensive database that encapsulates the entire language. Unfortunately commercial data gloves are expensive and so as part of this research, a low-cost data glove will be developed for the application of Automatic Sign Language Translation. The database and data glove will be used together with Neural Networks to perform gesture classification. This will be done in order to evaluate the gesture data collected for the database.

This research project has been broken down into three main sections; data glove development, database creation and gesture classification. The data glove was developed by critically reviewing the relevant literature, testing the sensors and then evaluating the overall glove for repeatability and reliability. The final data glove prototype was constructed and five participants were used to collect 31 different static gestures in three different scenarios, which range from isolated gesture collection to continuous data collection. This data was cleaned and used to train a neural network for the purpose of classification. Several training algorithms were chosen and compared to see which attained the highest classification accuracy.

The data glove performed well and achieved results superior to some research and on par with other researchers' results. The data glove achieved a repeatable angle range of 3.27 degrees resolution with a standard deviation of 1.418 degrees. This result is far below the specified 15 degrees resolution required for the research. The device remained low-cost and was more than $100 cheaper than other custom research data gloves and hundreds of dollars cheaper than commercial data gloves.
A database was created using five participants and 1550 type 1 gestures, 465 type 2 gestures and 93 type 3 gestures were collected.

The Resilient Back-Propagation and Levenberg-Marquardt training algorithms were considered as the training algorithms for the neural network. The Levenberg-Marquardt algorithm had a superior classification accuracy achieving 99.61%, 77.42% and 81.72% accuracy on the type 1, type 2 and type 3 data respectively.

Table of Contents

List of Figures

List of Tables

1. Introduction

1.1 Background to the study

According to the South African 2011 Census [1] there are over 51 770 people that are completely deaf and 258 800 people with severe hearing loss. It has been recorded that roughly 258 800 people use South African Sign Language (SASL) in their homes.

There has always been a definite breakdown of communication between the hearing and Deaf communities, so much so that the Deaf community has developed its own culture and identity. South African Sign Language, which was heavily influenced by German and American Sign Languages [2], is a language that is completely made up of hand, body and face gestures as well as unique grammatical structures to communicate information. No sound is required for the transfer of thoughts or information.

According to the census only 0.5% of the population speak South African Sign Language. While South African Sign Language is not one of the Official South African languages, it has been formally recognised by the government. “Sign Language has the status of an official language for the purposes of learning at public school” (South African School Act 1996).

Researchers have turned to technology in order to bridge this gap in communication. The research is known as Automatic Sign Language Translation which has been broken down into two major sections; recording user gesture data as a dataset and classifying the gestures for translation using machine learning algorithms.

There are two types of data when trying to record sign language, the first being static gestures like finger spelling and the second being dynamic gesture such as those used for entire words or phrases. The two most common methods of capturing SASL user data are: data gloves or visual systems. Data gloves use sensors that are directly attached to the SASL users’ hands in order to capture and record their gestures whereas the visual approach will use an RGB or 3D camera. Once the data has been captured and cleaned, it is used for classification using machine learning algorithms. The Cyber Glove [3] and the 5DT Glove [4] are common commercial data gloves that have been used for this purpose.

Machine learning is a branch of artificial intelligence that is used in order to develop and create systems that can learn from data. For the purpose of automatic sign language translation the two most common machine learning algorithms have been that of neural networks for static gestures and Hidden Markov Models for dynamic gestures. The success of research such as automatic sign language could have far reaching positive outcomes for both hearing and Deaf communities.

1.2 Objectives of this study

1.2.1 Problems to be investigated

While there has been a large amount of research into automatic sign language translation around the world, there has been little to none done on South African Sign Language. This is due to the lack of available digital SASL databases. This research aims to create a platform in order for SASL related ASLT to be possible. This will be done by developing a device to create a SASL database. This research will focus on the development of the data glove device and that of developing a static gesture database. This database will then be able to be expanded upon by other researchers in the future.

A major issue is that the University of Cape Town does not have its own device to develop the database and while commercial data gloves are available, they are very expensive. This research aims to solve this problem by developing a low cost data glove for the purpose of creating the SASL database. To develop the data glove different sensors must be tested and evaluated in order to determine the best fit for the device. This device will also be used to classify and translate SASL. While the focus of this research is to develop a static gesture database the data glove must be able to handle the recording and classifying of dynamic gestures which will allow it to be useful for potential future research.

This research has been broken up into three main facets:

i. Develop and evaluate data glove
ii. Create SASL static gesture database
iii. Validate the database using machine learning algorithms

1.2.2 Purpose of the study

The focus of this study is to take the first steps towards creating a full SASL database with the purpose of closing the communication gap between hearing and Deaf communities. This research aims to pave the way for ASLT research in a South African context by developing the necessary tools to make it possible.

A data glove will be developed in order to give future researchers a tool to expand the SASL database. Once the database has been expanded from static to dynamic gestures this will give researchers the resources they need in order to work on ASLT for SASL which will lead to improvements in potentially hundreds of thousands of lives.

1.3 Design Specifications

The research must adhere to the following specifications:

1.3.1 Glove

i. Functional Characteristics

The device must be compact and wearable. It should not hinder or restrict the movements of the user in any manner. The user should be able to perform day to day tasks while wearing the device. The device must be able to track the required number of degrees of freedom in order to effectively record and classify SASL. The device must be able to wirelessly connect to a cell phone or computer so that it is portable and does not confine the user to certain areas when using the device. The device should have the option of being powered by a battery or a USB cable. The device must be able to be used by multiple users and be able to simply calibrate to the current user.

ii. Quality

The data glove prototypes must be made of robust materials in order to ensure that it will survive the testing process and the rigours of day to day use. The lifespan of the device must be considered and extended as long as possible. The sensors used in the device must give clean, repeatable readings in order to ensure reliable results.

iii. Economic Factors

The research has been limited to funds allocated by the research supervisor and resources made available by the University of Cape Town. While there has been a budget allocated, the device must be made as low-cost as possible and should not have any unnecessary components that add no value to the device. The device should be made low-cost in order to improve its availability for further research. The cost of the device will compared to that of research and commercial devices.

iv. Ergonomic Factors

The data glove must be made to fit the user well and not cause any discomfort while the device is being used. It should also not restrict the user's movement at all as they need to be able to move their hands freely while performing gestures and signs. The glove should be easy to don and remove and must be simple to use and understand. The device must take variety of users hand sizes into consideration and adapt where possible.

1.3.2 Database

The database must be recorded using the data glove prototype. This database must be made up of the most common static gestures in SASL. Each gesture should have been recorded using multiple users in order to ensure that the database is robust. The database should contain the data in multiple scenarios from simple clean data to data that simulates 'real-life' scenarios.

1.3.3 Classification

Multiple classification algorithms should be evaluated in order to validate the database. While this is not a main focus of the research machine learning algorithms will be considered and implemented in order to validate the device and database that was created.

1.3.4 Timescale

The research began in February 2013 and concluded in August 2014.

1.4 Methodology Overview

A brief description of the methods used in this research is given below:

- A literature review was conducted in order to gain deeper insights into data gloves, sign language databases and machine learning algorithms.

1.4.1 Data Glove

- Specifications for the design were laid out.
- Sensors were chosen and evaluated to determine the best fit for the data glove.
- A prototype was built and evaluated in terms of repeatability and reliability using an established testing process. The results were compared to other similar devices.
- The prototypes were finalised and built.

1.4.2 Database

- The static gestures to be recorded for the database were determined.
- The testing process was developed.
- Data was collected in three scenarios in order to create a robust database.
- The data was organised and cleaned for use in the machine learning algorithms.

1.4.3 Classification

- Three classification algorithms were chosen for the static data and implemented in MATLAB
- The results of the algorithms were compared and evaluated in order to determine the best algorithm for the system.
- Feature extraction algorithms were considered for the continuous data
- The resulting features were input into the trained neural network to determine classification accuracy for continuous data

Once these sections had been completed, conclusions were drawn and recommendations for the future were made.

1.5 Significance of study

There is a large gap in the research of South African Sign Language classification and translation and this is due to the lack of the availability of digital database. Creating such a database and pairing it with a data glove system that can be used in conjunction with it would help with the research in this field. This research could ultimately improve communications between hearing and Deaf communities and positively affect hundreds of thousands of people living in South Africa. While this research will not provide a complete database of all the SASL gestures it will be a beginning for other researchers to build upon. Once the entirety of SASL has been recorded the research could then be commercialised in order to help people to communicate in their day to day life. This would lead to equal opportunities for Deaf people, ranging from education to careers. Another significant use of such a device would be that for teaching SASL.

1.6 Other applications for proposed investigation

While the main purpose of this research is automatic sign language translation (ASLT), a data glove device also lends itself to several other applications in a variety of fields.
The basis of ASLT is that of gesture recognition and used for improving machine-human interaction. This could make working machines far more intuitive and reduce the time it takes to learn new machinery. It could make interacting with machines or computers more natural.

Linguistically, the database can be beneficial as the dataset could be used to monitor the evolution of the SASL the influence of different cultures and language groups within South Africa on SASL and to compare the structure of SASL to other international sign languages to see what kind of effect other countries have had on our own sign language.

Data gloves have been used successfully in medical analysis and rehabilitation. An example would be for post stroke patients and the rehabilitation of their motor control. The data glove could be used by the patients at home saving them the hassle of location a medical practitioner. It is also beneficial for the medical practitioner as they have access to more information more frequently and can tailor exercises to individual patients in order to optimise recovery.

Data glove systems have been used effectively in several different industries from information interaction to ergonomics. As the technologies improve, data glove devices' capabilities increase and more applications will be found.

1.7 Scope and Limitations

The research focuses on the development of a low-cost data glove for the purpose of recording a SASL database. It involves evaluating various available sensors in order to determine which will be most suitable for that of ASLT. Within the scope of the research is to keep the costs of the device as low as possible, so the research limits the number of sensor by the available budget and of only using sensors that are applicable to recording SASL. A data glove device will be developed and evaluated in terms of repeatability, accuracy and reliability of its data. It shall be compared to other similar research devices and that of commercial devices available. While the scope of this research limits the device to only recording static gestures, the device should be designed so that it can easily be extended to that of recording dynamic gestures.

This research is limited to using the developed data glove to record the most common static gestures of the South African Sign Language. This limits the recordings to that of the SASL alphabet used for fingerspelling and the numbers from zero to nine. This data will be collected in three separate manners to make the database more robust and more similar to that of real life scenarios.

The classification of the database shall be limited to that of using already available libraries, resources and software in order to implement the machine learning algorithms, as this is mainly to show proof of concept and to show that the dataset can be used effectively. The neural networks will not be programmed from the ground up as this is not the main focus of the research.

Due to time and availability the number of test subjects did not exceed that of 5. Two of the subjects were female and three were male. The database was limited to static gestures and so the skill of professional SASL translators was deemed unnecessary. Creating a dynamic gesture dataset would require the use of professional translators as the naturalness and experience with the language would be important in recording the movements.

1.8 Plan of development

As mentioned previously this research has been divided into three main sections; glove development, database creation and gesture classification. These three sections have been set out over 7 main Chapters.

A critical review of relevant literature was conducted in order to gain more insight into the research. This literature review has been split into two chapters; Chapter 2 focuses on the development of the data glove whereas Chapter 3 focuses on the database creation and the gesture classification. Chapter 2 reviews the human hand and its constraints, the history of data gloves, current commercial data gloves and the various sensors that have been used to create these data gloves.
Chapter 3 reviews both database creation and gesture classification. The database creation review begins with the various problems and limitations found in automatic sign language research and then moves on to different approaches to modelling sign language. The gesture classification review focuses on neural networks; the design approach, background theory, various training algorithms and performance metrics.

The specifications of the data glove were laid out in Chapter 4 which focused on the hardware development. The sensors were tested and evaluated and then the overall glove was evaluated for repeatability, reliability and ergonomics. The finalised glove design can be found in Chapter 5. It breaks down all of the systems used in the data glove and discusses various design factors such as the safety and cost of the device.

The approach to recording the South African Sign Language database and the implementation of the neural network can be found in Chapter 6. The different types of data that were recorded are discussed as well as the methods for cleaning and analysing the data. The collected data was compared and evaluated. The chapter ends with the design and implementation of the neural network, the classification accuracy of the different datasets and which algorithm was superior.

The research concludes on Chapter 7 which discusses the various advantages and disadvantages of the research as well as recommendations for future work. Chapter 7 is followed by the references, appendices and EBE Ethics related documentation.

2. Literature Review: Data Glove

2.1 South African Sign Language

South African Sign Language is the primary language used by the South African Deaf community. It was developed with heavy influences from German and American Sign Language, but over time it has become a fully-fledged language in its own right. While this is not the only form of Sign Language in South Africa, it is supported by the government and Deaf organisations. American Sign Language is popular but teaching ASL in South Africa is discouraged in order to unify the Deaf community by having only one national Sign Language [2] . While SASL was developing there was a large problem with consistency which depended on several factors, such as location in South Africa, culture and home language could affect the South African Sign Language that one signs. The language was disjointed and each fragment had their own 'accents' in the gestures signed which affected the language. In order to remedy this problem an official version of SASL was agreed upon and this was promoted by using SASL translators on the news [2]. This has helped the language to become more uniform and gain more traction as an official language.

South African Sign Language is limited by its ability to be understood by the majority of the South African population. This has a negative impact on educational, social and economic opportunities for Deaf and hearing impaired people. A device that provides an interface between SASL and another language would allow a Deaf or hearing impaired person to communicate with a broader audience without the need for an interpreter.

While some progress has been made to make SASL more official within South Africa, there are still huge deficiencies when it comes to how Deaf learners are taught at schools.
"Section 6(4) of the South African Schools Act 84 of 1996 recognises South African Sign Language for use in teaching and learning (LOLT) of Deaf learners in public schools. In spite of this provision in SASA, there is currently no approved South African Sign Language curriculum for grades R-12 except pockets of similar curricula in provincial departments. Furthermore, South African Sign Language is not yet one of the official languages of government in South Africa. In line with this, the President of the Republic of South Africa proclaimed in December 2012 that South African Sign Language must be developed and standardised for it to be one of the 11 official languages of government." [2]

There are many disadvantages that come with being deaf or severely hard of hearing. The main concern is the lack of ability to communicate with people that do not use SASL. This affects many facets of their lives, especially their education. The communication gap is so large that the Deaf community considers themselves separate and are considered a culture unto themselves. This is why the "d" is capitalised when referring to the community.

"Language and communication are at the heart of everything we do as humans and without them any academic, cognitive, emotional or social development becomes difficult. " [2] [5]

Deaf learners are excluded from the current education system which leads them to have unequal education opportunities. Many Deaf learners leave school functionally illiterate which leads to the exclusion of further education and training outside of school. This becomes a major problem as it severely limits their employability. [2]

There is a major lack of institutions that specialise in Deaf education and the schools that do exist are inadequate. In many cases the teachers have not made the paradigm shift from a Medical Model (where Deaf learners are seen to have a medical problem that needs 'fixing' to become 'normal') to a Social Model (where the Deaf learner belongs to a language and cultural minority group). Despite the many attempts of institutions such as DeafSA, to enlist the support of the Department of Education, there is still no formal training of educators for SASL. [2]

"It is in the interface between the signed and the spoken language that technology plays a part, whether hearing aids, cochlear implants, cell phones, computers, webcams, iPods, or other appliances. The effectiveness of these devices is frequently oversimplified and overestimated. For instance, aural support that works in a special acoustic classroom may not necessarily be effective during school breaks, in the holidays, in a public meeting or a place of worship. In the highly individual responses to (and high cost of) most of this technology, learning their national Sign Language remains the most essential need for Deaf children. " [6]

Given the problems mentioned and the advancement of technology, research has been undertaken in order to translate Sign Language into spoken language. This research is called Automatic Sign Language Translation (ASLT) and it is being studied across the world for many different Sign Languages. One facet of ASLT is the capture and recording of data used to translate the Sign Language. There have been two major approaches to this; visual and glove-based. This research explores the development of a data glove based approach. Such a device could be hugely beneficial in bridging the communication gap between hearing and Deaf communities.

2.1.1 Characteristics of South African Sign Language

Each SASL sign is made up of a specific combination of five main characteristics. These characteristics usually occur simultaneously in order to create unique signs. These characteristics were taken from Finger Talk [7] and SLED's educational dictionary [8].

i. ***Hand Shape***

The hand shapes in SASL have been compared to that of the mouth shapes made in spoken languages. When your mouth is shaped in a certain way a certain sound will be made. Similarly in SASL the hand shapes show certain parts of a word.

Figure 1: Examples of the South African Sign Language Alphabet

Hand shapes are created by varying the flexion/extension of ones fingers as well as abduction and adduction. There are over 50 unique hand shapes in South African Sign Language. The six static gestures (figure 1) show the variety seen in the South African Sign Language alphabet. The gestures for A, B and C show the typical three positions of the joints. These three positions are fully flexed ("A"), fully extended ("B") and an in-between stage of roughly 30-60 degree flexion ("C"). In order to classify these gestures correctly, it has been decided that a minimum resolution of 30 degrees would be required. The second set of letters in figure 1 "R", "U" and "V" show that the abduction/adduction of the joints is also important for creating different hand shapes. Care must be taken to ensure that each static gesture can be differentiated with the use of sensors.

A number of different approaches and sensors have been used in order to record the positions of the Sign Language User's fingers including strain sensors, carbon ink based flex sensors, optical encoders, fiber optics, ultrasonic transducers, inertial devices, inductive encoders, Hall-effect based devices, piezo-resistive materials and capacitive or magnetic devices. These sensors will be discussed in greater detail in a later section.

ii. ***Orientation of palm***

The orientation of the palm refers to the direction of the Sign Language speaker's palm is facing. Inertial measurement units are the most effective sensors when trying to record orientation as they are light, small and can easily be mounted on the back of the data glove.

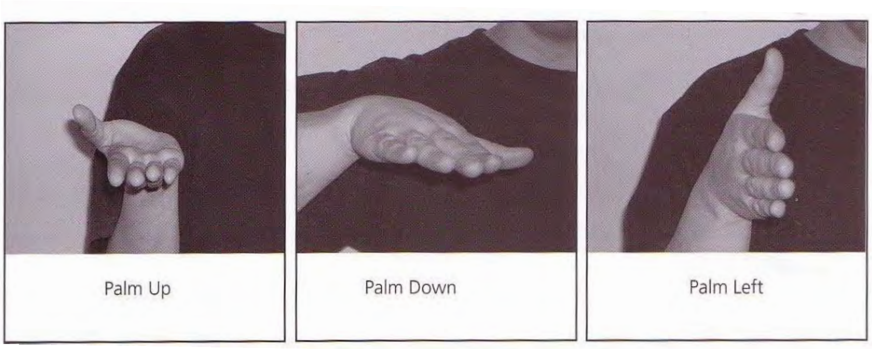


Figure 2: Examples of palm orientations used in SASL [8]

iii. ***Movement***

Dynamic gestures consist of motion in order to create more complex meanings. These movements contain specific movements and stops. Examples of motions would be arcs, zig-zags and circles. These movements can be repeated in various manners in order to create further meanings, an example would be the sign for the word sorry. It requires the signer to repeat the same gesture in front of their face. Most of the letters in the SASL alphabet are made up of static gestures but there are two exceptions; the letters "J" and "Z" (figure 3) are dynamic gestures and require movement in order to complete the gesture. When focusing on static gestures these letters usually are left out due to their dynamic nature. A variety of 3 dimensional trackers have been used in order to record these motions and they included magnetic, ultrasonic, optical and inertial trackers.

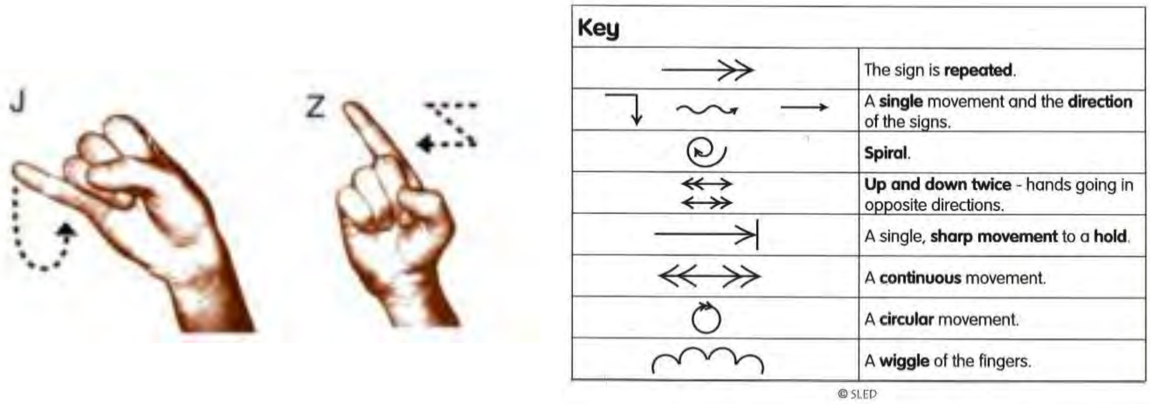


Figure 3: Left - South African Sign Language Letters That Are Dynamic Gestures. Right - Example of a movements used in SASL dynamic gestures [8]

iv. Location

This characteristic refers to the location in space where the sign or gesture is taking place. The space in front of one's body is called the neutral space. Different words are made in different locations around the body; an example of this would be the gesture of the word dog versus the word for woman. The gesture for dog is made at the speaker's hip, whereas the gesture for woman is made at chest level. The amount of space used to make the gesture also refers to the loudness of the word being signed. The more space used, the 'louder' the word being signed. Smaller gesture spaces used refer to 'softness', akin to a whisper. The same sensors have been used to measure location as the ones used to measure movement.

v. Non-manual features

Facial expressions are used to add extra meaning to the gestures being used; the expressions are used to convey various emotions. Given the scope of this research, which is limited to the development and use of a data glove, this characteristic has been omitted. The literature is unclear as to whether using non-manual features are relevant for classification or not. Some Sign Language Translators maintain that non-manual features can completely change the meaning of the gestures. They believe that a gesture coupled with different non-manual features should be classified as different gestures. Others state that it merely gives a bit more information such as emotions portrayed, and will not change the fundamental meaning of the gesture. As non-manual features are outside of the scope of the research, this question will not be researched but should be considered in future work.

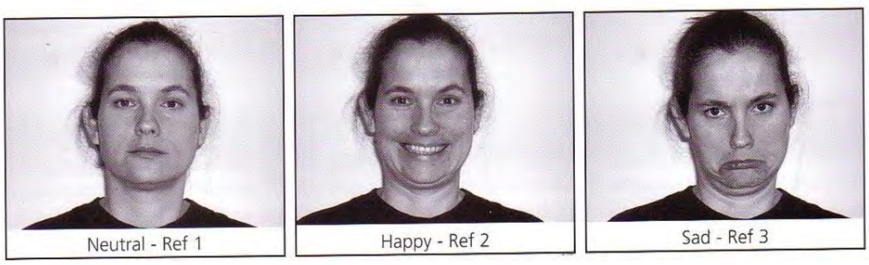


Figure 2: Examples of non-manual features used to portray emotions [8]

2.2 The Human hand

2.2.1 Degrees of freedom

The human hand is an amazing tool and has a remarkable 27 degrees of freedom at its disposal. The first 21 degrees of freedom comes from the fingers: "The distal interphalangeal (DIP) joint and proximal interphalangeal (PIP) joint each has one DOF and the metacarpophalangeal (MCP) joint has two DOF due to flexion and abduction. The thumb has a different structure from the other four fingers and has five degrees of freedom, one for the interphalangeal (IP) joint, and two for each of the thumb MCP joint and trapeziometacarpal (TM) joint both due to flexion and abduction." [9]

The final 6 degrees of freedom come from the hand's movement in space. Translation in the Cartesian x, y, and z planes as well as rotation in the form of roll pitch and yaw accounts for the final 6 degrees of freedom.

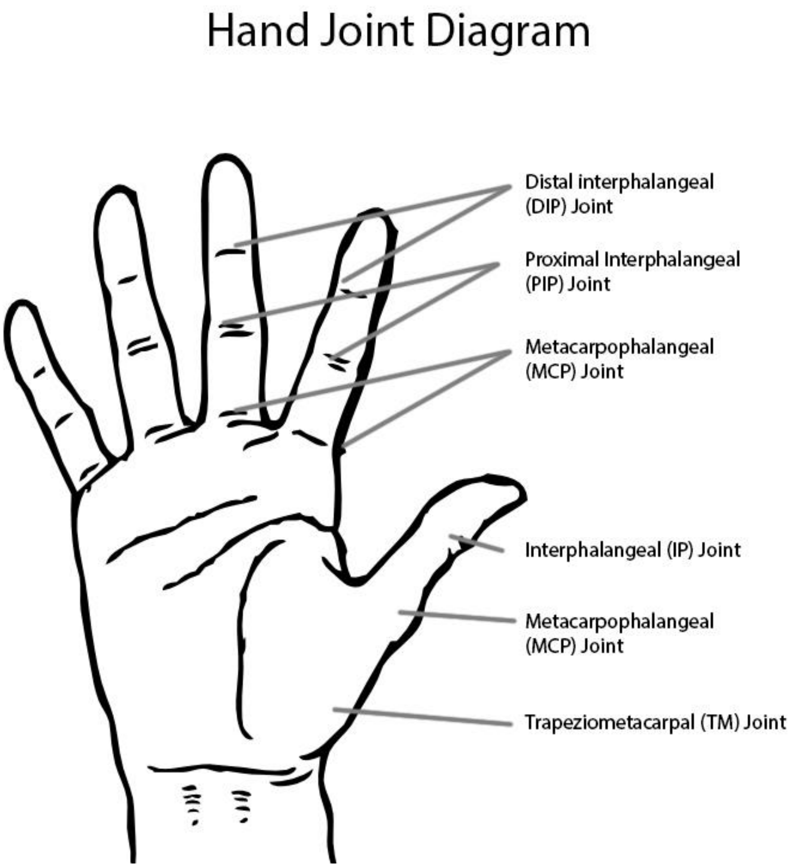


Figure 3: Diagram of human hand joints

2.2.2 Hand Constraints

While the human hand is a highly articulated tool, it does have many constraints due to its anatomy which must be considered when designing and implementing a data glove. For example, fingers cannot bend backwards by themselves. Three types of hand constraints have been established by and taken from Lin [10] as well as Lee and Kunii [11]:

i. Type 1

This type of constraint refers to the range of motion of the fingers. This type of constraint is limited to finger motion without the aid of external forces, such as bending fingers backwards with the use of the other hand.

Type 1 constraints were described using a set of inequalities:

$$0° \leq \theta_{MCP\_F} \leq 90° \quad (1)$$

$$0° \leq \theta_{PIP\_F} \leq 110° \quad (2)$$

$$0° \leq \theta_{PIP\_F} \leq 110° \quad (3)$$

$$0° \leq \theta_{DIP\_F} \leq 90° \quad (4)$$

$$-15° \leq \theta_{MCP\_AA} \leq 15° \quad (5)$$

Where, F denotes flexion and AA denotes abduction/adduction.

Another limitation which is commonly agreed upon is that the middle finger generally displays minimal abduction/adduction motion, which leads to the following constraint for the middle finger:

$$\theta_{DIP\_AA} = 0° \quad (6)$$

A similar constraint is given to the TM joint for the thumb due it its minimal motion:

$$\theta_{TM\_AA} = 0° \quad (7)$$

This reduces the degrees of freedom of the thumb from five to four parameters.
Finally the index, middle, ring and little fingers are considered planar manipulators due to the fact that the DIP and PIP joints only have one degree of freedom, flexion.

ii. ***Type 2***

Type 2 constraints are often called dynamic constraints as they relate to the limits imposed on joints during finger motion. Dynamic constraints can be divided into interfinger and intrafinger constraints. Interfinger deals with constraints imposed on joints by adjacent fingers. An example would be that when the index MCP joint is bent, the middle MCP joint is forced to bend as well.
Intrafinger deals with constraints imposed on joints by joints on the same finger. This constraint is commonly understood as needing to bend your DIP joints in order to move you PIP joints for your index, middle, ring and little finger. The constraint can be defined as:

$$\theta_{DIP} = \frac{2}{3}\theta_{PIP} \quad (8)$$

It is quite clear that, due to this constraint, the DIP joints do not need to be measured using sensors as they can be approximated easily enough. This simplifies the development of the glove design and helps to reduce costs. Combining the constraint equations mentioned above allows us to reduce the hand model from 21 DoF down to 15 DoF. Experiments have shown that reducing the number of degrees of freedom in this manner does not degrade the performance of gesture recognition [10] [11].

iii. ***Type 3***

This type of constraint deals with the 'naturalness' of hand movement and cannot be as simply defined as inequalities with the previous constraints [10]. An example to explain this constraint would be making a fist with your hand. People generally tend to curl all of their fingers at once instead of curling each individually. The naturalness of hand movements differs from person to person but in general they are similar enough for everybody. This constraint cannot be simply represented using equations or inequalities. This constraint affects dynamic gestures more than that of static gestures but it will affect static gestures when trying to classify them when considering continuous data.

2.3 Data Gloves

A data glove falls in the category of glove-based systems, which is composed of an array of sensors and electronics used in order to acquire and process user data. These sensors and electronics are mounted on the user's hand. The sensors on a data glove are generally used to measure the configuration/movement of the user's fingers as well as the motion of the user's hand. This data can then be used in a variety of different applications from medicine to sign language translation.

2.3.1 History and Context

The idea of the data glove was first created by Richard Sayre although he never managed to make such a device. The first actual implementation of the data glove was in 1977 by Thomas de Fanti and Daniel Sandin. They named the device the Sayre Glove in honour of the Richard Sayre. The glove measured flexion by using flexible tubes, on one side there would be a light source and on the other side a photocell. As the user's fingers bent, less light would reach the photocell which led to a change in voltage that could be correlated to the angle of the finger flexion [12].

Gary Grimes created the next worthwhile data glove in 1983. It was called the Data Entry Glove [13]. This device used a host of different sensors; touch/proximity sensors to sense if the thumb was touching another part of the hand, "knuckle bend sensors" to measure the flexion of the thumb, index and small finger, two tilt sensors for measuring the tilt of the user's hand in the horizontal plane and two inertial sensors to measure the forearm movement and the flex of the wrist. The device was as an alternative input device which mapped various combinations of sensor readings to that of ASCII characters.
Unfortunately these gloves were cumbersome, had limited sensors and were hard wired for specific functions. This limited their usefulness and popularity, hence they were never commercialised.

The DataGlove [14], initially developed by Zimmerman in 1982, was the first glove to be widely used in applied research. The DataGlove was released with new fibre optics in 1987 and came with 5 to 15 sensors. Data gloves became more popular in applied research worldwide after the release of the DataGlove in America. Most of the DataGloves had ten flex sensors; eight measuring MCP and PIP joints and two measuring the thumb. Some versions came equipped with sensors to measure abduction/adduction.

Eventually a low cost version of the DataGlove was released and was called the Power Glove which was used as a controller for the Nintendo console of the time. From this point on a number of gloves were created and as technology improved so did the gloves, allowing them to be used for more applications and various fields of research. Although these gloves differed in many ways from the original DataGlove they all had similar characteristics of measuring finger flexion, measuring hand movement, having sensors mounted on gloves and attempting to be a general purpose device.

2.3.2 Current Gloves

Since the development of the Sayre Glove in 1977 [12], there have been many different data gloves developed for a range of applications. For an extensive history and review of data gloves refer to "A Survey of Glove-Based Systems and Their Applications" [15]. The most commonly used data gloves used in automatic sign language translation (ASLT) are the Cyber Glove III [3] and 5DT Data Glove [4].

A pair of 5DT gloves costs € 1526 (innovatecno.com), just under R22 000, at the time of this research. This shows how expensive consumer data gloves can be and why it is important to develop a low-cost data glove for local use. There have been many custom gloves designed by researchers over the years but they are generally built for a specific task and the plans have generally not been released.

A short summary of current gloves developed for the application of ASLT can be found below:

Device	Technology	Sensors	Precision	Application
StrinGlove	Magnetic	24 Flex Sensors	12 bit	Sign Language Translation
5DT Data Glove 14 Ultra	Fiber Optic	14 Flex Sensors	8 bit	Sign Language Translation and Virtual Reality
AcceleGlove	Accelerometers	6	6.5 deg	Sign Language Translation
Cyber Glove	Piezo-resistive	22 Flex sensors	8 bit	Sign Language, Virtual Reality, Robotics, Entertainment, 3D Modelling, Analysis of Motor Performance
Human Glove	Hall Effect Sensors	22 Flex Sensors	0.4 deg	Medical and Analysis of Motor Performance

Table 1: Abridged summary of current data gloves taken from [15]

2.3.3 Applications of Data Gloves

It is important to note that while the data glove that is being developed in this research is mainly aimed at Automatic SASL translation, a well-developed data glove can be used for many different applications. As technology improved and sensors became more reliable and accurate, the number of potential applications grew. A brief list of these applications has been given below:

i. Human-Machine Interaction

Data gloves can be used in order to interact with machines and computers in a more natural and intuitive manner. It can be used with CAD programs in order to 'reach' into the computer and shape components. Gestures can be used in order to visualise and interact with data in a more fluid manner. Robots could even be taught movements by the user or controlled via the glove in a telecommunication manner.

ii. Art and Entertainment

Data gloves have been used as controllers for video games, used for animation in movie productions and even as an instrument to create music (Imogen Heap)

iii. Medicine

The use of data gloves in medical applications is only a more recent application of the data glove as sensors are only becoming accurate and reliable enough for medical related tasks. Data gloves can be used in motor rehabilitation for post stroke patients, analysis of motion and even for medical education and training.

iv. Ergonomics

Data gloves have many uses in the design of ergonomics. Data gloves can be used as an alternative to traditional motion analysis and be used to see how well a user's hand fits or grips an object. The data recorded by the glove can be analysed in order to reduce strain and increase comfort and ease of use of tasks and products

2.4 Characteristics

When developing a data glove there are many factors to consider in order to ensure that the specifications of the design are met and that the glove will be suitable to the task laid out for it.

2.4.1 Gesture Types

The type of gestures used in SASL can be broken down into two main types; static and dynamic.
Static gestures are made up of hand shape, palm orientation and location. An example of a static gesture would be that of the SASL alphabet and number system. Each sign has a unique hand shape, position and orientation. There is no movement required to form the gesture, a "thumbs up" gesture would be considered a static gesture as no movement is required to understand the gesture.
A dynamic gesture couples a static gesture with a movement component in order to complete the sign. A wave or drawing a circle in the air with your hand would be an example of a dynamic gesture. There are many issues when considering dynamic gestures. These problems will be discussed in a later section.

2.4.2 Sensor Characteristics

There are many characteristics that must be considered when designing and choosing sensors for a data glove. These characteristics affect the system's ability to be effective in various scenarios. The sensors used on the glove will affect the devices capability to sense gestures, its mobility, its comfort for the user, etc.

i. Data type

The sensor data can either be discrete or continuous. Discrete data can only read set values in a range for example binary data can only be 0 or 1; there are no values in between. An example would be that of a contact sensor which reads a 1 when touching and a 0 otherwise. Continuous means that it can read numbers between the 0 and 1; the number of values it can read between 0 and 1 depends on the resolution of the sensor. Flex sensors can be used to obtain continuous data when measuring finger flexion.

ii. Number of Sensors

The number of sensors used will affect the robustness of the system. This means that if there are too few sensors then the system will not reliably be able to fully translate SASL. While there are 27 degrees of freedom in the human hand, there are many constraints placed on the hand by its anatomy. This means that not all of the DOF have to be covered in order to effectively record a hands movement. Previously it was shown that only 15 degrees of freedom are needed to sense SASL. Sensors are limited to the data glove and so non-manual gestures will not be considered.

iii. Method of mounting

How the sensors are mounted on the data device will affect their performances. A common approach is to mounting the sensors directly onto the users hand or on a glove and an alternative is to mount the sensors on an exoskeleton device.

The exoskeleton approach places the sensors on a frame that is then mounted on the user's hand. Exoskeletons are not as comfortable as gloves but they do give accurate data from the sensors. The main problem is that they are bulky and clumsy to wear. A product for automatic sign language translation should be lightweight and hardly noticeable for the user as it will be used on a day to day basis.

Figure 4: Left- Festo Exohand (festo.com). Right - An example of a basic data glove

The glove approach is most common for a number of reasons; it is simpler to make, more comfortable, the materials are simple to use and readily available. However there are some downfalls to this approach; if the glove does not fit your hand well it will skew the data read by the sensors, the material used for the glove is likely to have some stretch in it and over time the accuracy and repeatability of the data from the sensors will be affected. The major trade-off to consider is either to have the glove custom made for each user which is time consuming but will yield accurate results or to have gloves made for general sizes (S, M or L) in order to cut down prices and insure that the glove will fit most people but this approach gives less accurate results. While this is a major issue to consider when choosing the mounting method, the extent of the problem can be reduced using circuitry and software to be able to recalibrate the sensors for the user over time. Mounting the sensors on a glove was chosen for this research due to the reasons mentioned above.

iv. Sensor Resolution

If the sensors do not have a high enough resolution it will be hard to accurately distinguish between similar signs and gestures. The precision of the sensors depends on the intended application of the device. For simple gestures such as static gestures it would be fine to use lower accuracy sensors but for more complex/dynamic gestures it would be beneficial to use higher accuracy sensors. If possible higher resolution sensors should be considered in order to expand the potential number of applications of the glove where possible. While resolution is important, the repeatability and reliability of the sensors is also very important to consider.

v. Sampling Rate

The speed at which the data is read will affect the ability of the glove to recognise gestures. It would be redundant to receive copious amounts of data for static gestures as there will not be much change in the data but if the sample rate is too slow for a dynamic gesture then information will be lost which could result in incorrect classifications. A fine balance must be found so that the gesture will not be over read but data must not be lost.

2.5 Sensors

The sensors used for data gloves can be broken down into two main groups; sensors that measure the flexion/movement of the fingers and the second group measure the movement and orientation of the entire hand. Other sensors have been used to increase the functionality of the device but when it comes to ASLT those are the two groups that are required.

2.5.1 Finger

The finger sensors are used to measure two aspects of the users fingers; the angle of flexion of the joints and the abduction/adduction of the fingers. Sensing the position and movement of the fingers is crucial for automatic sign language translation. If you cannot track the user's fingers then you cannot detect the current hand shape that the user is signing. Hand shapes, arguably are one of the most important characteristics of the 5 that make up sign language [8].

i. Optical Linear Encoder (OLE)

The basic theory of how OLE's work is straight forward, the sensor requires a LED, a photosensor and an encoder of some sort. The encoder can be a disk with evenly spaced slits in it. The light passes through the slits and is picked up by the photosensor. When the disk is moving a pulse train is sensed on the photosensor side. Each pulse is a count and related to the movement of the disk. This sensor can be customised to suit ones need; instead of a slit on a disk it could be alternating strips of black and white which reflects the light onto the photosensor.
A data glove called SmartGlove [16] [17] was designed using OLE sensors. They performed well but the sensors were expensive ($50 per unit) and were custom made which means that they were not commercially available.

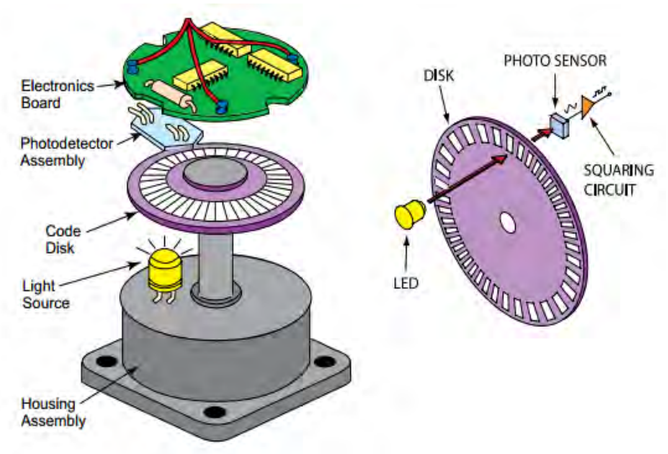
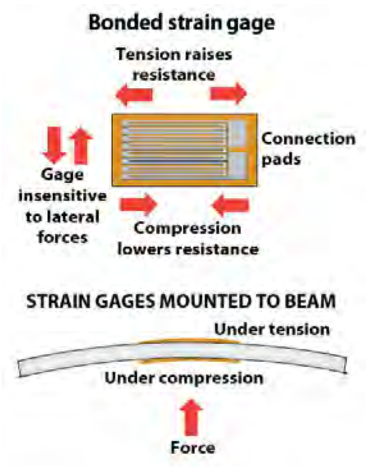


Figure 5: Exploded view of an Optical Linear Encoder

ii. Strain Gauges

A strain gauge is a sensor that varies its resistance as a result of force, pressure, tension, weight, etc. The resistance change is minute and so they must be connected to a circuit that can detect very small changes in resistance. Four strain gauges are usually connected in a Wheatstone bridge circuit, which is suited to temperature compensation, in order to detect the changes in resistance effectively.

Strain gauges have been mounted on flexible materials in order to measure finger flexion. The main problem is that strain gauges are expensive and due to the nature of the circuit required to use them effectively, they are not a viable option for a low cost data glove.



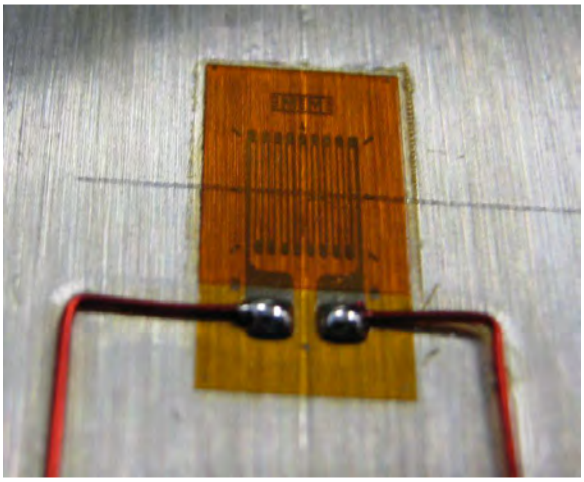


Figure 6: Left - Visual explanation of how a strain gauge works (machinedesign.com). Right - Photograph of a strain gauge (kwantlen.ca)

iii. ***Hall Effect Sensors***

A Hall Effect sensor that reacts to magnetic fields around the sensor to produce a reading. The most common use is that of a digital switch when a strong enough magnetic field is introduced. Hall Effect sensors can output analogue values that depend on the strength and polarity of the magnet. These sensors were used in the HumanGlove [18]. Hall Effect sensors are not affected by ambient conditions (such as dust or vibrations), do not require contact and can operate at high speeds. There are also some disadvantages to these sensors, their readings can be effected by external magnetic fields, have large temperature drift as well as large voltage offsets.

iv. ***Inertial Measurement Units***

IMU's have been used in the development of data gloves. Usually they are mounted on the back of the glove in order to sense movement of the entire hand but in some cases, such as the Acceleglove [19], an IMU was placed on the back of each finger in order to determine joint flexion by using the gravitational vector. Disadvantages of using IMU's for flexion sensing are that the sensors drift over time and the sensors can be quite expensive for early development and creating prototypes.

v. ***Robotic Skin Approach***

Researchers have managed to develop a robotic skin than can sense bends and pressures applied to it. A glove made from a similar robotic skin could potentially measure the hand shape of a user very accurately. If one of these skins is shaped like a glove then the flexion of the user's fingers will cause pressures at the finger joints and knuckles, which can then be related to the angle of flexion of the finger. The three main methods of creating such a robotic skin are resistive, capacitive and piezoresistive. A resistive approach was tried by the authors in [20] but this approach was complicated, required very high resolution ADC's and it required a lot of computer processing power. A capacitive approach can be seen in [21], this approach requires precision construction techniques that would be unavailable for this project. A detailed overview of robotic tactile sensing can be found in "Tactile Sensing—From Humans to Humanoids" [22]. The disadvantage is that due to the custom nature of the approach it would take time and money in order to develop it. Another problem is that it is dependent on specialised materials that have to be ordered internationally. This was the initial approach used for this research but due to unforeseen problems with acquiring the specialised materials this approach could not be implemented.

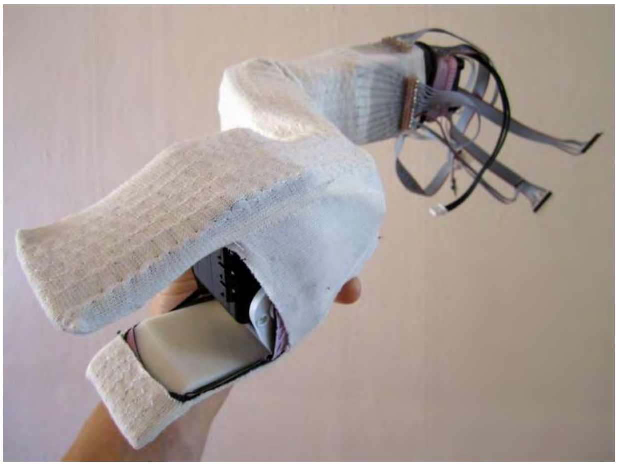


Figure 7: rSkin developed by Hannah Perner-Wilson (plusea.at)

vi. ***Flex Sensors***

The conventional manner of measuring finger flexion and movement is by using commercial flex sensors; an example is the flex sensor from Spectra Symbol or FlexPoint. These sensors are affordable, available and have good characteristics. This makes them a very good choice when designing data gloves. Commercial flex sensors have been used in commercial data gloves such as the CyberGlove [3]. These sensors are also favoured by researchers for prototyping data gloves. One of the major problems with the use of commercial flex sensors is that they are usually mounted on top of the data glove; they can slip, and are difficult to position exactly over the flexion points that need to be measured. These problems can be mitigated by using calibration software. It is possible to create homemade flex sensors for a fraction of the cost but they have been proven unreliable and inaccurate for continuous data recording. They can be used as a binary switch but not much more. Commercial flex sensors were chosen for the sensing of finger flexion in this research.

Figure 8: Flex Sensor from Spectra Symbol

2.5.2 Movement and Orientation

The tracking of the entire hand can be broken down into movement, location and orientation. While the sensors required to measure the movement and orientation of the users hand would be mainly be useful for dynamic gestures, they can be useful for feature extraction for static gestures as well as calculating the angle of the user's hand.

i. ***Mounted***

The only sensor that can be completely mounted on the glove is an inertial sensor. This sensor would be used to measure the acceleration of the hand and using this data position, movement and orientation could be calculated. The major downfall of IMU's that they drift over time and this causes inaccuracies in their data. This can be sorted out by resetting the sensor to a known position at regular intervals. This drift can also be reduced by using a more advanced IMU. For example, a 6 DOF IMU cannot compensate for yaw because it has no reference to gravity. By adding a 3 DOF magnetometer to the device, making it a 9 DOF IMU, the sensor can now compensate for yaw and give more stable results.
Another approach would be to attach multiple sensors to the user's arms in order to measure their position but this leaves the realm of data gloves and moves into wearable computers.

ii. ***Radiant***

The second option would be to have an external sensor tracking the hands motion. This can be done using ultrasonic transducers which has an accuracy of centimetres or magnetic trackers can be used as they have accuracy of millimetres. The problem with these two options is that they require the user to remain close to a radiant source and can be very expensive [23] [24]. An example of a magnetic tracker that has been used for this type of research is the Flock of Birds tracker. This device has been used by many data glove researchers in order to accurately track the participants hand movements accurately. Unfortunately this device is very expensive, costing roughly $2500 which is far out of the available budget for this research.

Microsoft Kinect

Microsoft's Kinect sensor has been successfully used in order to track body and limb motion. Advances in the software have also allowed the device to track the hand and finger movements of the user as well. This device is not portable enough for a person to use during every day activities.

Figure 9: Microsoft Kinect Sensor (microsoft.com)

Leap Motion

The latest technology that has been developed in recent years is a 3D sensor called the Leap Motion (leapmotion.com). The sensor was successfully funded through Kickstarter and made available to the public in 2013. The device was created as a novel human-computer interface and could track the user's hands very accurately with low latency tracking. Disadvantages of the device are that it has to be tethered to a PC which limits the portability of the device. It also requires a lot of processing power in order to track the users hand data. While this device looked promising at first its software fell short of the promises made before the release. The device would lose track of fingers and even the user's entire hand if placed in certain positions. The demonstration videos also implied that the user would have access to the sensors raw data in order to create point cloud tracking but this was not the case and so the device was not a useable option for this project. Mid-year 2014, Leap Motion released updated software that has very accurate hand and finger tracking. It would be perfect for the purpose of ASLT but unfortunately the software update was released too late to be considered for this project.

Figure 10: Leap Motion detecting a user's hands as inputs (leapmotion.com)

3. Literature Review: Database and Classification

3.1 Sign Language Modelling

While this research focuses solely on the recording and classification of static gestures for South African Sign Language, it is important to consider and study the language as a whole. This is done in order to gain a deeper insight into the structure as well as the major problems when trying to implement automatic sign language translation. These factors must be explored in order for this research to be expanded to dynamic gestures in the future.

As mentioned previously, there are two main approaches to capturing sign language data. The first being data glove based and the other being vision based. The advantage of using a data glove is that the data is extracted faster and more easily. When using a vision based system there are many complicated calculations to perform. First the hands and face must be located and tracked. This can be difficult as lighting conditions and contrast can drastically affect this process. Once located, the five main sign language characteristics must be determined using the 2D image. This can take a lot of time and effort to implement. The main disadvantage of the data glove approach is that the device does not record the non-manual features.

3.2 Problems and Limitations

Automatic Sign Language Translation, and gesture recognition in general, has many problems and limitations to overcome in order to correctly classify and translate sign language signs into text. The major problems have been clearly outlined in [25], [26] and by the authors of SignSpeak [27]. These problems have been described below:

3.2.1 Variation

Each person will perform their own interpretation of given gestures or signs and this causes variation in the characteristics of the signs. Another concern is that fact that the same person's gestures will vary depending on their mood, emotional or physical state. If a person is tired their signs may be more lazy and sluggish, this will cause issues with the classification phase. An angry person's signs may be more fast and abrupt. Variations in how signs are executed will also be affected by the area, culture or home language of the speaker as these factors create 'accents' on the gestures. It has been seen in various studies that the accuracy of the classification drops off significantly as more participants are used for training data.

3.2.2 Co-articulation

Signs can overlap when sentences are being signed. Different signs end in various locations and the next sign will start where the previous one ended. This means that the same dynamic gesture can start from several different points depending on what the previous gesture was. This problem is called co-articulation.

3.2.3 Anticipation

The non-dominant hand starts moving towards its next position before the current sign has been completed. This can cause confusion for classification process as the non-dominant hands motion has nothing to do with the current signs gesture. In real world situations a sign language speaker's gestures unconsciously flow together more naturally to form a sentence instead of being separated perfectly into each distinct word.

3.2.4 Breaking up gestures

One of the major problems in gesture recognition is deciding where one gesture ends and the next one begins. As mentioned previously problems such as anticipation and co-articulation make it difficult to determine where one gesture starts and another ends. This is especially important in sign language translation as entire sentences made up of many gestures need to be correctly broken up into different parts. This problem also becomes a problem in more static scenarios such as finger spelling words using the sign language alphabet.

3.2.5 Distinguishing between gestures and signs

Humans often gesture with their hands while speaking; they use their mouths to speak and their hands to gesture. A Deaf person uses their hands for both speaking and for gesturing, this becomes a problem because the gestures and signs start to blur. This causes issues for the translating device.

3.2.6 Context dependency

A problem that is largely ignored by ASLT researchers is the fact that natural language, including sign language, is very context dependant. Most of these context dependant issues revolve around special gestures and signs. The translator often has no knowledge of the local area and will not be able to understand the context of the situation. Various positions would have to be localized in order to be understood and once they had been localised the device would have to store this information as it would not be signed again. Contexts for many different scenarios would have to be understood by the translator in order to effectively translate sign language.

3.2.7 Basic Unit of Modelling

When trying to classify gestures or signs they must be broken up into their most atomic level. In the case of sign language translation this means that the sentences must be broken up into words or phonemes. Phonemes are the preferred basic unit of modelling for Automatic Speech Recognition. "A phoneme is the smallest contrastive unit in the sound system of a spoken language. For sign languages, however, it is not at all clear which part of a sign should be taken as the smallest contrastive unit. In signs, several features carry meaning: hand shape, palm orientation, position, motion, and the non-manual component. If one of these elements changes, the meaning of the sign usually changes as well, so these features could be considered the basic units of sign language. However, one cannot use the same techniques that automatic speech recognition uses in spoken phoneme modelling to model these features because spoken phonemes are largely sequential, whereas the features of sign language can be simultaneous as well as sequential. Although certain features can occur only in sequence (e.g., two different hand shapes), the features of sign language are often simultaneous (e.g., a hand shape, position, orientation)." [28]

3.2.8 Transitions between signs

When two gestures are made in a row there will be extra movement between the two gestures, this is done because the user cannot instantly transport their hands from the end position of the gesture and the beginning position of the next gesture. This extra piece of movement is meaningless and does not add to any gestures despite this the data glove/camera will pick up this movement and try to classify it.

3.2.9 Repetition

Some signs require repetition in the gesture and it is important that the classifier can determine if each repetition is a different sign or if all of the repetitions make up one sign. This can cause confusion in the classification stage.

It can be seen that there are many problems that need to be considered when trying to classify sign language. This research focuses mainly on the capture and classification of single static gestures but some of the data will have to deal with the problems mentioned above, as some of the data recorded has basic dynamics added in. This is done in order to display more real world situations.

3.3 Structure of Sign Language

There have been different approaches to modelling sign language in order to record it and use it for translation purposes. A popular approach has been the phonetic structure approach which breaks up the language in to its separate phonomes. This approach was inspired by the success of the phonome approach used in spoken languages. While there have been some researchers that focus on building databases using the entire words, this has been generally accepted as an uneconomical approach. The phonome approach has been favoured by researchers because the main characteristics that make up this structure are common to all sign languages. This means that the approach can be used universally no matter which sign language you research.

3.3.1 Phonetic Structure

While there has been debate among researchers as to the best method of modelling sign language into its most basic elements when creating a database, most consider hand shape, palm orientation, hand location and movement to be important [24]. This makes logical sense as these are parts of the 5 characteristics that makes up sign language. The only one that is not considered is that of the non-manual features and this is because the focus is on data gloves which cannot record non-manual features.

An example of a Sign Language model is the Pose-Movement Model [24], it was defined as follows:

"A phonetic model that treats each sign as a sequential execution of two measurable phonemes: one static, and one dynamic.

Definition 1: A ***pose*** is a static phoneme composed of three simultaneous and inseparable components represented by vector **P** = [hand shape, palm orientation, hand location]. The static phoneme occurs at the beginning and at the end of a gesture.

Definition 2: A ***posture*** is a vector of features **Ps** = [hand shape, palm orientation]. Twenty-four out of the 26 letters of the ASL alphabet are postures that keep their meaning regardless of location. The other two letters are not considered postures because they have movement.

Definition 3: ***Movement*** is a dynamic phoneme composed by the shape and direction of the trajectory described by hands when traveling between successive poses. **M** = [direction, trajectory].

Definition 4: A ***manual gesture*** is a sequence of poses and movements, P-M-P.

Definition 5: **L**, the set of purely manual gestures that convey meaning in ASL is called the ***lexicon***.

Definition 6: A manual gesture **s** is called a ***sign*** if s= L.

Definition 7: ***Signing space*** refers to the physical location where signs take place. This space is located in front of the signer and is limited by a cube bounding the head, back, shoulders and waist."

3.3.2 Existing Sign Language Databases

There are a large amount of video databases for various sign languages but since visual ASLT is out of the scope for this project they were not considered. The only available Sign Language database that was created for data gloves is a database for Australian Sign Language (AusLan). There are two versions of this database; a low and high quality version, both unfiltered. The low quality database was created using the very out-dated Nintendo Powerglove which was originally released in 1989. The low quality database will be ignored due to the availability of a higher quality database. The high quality database was created using 2 5DT gloves and a Flock of Birds magnetic tracker. The following characteristic/attributes were recorded and represented for the dataset as follows [29]:

i. Location and Orientation:

X position: expressed relative to a zero point set slightly below the chin. Expressed in meters. **Y position**: expressed relative to a zero point set slightly below the chin. Expressed in meters. **Z position**: expressed relative to a zero point set slightly below the chin. Expressed in meters. **Roll:** expressed as a value between -0.5 and 0.5 with 0 being palm down. Positive means the palm is rolled clockwise from the perspective of the signer. To get degrees, multiply by 180.
Pitch: expressed as a value between -0.5 and 0.5 with 0 being palm flat (horizontal). Positive means the palm is pointing up. To get degrees, multiply by 180.
Yaw: expressed a value between -1.0 and 1.0 with 0 being palm straight ahead from the perspective of the signer. Positive means clockwise from the perspective above the signer. To get degrees, multiply by 180.

ii. Finger Positions:

Thumb bend: measured between 0 and 1. 0 means totally flat, 1 means totally bent.
Forefinger bend: measured between 0 and 1. 0 means totally flat, 1 means totally bent.
Middle finger bend: measured between 0 and 1. 0 means totally flat, 1 means totally bent.
Ring finger bend: measured between 0 and 1. 0 means totally flat, 1 means totally bent.
Little finger bend: measured between 0 and 1. 0 means totally flat, 1 means totally bent.
Note: Finger positions are not very exact
While this database was available it could not be used in conjunction with the data glove developed during this research. This is because of the slightly different approaches to finger tracking. The Auslan database only uses one flex sensor per finger whereas the data glove developed in this research uses 2 flex sensors per finger. Another reason that the Auslan dataset could not be used is the fact that it is all dynamic gestures whereas this research focuses on static data.

3.3.3 South African Sign Language Static Gestures

One of the main focuses of this research is to develop a static gesture database for South African Sign Language. While there are over 50 various static gestures used in SASL, the most commonly used static gestures are those used in the SASL alphabet and number system. The alphabet is generally used in fingerspelling words that do not have their own unique gesture. An example would be spelling out a name of a person. The SASL alphabet and number system can be seen below (figure 11).

There are a total of 36 gestures in total that make up the SASL alphabet and basic numbers system. From these 36 gestures only 31 of the gestures can be recorded for the static gesture database for the following reasons:
Out of the 26 letters in the alphabet only 24 of them are static gestures. The gestures for ‘j’ and ‘z’ have a dynamic component and were not considered for the static gesture database.
There have also been static gesture repeats between the alphabet and the number system. The following gestures overlap with one another: Zero and ‘o’, two and ‘v’ as well as six and ‘w’. Since these gestures are the same, the gesture can mean either one or the other depending on the context it is used in.

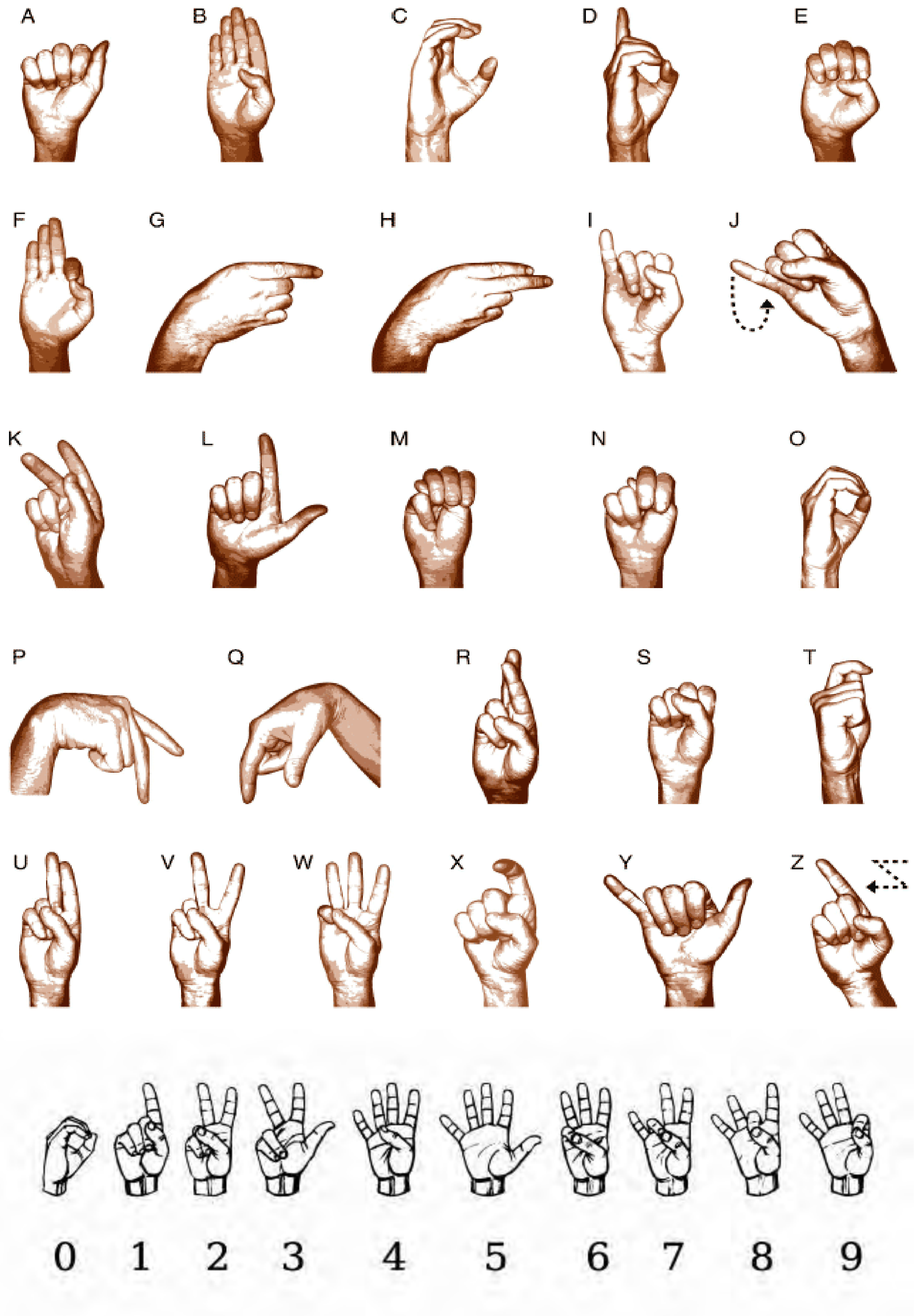


Figure 11: SASL fingerspelling alphabet [56] and numbers [54]

3.4 Machine Learning

3.4.1 Machine Learning Applications

Machine learning algorithms are very useful as they are able to adapt to many different scenarios. This can lead to very robust systems that develop behaviours that can succeed in partly unknown or dynamic environments. Different applications have different approaches and algorithms to solve the problem [30] [31]. These different types of scenarios can divided into three main categories:

i. Unsupervised Learning

This process involves a system that learns to extract statistically relevant information from a set of input patterns. The system is able to determine important information within the input patterns without any external feedback. This learning process usually occurs by detecting and extracting common or distinctive features within the inputs. One example is taking data with several different variables, using unsupervised learning and determining if there are any hidden relationships between those variables in order to categorise them.

ii. Supervised Learning

This involves algorithms that guide the training process by having examples with desired or target values. This desired value acts as the teacher for the system. These training examples help to shape the system by reducing the error between the desired response and the actual response of the algorithm. This will help it to output the correct value when the input example has not been seen by the system before. These algorithms are used for data fitting problems or classification into known categories such as in the case of automatic sign language translation.

iii. Reinforcement Learning

These algorithms operate using a kind of learning that is linked to the consequences of the agent's behaviour. The agent will perform an action and as a consequence it will be rewarded or punished depending on the result of that action. An example of this would be a hexapod robot teaching itself to walk using different gaits. The hexapod would move using a gait and would either be rewarded or punished depending on if it is moving faster than the previous attempt.

3.4.2 Related Work

There has been a lot of successful research into automatic sign language translation. Some of the most effective methods have given upwards of 90% correct classification rate. However these results can be somewhat misleading as they focus on specific scenarios that the researchers were concentrating on. When looking into the research of automatic sign language translation, different factors must be considered and these include; size of vocabulary, number of test subjects, device being used, classification technique being used and recognition accuracy. [26]

The extent of the vocabulary used can affect the overall accuracy of the classification. Unfortunately creating an extensive dataset can be very time consuming and expensive. Many researchers use a single participant to capture both the training and the testing data. As the number of participants increases the accuracy tends to drop as mentioned previously. Another factor that must be considered is if the researcher chooses to use 'unseen' participants who are completely independent of the training data. This can hugely decrease the accuracy of the classification. [25]

The type of gesture being classified is very important to the end results of the research. The gestures being considered could be that of single letters which are static gestures or full words which can be complex dynamic gestures. The data can be a single snapshot of the sensors while the user holds a gesture (in the case of static gestures) or it can be continuous data that needs to have features extracted in order to classify. Another aspect to consider is whether the classification of the gesture is continuous or isolated. The continuous case deals with having more than one gesture in a data stream and having to classify all of them accurately. The isolated case is where the user will move their hand from a neutral space to the gesture and then back again. The isolated case is simpler to implement and far more common than that of continuous classification. [25]

In automatic sign language translation there have been two prevalent techniques for classification and these are Neural Networks and Hidden Markov models.

i. ***Neural Networks***

Neural Networks are composed of elements called neurons. This technique was inspired by biological nervous systems. These neurons are connected in parallel and operate in this manner to form a network. The networks function is determined largely by the connections between them. Neurons can receive inputs from other neurons or from external stimulus. These networks are usually trained so that certain inputs lead to a targeted output. The network can adjust itself by comparing the output and the target. To train the network effectively many input/target pairs are required.

Neural networks have been successful when applied to gesture and pattern recognition applications. The advantages of Neural Networks is that they are able to get good recognition results despite noisy or incomplete data and they are also very good at pattern generalisation.

The strengths of Neural Networks are that they do not require calibration, Neural Network packages are readily available, if there is sufficient data they will give good results and unknown inputs will be classified very quickly.

Weaknesses of Neural Networks are as follows: A large amount of labelled data is required, if a new gesture is added/removed then the whole system must be re-trained for best results, Neural Networks can overlearn and this leads to a system that is not very robust, Neural Networks require a large amount of processing power and time especially when training the system and there are no formal methodology to setting up the various sections of the Neural Network (topology, unit activation functions, learning strategy, etc.) as they are usually created by trial and error. [23]

ii. ***Hidden Markov Models***

Hidden Markov Models (HMMs) have been very successful in automatic speech recognition and this is why they have become a popular tool for Automatic Sign Language Translation. HMMs are statistical models that are used in order to represent certain elements such as words or phonemes. These elements are trained by giving enough examples to the statistical model. Due to their statistical nature HMMs are able to handle variation well. Another huge benefit of using HMMs is that explicit language rules do not need to be formulated in order to classify languages, which is useful as language rules can become very abstract. Once the models are trained they are able to work in real time and do not need large amounts of processing power.

The downfall of HMMs is how long it takes to train the models. The more the variety, the more examples are required in order to train the models effectively. This means that it will take a lot longer to train the models if multiple signers are used as opposed to a single signer. This downfall has led researchers to train phonemes instead of full words. A handful of phonemes can be combined in order to make a vast collection of words.

HMMs would be the recommended approach for classification of dynamic gestures since there have been many successful studies using HMMs for ASLT. [28]

iii. ASLT Research Comparison

Tabulated summaries of automatic sign language translation research has been provided below:

Classification Technique	Reference	Isolated or Continuous	Number of Subjects	Vocabulary Size	Recognition Accuracy (%)
Recurrent NN	Murakami [32]	Isolated	1	10 JSL	96
Multilayer Perceptron NN	Vamplew [33]	Isolated	7	52 Auslan	94.2
Hyperrectangular Composite NN	Su [34]	Isolated	2	90 TWL	94.1
Two-stage neural network	Waldron and Kim [35]	Isolated	6	14 ASL	86
Artificial NN	Oz [36]	Isolated	1	50 ASL	90
Elman Back Propagation Neural Network	Saengsri [37]	Isolated	1	16 TSL	94.44

Table 2: Summary of automatic sign language translation research that specifically uses Neural Networks for classification of static gestures.

Classification Technique	Reference	Isolated or Continuous	Number of Subjects	Vocabulary Size	Recognition Accuracy (%)
Simple Recurrent Network + HMMs	Fang [38]	Continuous	5	208 CSL	92.1
HMM model sequential subunits	Wang [39]	Isolated	1	5119 CSL	92.8
HMM model sequential subunits	Wang [39]	Continuous	1	274	86.2
HMM and Parallel HMM	Vogler [40]	Continuous	1	22 ASL	95.5
Whole-word HMMs combined with efficiency techniques	Chen [41]	Continuous	1	5113 CSL	92

Table 3: Summary of automatic sign language translation research that specifically uses Hidden Markov Models for classification of dynamic gestures.

3.5 Neural Network Theory

The Artificial Neural Network developed out of the need to expand computers' information processing capabilities. Tasks that are straightforward for humans such as recognising faces or hand writing was a challenge even for the most sophisticated computer and so researchers were inspired to mimic the human information processor, the brain. This was done after noticing that the brain computes in a completely different manner than conventional computers. The brain performs certain computations such as pattern recognition or perception by organising its neurons into layers and these layers are connected into a network; by doing this the brain is able to process these functions faster than a computer. Researchers strove to learn enough about biological neural networks in order to create artificial models of them. This was done by studying the smallest processing units of the brain; neurons and their synapses. [42]

Artificial Neural Networks loosely model the human brain by simulating multiple layers of neurons by means of sophisticated software. Each neuron is linked to neighbouring neurons in other layers and each one of these links has a certain value associated to it. This value represents the strength of the connection and scales inputs and outputs of the neurons. The system learns by adjusting these weights in order to output an appropriate result.

3.5.1 Biological Neuron

The fundamental information processing unit of the brain is the neuron. Many of these units are connected together in order to create an organic neural network. The basic functionality of the neuron is to receive inputs from various sources, combine them some manner, perform a generally non-linear operation and then outputs the result. An image of a neuron is given below:

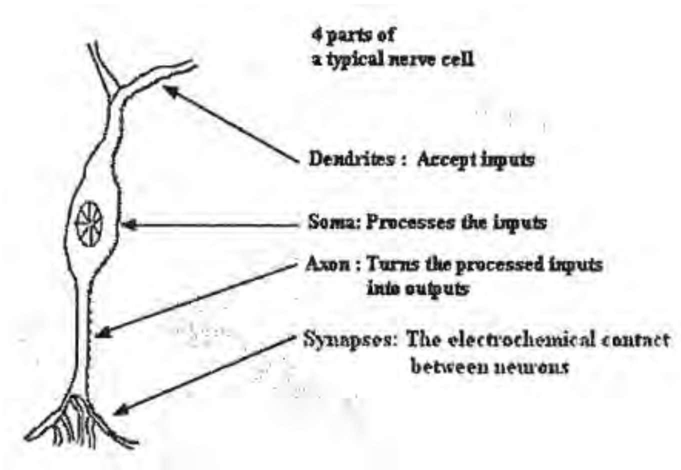


Figure 12: Labelled biological neuron [43]

The neuron can be broken up into four major components; Dendrites, Soma, Axons and Synapses. The Dendrites receive the inputs from various different sources and passes them into the Soma, which processes the inputs. The Axon turns the processed inputs into an output and the synapses connects this output to other neurons. [43]

3.5.2 Artificial Neuron

i. ***Simple Neuron***

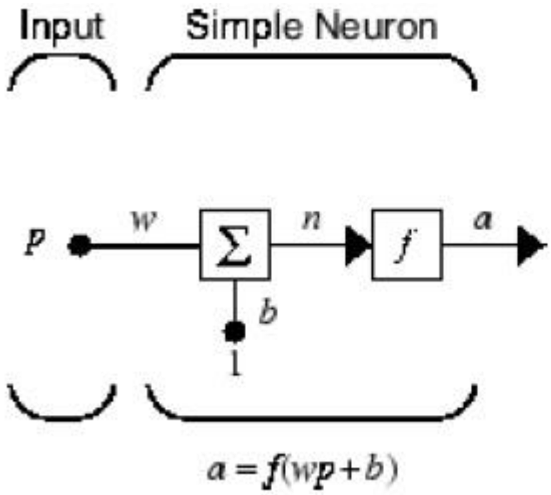


Figure 13: Block diagram of a single artificial neuron [42]

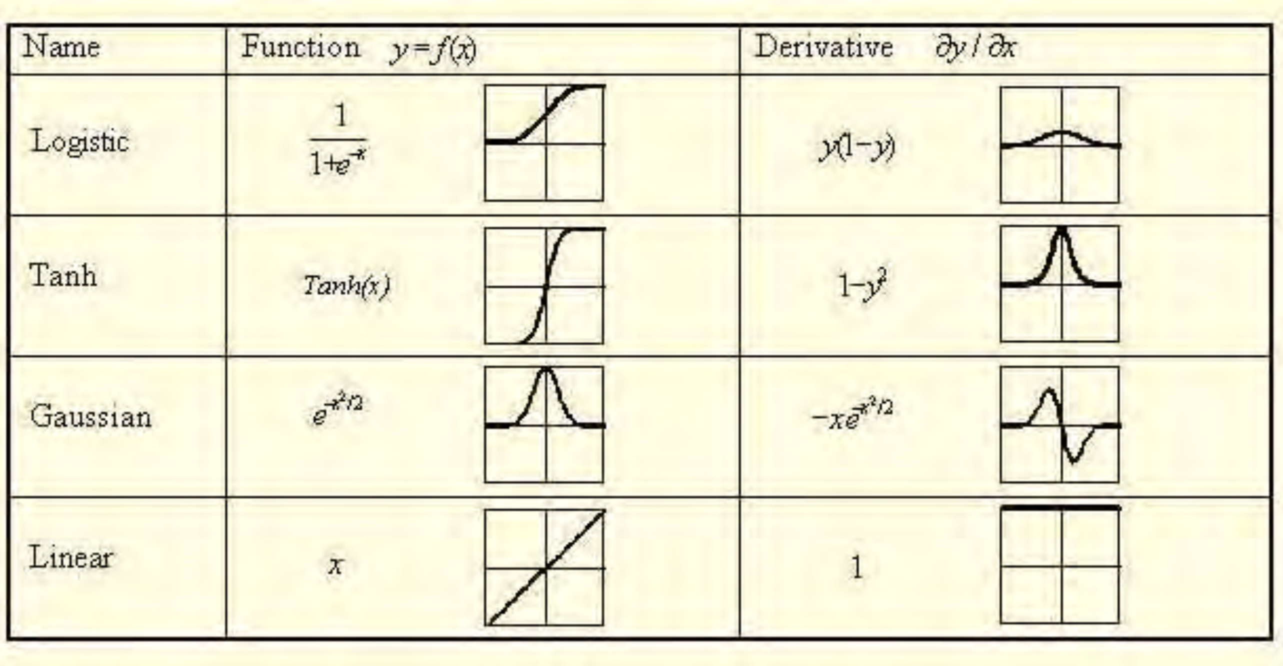
The simplest artificial neuron is that of a single input neuron. In this example the neuron has three functions. First the input p is scaled by weight w to form the weighted input wp. This is the weight function. Second, the bias b is added to wp to form the net input n. This is the net input function. This value b is used to shift the function to the left and can be considered similar to the value w but with a constant input of 1. The final step passes n through the activation function f which produces an output a. This is transfer function. [42]

It is important to note that both w and b are adjustable parameters of the neuron. The central idea is that each neuron's parameters can be individually adjusted so that the network can exhibit some desired behaviour. These parameters are adjusted during the training phase of the neural network design process. [42]

Figure 13 shows the simplest form of artificial neuron. In more complex cases there can be multiple inputs that are all scaled by different weights. These weighted inputs are summed with the bias and this summation is then passed through the activation function to transform the weighted sum into the final output of that particular neuron.

Transfer functions

The transfer function is also sometimes known as the activation function. The weighted sum of the inputs is passed into the transfer function which transforms the weighted input into an output. A variety of transfer functions have been used in neural networks. While linear transfer functions have been used, they are not very useful because the output is simply proportional to the input. Usually the transfer functions are non-linear. Sigmoid and hyperbolic tan (tanh) are examples of common non-linear transfer functions. These functions are popular because their derivatives are continuous. The most common transfer functions are shown below:



Name	Function $y=f(x)$	Derivative $\partial y / \partial x$
Logistic	$\frac{1}{1+e^{-x}}$	$y(1-y)$
Tanh	$Tanh(x)$	$1-y^2$
Gaussian	$e^{-x^2/2}$	$-xe^{-x^2/2}$
Linear	x	1

Figure 14: Examples of various activation functions and their derivatives. Image taken from http://condor.depaul.edu/

3.5.3 Neural Network Design Workflow

When implementing a neural network, the design process can be broken down into four major steps as outlined in [42]:

i. Develop dataset

This usually occurs before the neural network design process has started. It involves researching the application and determining the important variables. Usually a trade-off is made in deciding the number of training samples as well as the number of variables. Generally the more samples the better the results of the neural network will be. The variables chosen can also be analysed using Principal Component Analysis (PCA) techniques in order to determine if all of the variables are important to the application. PCA can also be useful in reducing the dimension of the dataset in order to save processing power and training time of the network.

ii. Create and Configure Network

This involves choosing the architecture of your neural network, which involves determining the number of layers to use, the number of hidden neurons and initialising biases and weights.

iii. Train and Validate Network

Once the network has been configured it must be trained with the available data. The data must be broken up into training, validation and test data. Once this is done the training algorithm must be chosen and then the network is trained. To validate your network various performance metrics can be considered in order to determine its effectiveness.

iv. Use Network

After the network has been validated if can then be implemented for its necessary application, whether it is classification as in the case of this research project or for some other application.

3.5.4 Architectures

When designing a neural network system there are many architectures that can be used and their characteristics differ depending on how the network is set up.

i. Single Layer

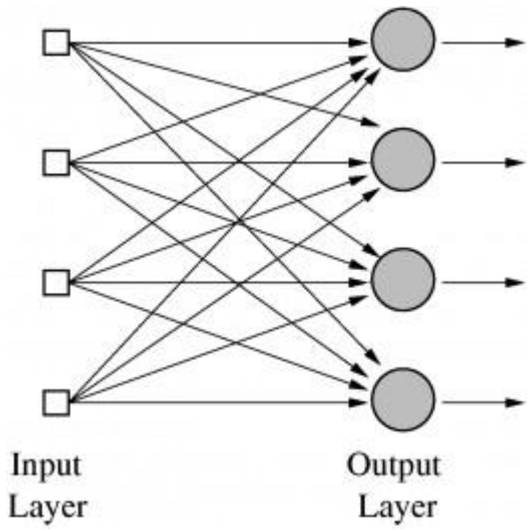


Figure 15: Single layer neural network taken from http://ansonabey.hubpages.com/hub/Artificial-Neural-Network

A single later architecture is the simplest form that any neural network can take. It consists of two components, the input node and the output neurons. The input nodes do not count as a layer as there is no computation done at these nodes. The inputs travel from the nodes into the single layer of output neurons and only move in that direction. The information cannot move backwards through the system, hence it is known as a feed-forward type system. [42]

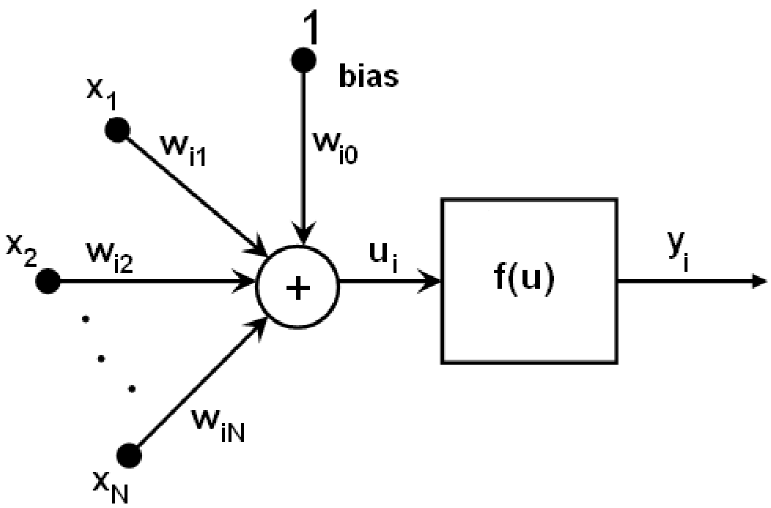


Figure 16: Multiple inputs into a single node. Image taken from http://docs.opencv.org/modules/ml/doc/neural_networks.html

At each node a calculation must be made in terms of the weights and then the activation function. The general equation used to calculate the output at each node of a single layer neural network is as follows:

$$u_i = \sum_j (w_{i,j}^{n+1} \times x_j) + w_{i,bias}^{n+1} \tag{9}$$

$$y_i = f(u_i) \tag{10}$$

Where,
n is the layer number
j is the number of inputs into the current node
i is the number of the current node in the layer
f is the activation function used for the current node

ii. Multiple Layers

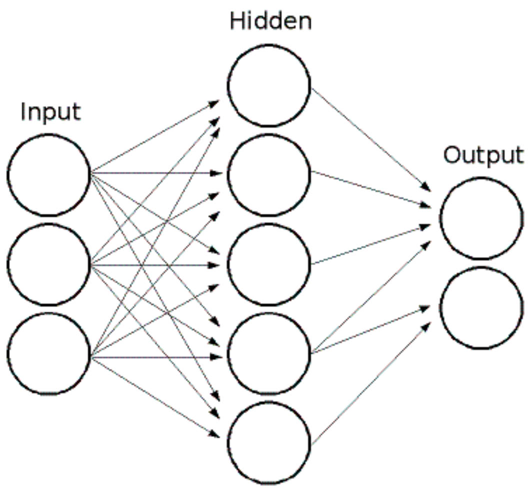


Figure 17: Multiple layer neural network with one hidden layer. Image taken from http://docs.opencv.org/_images/mlp.png

When single layer networks are cascaded, it forms a multi-layer system. This gives rise to the concept of the hidden layers. The reason that they are called hidden layers is because the effect of the hidden layer is not seen at the input or the output. The function of the hidden layer is to intervene between the input and output in some useful manner.

The multi-layer system works in a similar manner to the single layer. The input nodes pass the data into the first layer of neurons; the neurons transform this data and output some values. These output values become the inputs to the next layer. This process repeats until the data reaches the last layer of neurons, known as the output layer. In order to save time when describing a neural network, a naming convention was developed. Each layer is described by the number of neurons in it. For example a 3-10-12-2 network has 3 input neurons, 10 hidden neurons in the first hidden layer, 12 hidden neurons in the second hidden layer and 2 neurons in the output layer. In general the naming convention looks as follows: m – h_1 – h_2- … - h_n – q. Where m is the number of input neurons h_i is the number of hidden neurons in hidden layer I and q is the number of output neurons. The connectivity of the neurons can be described in two ways, fully connected or partially connected. Fully connected neurons connect to all of the neurons in the following layer whereas partially connected neurons only connect to some of the neurons in the following layer. [42]

iii. ***Back propagation Algorithm***

If an input is passed into the neural network, these inputs are passed through the system, from layer to layer, until the final values are calculated at the output layer. This actual output is then compared to the target output value linked with the input sample. The error is calculated using these two values and this error is then propagated backwards until an error can be determined for each node in the network. These errors can be used in order to calculate the partial derivatives. These partial derivatives are used with gradient descent in order to minimise the cost function and update all of the weights of the network.
The following equations were derived by Harris [44] and are defined as follows:

The sigmoid function is the most common transfer function used in neural networks:

$$\sigma(x) = \frac{1}{1 - e^{-x}} \tag{11}$$

The derivative of the transfer function is also required:

$$\sigma'^{(x)} = \sigma(x) - \sigma(x)^2 \tag{12}$$

$$\sigma' = \sigma(1 - \sigma) \tag{13}$$

The error equation is defined as:

$$E = \frac{1}{2} \sum_{k \in K} (O_k - t_k)^2 \tag{14}$$

Where,
O_k is the output of node *k*
t_k is the target value of node *k*

The derivative of the error with respect to the connective weight must be calculated so that it can be minimised. Two cases must be considered; that of the output layer and that of a hidden layer.
Output layer node:

$$\frac{\partial E}{\partial W_{jk}^{l}} = (O_k - t_k) O_k (1 - O_k) O_j \tag{15}$$

Where,
W_{jk}^{l} is the weight from layer *l-1* node k to layer *l* node *j*
This can be rewritten as

$$\frac{\partial E}{\partial W_{jk}^{l}} = O_j \delta_k \tag{16}$$

Where,

$$\delta_k = (O_k - t_k) O_k (1 - O_k) \tag{17}$$

Hidden layer node:

$$\frac{\partial E}{\partial W_{jk}^{l}} = O_i O_j (1 - O_j) \sum_{k \in K} \delta_k W_{jk}^{l} \tag{18}$$

Similarly to the output layer the equation can be defined as:

$$\frac{\partial E}{\partial W_{jk}^{l}} = O_j \delta_j \tag{19}$$

Where,

$$\delta_j = O_j (1 - O_j) \sum_{k \in K} \delta_k W_{jk}^{l} \tag{20}$$

If the bias term θ is incorporated into the equation:

$$\frac{\partial O}{\partial \theta} = O(1 - O) \frac{\partial \theta}{\partial \theta} \tag{21}$$

And

$$\frac{\partial \theta}{\partial \theta} = 1 \tag{22}$$

Therefore

$$\frac{\partial E}{\partial \theta} = \delta_j \tag{23}$$

This holds for any layer l

The weights and bias terms of the network are then updated using the following equation:

$$W + \Delta W \rightarrow W \tag{24}$$

$$\theta + \Delta \theta \rightarrow \theta \tag{25}$$

Where

$$\Delta W = -\eta \delta_l O_{l-1} \tag{26}$$

$$\Delta \theta = -\eta \delta_l \tag{27}$$

3.5.5 Training Algorithms

Once the architecture of the neural network has been decided upon and configured, it must then be trained. The neural network is trained using algorithms that will most effectively minimise the error of the network output. There are two main styles of training a network, incremental and batch training. Incremental training refers to the weights and bias values being updated after each input is put through the system. Batch training works by updating the weights and biases after all of the inputs have been put through the system [42].

The most commonly used classification training algorithms are briefly outlined below:

i. ***Gradient Descent***

This is the most basic algorithm that is used to train a neural network. This algorithm updates the weights and biases of the network by changing them in the direction that will cause the cost function to decline in the greatest manner.

The update equation:

$$W_{k+1} = W_k - \eta_k g_k \tag{28}$$

Where,
W_{k+1} is the vector of the updated weights or bias
W_k is the vector of the current weights or bias
η_k is the learning rate
g_k is the current gradient

The learning rate multiplied by the gradient is what affects the weight or bias value. The only parameter that the user can adjust is the learning rate. The larger the learning rate, the larger the jump is for each update iteration. If the learning rate is set too high it could cause the algorithm to become unstable and diverge. If the learning rate is set too low it could take a very long time to converge.

When the algorithm is far from the minimum it will have a steep gradient and approach the minimum fast but as it approaches the minimum the steps it takes get smaller and smaller until it the algorithm is close enough to convergence and stops.

The gradient descent algorithm can be improved in two simple methods. The first is using a concept called momentum and the other is to have an adaptive learning rate.

Momentum is useful to help the algorithms to get out of local minima as well as to speed up the convergence of the algorithm. Adding momentum will also dampen the oscillations of the algorithm. The momentum concept works in a similar manner to momentum in the physical world. For example when a ball is moving down a slope it will build up momentum and speed up but when it heads up a hill it will lose momentum and slow down [45]. The gradient descent algorithm becomes:

$$W_{k+1} = W_k - \eta_k g_k + mW_k \tag{29}$$

The value of m is between 0 and 1. The momentum constant is usually determined by trial and error. When the gradient keeps pointing in the same direction the size of the steps will increase proportional to m and will decrease proportional to *m* if the gradient changes direction [45].

Adaptive learning can be used to speed up convergence and there are many methods of updating the learning rate. An example is the bold diver algorithm which works as follows:

After each iteration a new error is calculated and compared to that of the previous iteration. If the error has decreased then the learning rate is increased by a small proportion (between 1% and 5%). If the error between iterations increases, then decrease the learning rate by a large margin (roughly 50%) [45].

ii. ***Conjugate Gradient Descent***

In the basic gradient decent algorithm the weights are adjusted in the direction that minimises the cost function the most rapidly. While this method does produce the most rapid decline it does not mean that it is taking the most efficient route to convergence. The Conjugate Gradient algorithm performs a search along conjugate directions. This generally leads to a faster convergence than that of the basic gradient decent algorithm.
The basics on how this algorithm works is as follows. First the steepest gradient descent must be determined. Once this is done a line search is performed along that direction. The next search direction is determined so that it is conjugate to the previous search directions. [31] [42] [46]

iii. ***Resilient Back Propagation***

One of the most common transfer functions used for multi-layer neural networks is the sigmoid function. The sigmoid is often referred to as a "squashing" function. This is because it can take an infinite input range and transform it into a finite output range. Another characteristic of the sigmoid function is that as the input gets very large, the gradient tends towards zero. These small gradients can have a negative effect on algorithms such as the gradient descent because it leads to smaller steps even when the weightings are far from their optimal value. This leads to more iterations being required for the algorithm to converge. To remedy these harmful effects resilient back propagation was created and is commonly known as RPROP.

RPROP fixes this problem by only focusing on the sign of the gradient and ignores the magnitude when updating the weights and bias. The update value is increased by a factor every time the previous and current iteration's gradients have the same sign. When the gradients have different signs then the update value is reduce by a factor. If the derivative is zero then the update value remains the same. [31] [42] [46]

iv. ***Levenberg-Marquardt***

This algorithm was developed with the goal of achieving second order training speeds without computing the Hessian Matrices. Feed forward networks usually have a performance function that has the form of a sum of squares. This means that the Hessian can be approximated:

$$H = J^T J \tag{30}$$

And the gradient can be computed as follows:

$$g = J^T e \tag{31}$$

J is the Jacobian Matrix
e is the vector of network errors

The Levenberg-Marquard algorithm uses the approximation in the following way:

$$x_{k+1} = x_k - [J^T J + \mu I]^{-1} J^T e \tag{32}$$

When the μ scalar is zero then this equation works like the Newton's Method with an approximated Hessian. When μ is large then the algorithm acts like a gradient descent algorithm with small steps. Newton's method is faster and more accurate near a minimum error and so μ is reduced with every successful step μ would increase if the performance is getting worse. [31] [42] [46]

3.5.6 Performance Metrics

Once the neural network has been created and trained its results must be validated and evaluated in order to judge if it gives acceptable results. Two metrics have been considered for this research, these are the Mean Squared Error (MSE) and the confusion matrix.

i. Mean Square Error

This calculation measures the average of the squares of the errors for the neural network. It is a good performance indicator for the neural network as it shows how close the outputs are to the desired outputs. When the MSE decreases the classification accuracy of the network increases.

$$MSE = \frac{1}{N}\sum_{j=1}^{N}(Y - Y_{desired})^{2} \tag{33}$$

Where,
N is the number of samples
Y is the neural network output
$Y_{desired}$ is the desired output of the network

ii. Confusion Matrix

The confusion matrix can be tabulated in a manner that allows for a visual method for judging the performance of a classification algorithm. Each column of the confusion matrix represents the instance in a predicted class and each row gives the instances of the actual class. The confusion matrix can then be used to calculate other performance indicators such as the percentage of errors and number of true positives, true negatives, false positives and false negatives. This can be very useful when evaluating a trained neural network. The diagonal shows the number of correct classification and any numbers that are not on the diagonal shows misclassifications. The table shows which gesture was misclassified as well as which gesture the classification algorithm misclassified it as. The confusion matrix is a useful tool for determining which gestures are misclassified because their data is very similar.

4. Hardware Development

4.1 Specifications

The hardware design is vital to correct functioning of the device. If the system is designed badly, the sensors will not record the data accurately. This means that the gesture database will be flawed and in turn the SASL static gesture classification of glove will fail. Certain specifications were laid out in order to ensure high quality results.

4.1.1 Physical Glove Specifications

The glove must:

- be able to fit multiple users
- be breathable and comfortable for extended use
- not force the users hand into uncomfortable positions
- be simple to put on and remove
- not limit or restrict the users hand in anyway
- be portable and lightweight
- be low cost

4.1.2 Electrical Specifications

The device must:

- be able to communicate wirelessly and/or with a tether
- be able to communicate with a PC or cell phone
- be powered by a battery or be able to make use of a tethered power supply
- have a long battery life to reduce the inconvenience of charging it
- be able to read and send the sensor data real time
- have enough sensors to accurately record static gestures
- be able to expand to dynamic gesture recording and classifying
- be able to record high resolution data for classification purposes
- be repeatable and reliable, giving readings that are on par with, or better than, current available technologies
- be able to calibrate itself to different users
- be simple to use and quick to set up

4.1.3 Safety Specifications

The device must:

- Not have any exposed wires carrying a current of 1mA or greater as per biomedical safety specifications
- Not cause harm to the user in any way

4.2 Initial design approach

4.2.1 Robotic skin based design

The initial data glove design was based on the idea of creating a robotic skin. Unfortunately this idea and design had to be abandoned due to the fact that Eeonyx (the only supplier at the time) went through a major restructuring and were not selling the material required to implement the design. This led to a more traditional data glove design approach.

The robotic skin approach was designed as follows:

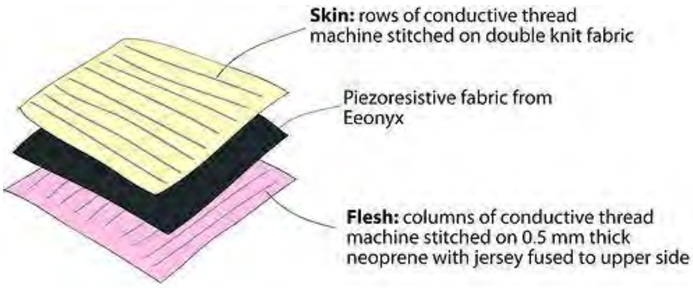


Figure 18: Robotic skin basic implementation based on the rSkin design by Hannah Perner-Wilson (plusea.at)

The skin is made up of three layers, the bottom layer with rows of conductive thread sewn into the material, the next layer is the Piezoresistive layer (Material from Eeonyx) and the top layer has columns of conductive thread that run perpendicular to the bottom layers' rows. The top layer would have a voltage applied to it while the bottom layer would be attached to the multiplexer that connects to the microcontrollers ADC. The top and bottom layer would be placed in such a manner that the conductive thread tracks would overlap/cross throughout the skin. The materials used would allow for stretch so that they do not limit motion. This motion will compress the piezoresistive layer of the skin which will reduce the electrical resistance. This reduction in resistance, in conjunction with a voltage divider circuit, will cause a change in voltage between a crossed area of a row (ADC layer) and column (top layer). This change in voltage would be read on the microcontrollers ADC. Each crossed area would be considered by cycling through each point individually with the use of 4-to-16 multiplexers. We can do this by powering one column and reading a value from one row. This method was developed by Hannah Perner-Wilson in order to create an artificial robot skin (rSkin).

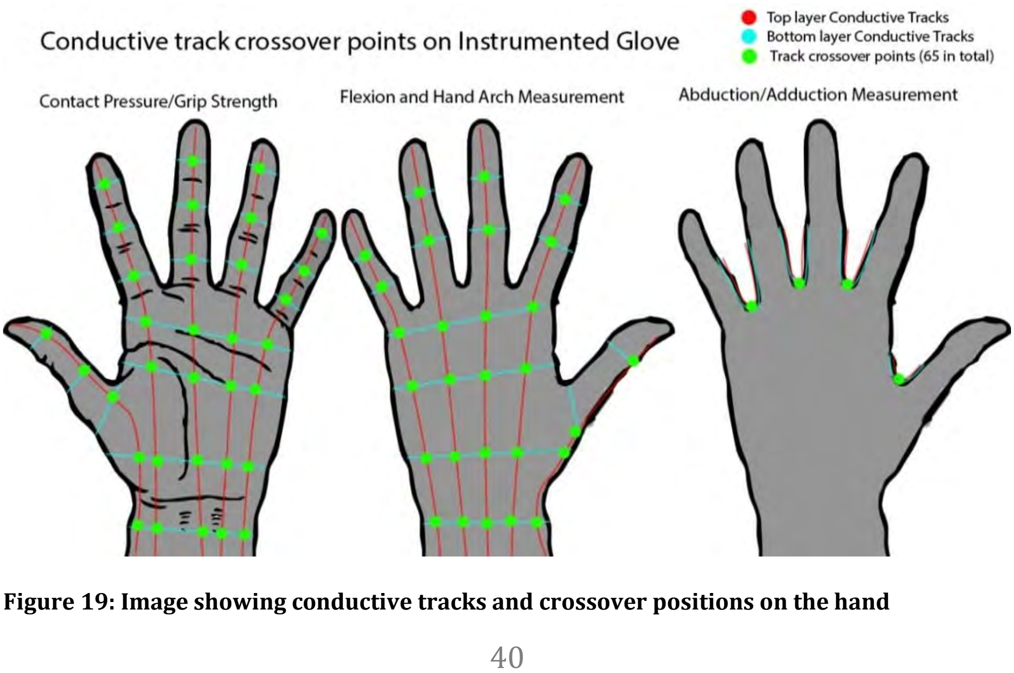


Figure 19: Image showing conductive tracks and crossover positions on the hand

The glove would be created in a similar manner to the basic setup but would be designed to fit a human hand instead of being flat, for table top use or in a tube for the robotic arm. The readings from the crossed over points would determine the angle or position that the fingers, wrist and hand are in. The change in voltage would relate to the change of angle or position. This relationship would be tested and modelled in order to ensure accurate results. Two 4-to-16 multiplexers will be used. This will allow for a maximum of 256 points to measure on the user's hand, of that roughly 65 points will be used.

Each of the green dots shown can be measured individually, there are 65 in total, and therefore the glove would be able to attain a high level of accuracy with regards to the position of the hand. Not all of the points have to be used but they are available if needed. Principal Component Analysis would be used in order to determine which data points were most important. Only the points required for SASL would be used.

4.3 Glove Design Process

During the development of the data glove several iterations were built and were added upon in order to get to the final design. A summary of the iterations is given below:

- The first glove design only had ten flex sensors that were sewn to a store bought glove. After basic tests and gaining a deeper understanding of the flex sensors and how they work, it was realised that the glove would have to be custom made to be a tight fit. This would give the best results. The electronics were on a breadboard at this point.
- The second iteration was hand sewn out of Lycra, this gave a tighter fit which led to better sensor readings. The hand sewn approach was very time consuming and due to lack of skill, was not very sturdy. The electronics were still run on a breadboard.
- The third iteration was machine sewn out of Lycra by a professional and this resulted in a superior glove with better sensor readings. The first attempt at creating the contact sensors was done on this glove. The contact sensors worked but were inconsistent and needed improvement. This was also the iteration where it was noticed that an 'R' contact sensor would be needed for the SASL classification.
- The fourth and final iteration included the 10 flex sensors and 4 contact sensors. The sensors were secured well, worked accurately and gave repeatable results.

After the final glove design had been constructed, the electronics had to be moved off the breadboard and onto a dedicated circuit board. The first design worked well but it was cumbersome and uncomfortable. This was unacceptable for a portable device and so the circuit board was redesigned. This led to a more compact design that fit easily on the back of the glove and did not hinder the users activities.

4.4 Hardware Evaluation

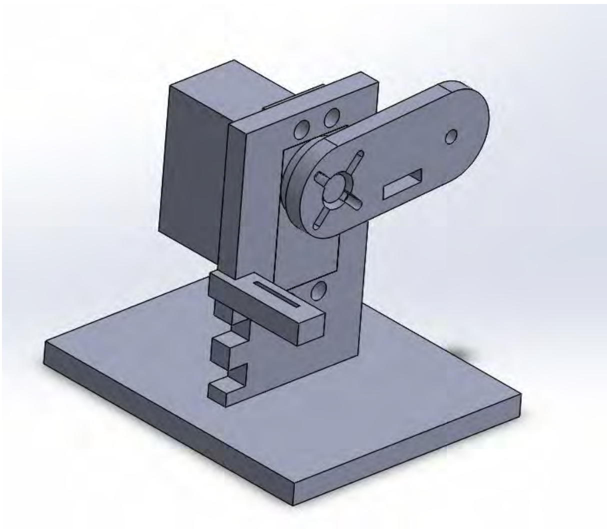
4.4.1 Sensor Characterisation

In order to use the sensors in the most effective manner their characteristics must be understood. The sensors were put through a variety of tests in order to make sure these characteristics are well understood. Some researchers opted to modify existing sensors in order to improve the readings. The sensors were put through the testing process to determine the necessity of the modifications to the flex sensors.

4.4.2 Testing Rig

For the purpose of testing the flex sensors, a basic test rig was designed using SolidWorks. The components were laser cut out of Perspex and glued together. The rig was made up of a base, stand, a slot, a 'finger joint' and a servo motor. The servo motor is mounted on the stand and the 'joint' was connected to the servo. The test rig was made to mimic a simplified finger flexing and the 'joint' was rounded in order to simulate the curvature of a MCP or DIP joint.

The flex sensor can be attached to the 'joint' as it would be attached to a finger. The sensor is placed in the slot and when the servo moves, the sensor is forced to bend along the 'joint'. The servo motors range of motion was set to move between 0 to 90 degrees, to simulate the restrictions of the finger. The servo motor was controlled using an Arduino Microcontroller. A series of tests were designed in order to characterise the flex sensors for use in the glove. The same circuitry that was used to test the sensor was used in the data glove. The data was recorded and logged using the Arduino microcontroller.



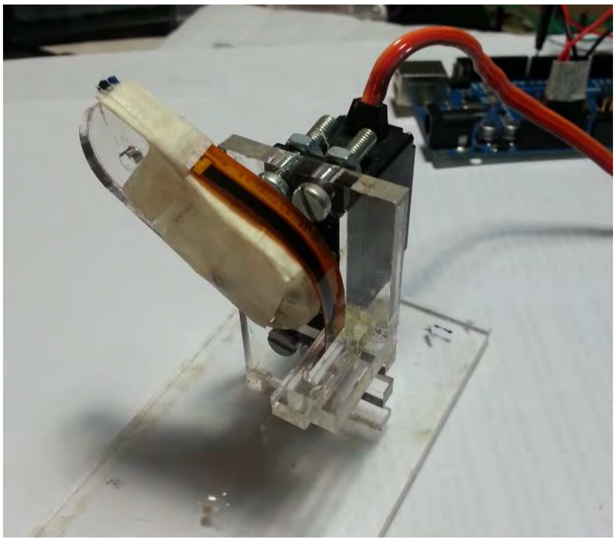


Figure 20: Left - Solidworks model of the test rig. Right - Flex sensor mounted on test rig

4.4.3 Flex Sensor

i. Step Response

The step response served as a measure of how quickly the resistance of the sensor could change, how quickly it settled and how the sensor decayed over time. The largest required deflection was used as the basis of this test. The sensor was moved from 0 to 90 degrees deflection. The Flexpoint flex sensors' resistance increases proportional to the deflection angle. This means that the results for this test will start at a certain value and drop once the step has been introduced.

The test rig was set to 0 degrees deflection and then the flex sensor was mounted on it. The test rig was then set to move to 90 degrees deflection as fast as it could. Once the 90 degree deflection was reached, the sensor was left in its flexed state to record the decay of the sensor over time. The sensors signal was recorded for the entire event using a voltage divider and a 10-bit ADC that is built into the Arduino Mega. This circuitry is the same as that used in the data glove. This test was repeated five times for five different sensors in order to determine a general response time.

The test was then repeated in reverse; the sensor was set at 90 degrees deflection and was rotated to 0 degrees deflection in order to determine if the rate of flexion was the same as extension.

Figure 21 shows the average step response of one of the sensors during the test. This particular sensor had an average reading of 500 on the ADC at 0 degrees deflection and a reading of 380 on the ADC at a deflection of 90 degrees. This means that it had a range of 120. At 0 degrees deflection the sensor showed little noise and was stable. At a deflection of 90 degrees there was some noise that affected the reading but it was minimal (maximum fluctuation <0.05%). The data shows that the sensor has an

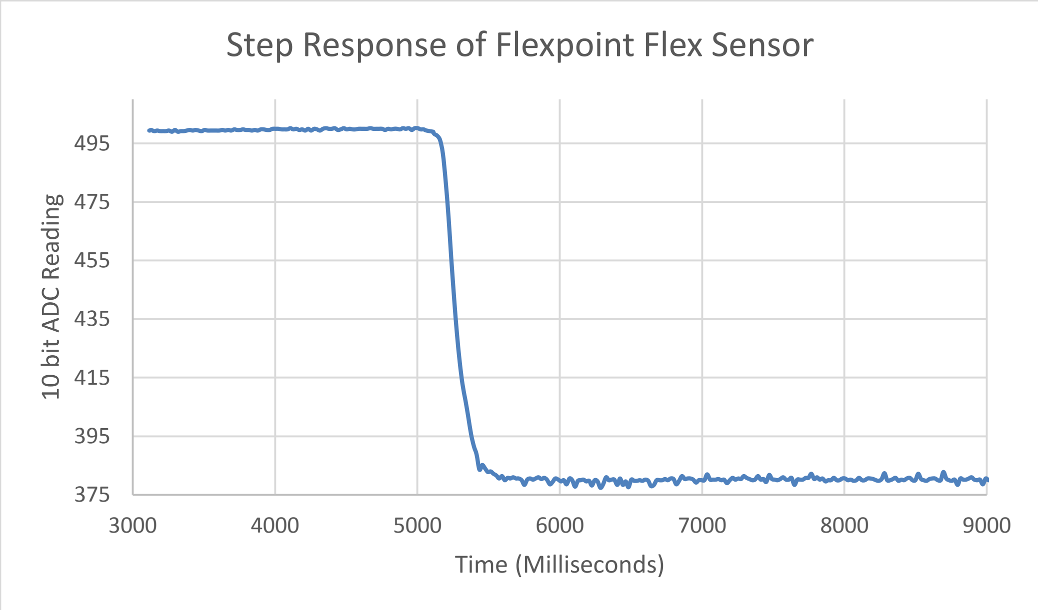


Figure 21: Average step response of the Flexpoint flex sensor

average settling time of 316 milliseconds for deflection and 277 millisecond settling time for extension. The decay of the sensor over the 3 seconds it was held in that position was also negligible, less than 0.05%. While it has been shown that these sensors have larger decay rates over longer periods of time, this will not be a problem for the intended purpose of the glove. The static gestures studied in this research will not need to be held longer than three seconds in order to record or classify them.

The rest of the sensors' averages for the step response test have been tabulated below:

Sensor Number	1	2	3	4	5	Average
Value at 0 degrees deflection	488	510	500	489	476	492.6
Value at 90 degree deflection	394	410	380	358	385	385.4
Range	94	100	120	131	91	107.2
Maximum noise fluctuation	11.2	5.2	5.6	8	5.6	7.12
Deflection settling time	246	286	316	286	277	282.2
Extension settling time	257	298	277	317	238	277.4
Decay over 3 seconds (%)	0.119	0.052	0.047	0.061	0.062	0.068

Table 4: Summary of the results from the step test conducted on the spectra symbol step test

The table shows that the sensors have a good step response as they have very short settling times, the noise fluctuations and decay rate at maximum deflexion are negligible. These results show that the Flexpoint flex sensors are ideal for the use in the data glove.

ii. Linearity

The linearity of the sensor refers to the change in sensor resistance in relation to the change in bend angle. The sensor resistance can be converted to voltage by using a voltage divider circuit. The linearity of the sensor was determined by placing the sensor on the test rig and recording the sensor signal at various angles of deflection.

The sensor was mounted on the rig and set to 0 degrees deflection. The test rig was rotated to a deflection of 5 degrees, settled for 1 second and then then sensors output was recorded. The sensor was then moved back to 0 degrees of deflection. The same process was repeated in increasing intervals of 5 degrees until the sensor was moved to 90 degrees of deflection. This test recorded at 18 different positions and its associated output. The entire process was repeated 5 times with 5 different sensors in order to determine the average linearity of the sensors. Once the linearity was established, the need for modification 2 was considered.

The repeatability of the sensors were also determined using the data from this test. The standard deviation of the results were calculated to determine whether a resolution of 15 degrees could repeatedly be achieved.

The data from each of the sensors was normalised, averaged and visualised in Figure 22 below. The graph also displays the standard deviations of the results at various deflection angles. Overall the sensors show a linear relationship. A nonlinearity can be seen at the smaller deflection angles (0 – 20 degrees) but this is not an issue as the resolution specification set out in this study only requires a resolution of 15 degrees. The average standard deviation for the sensors is 0.05 (5%). It can clearly be seen that the sensors can achieve this resolution as none of the standard deviations overlap with other 15 degree intervals.

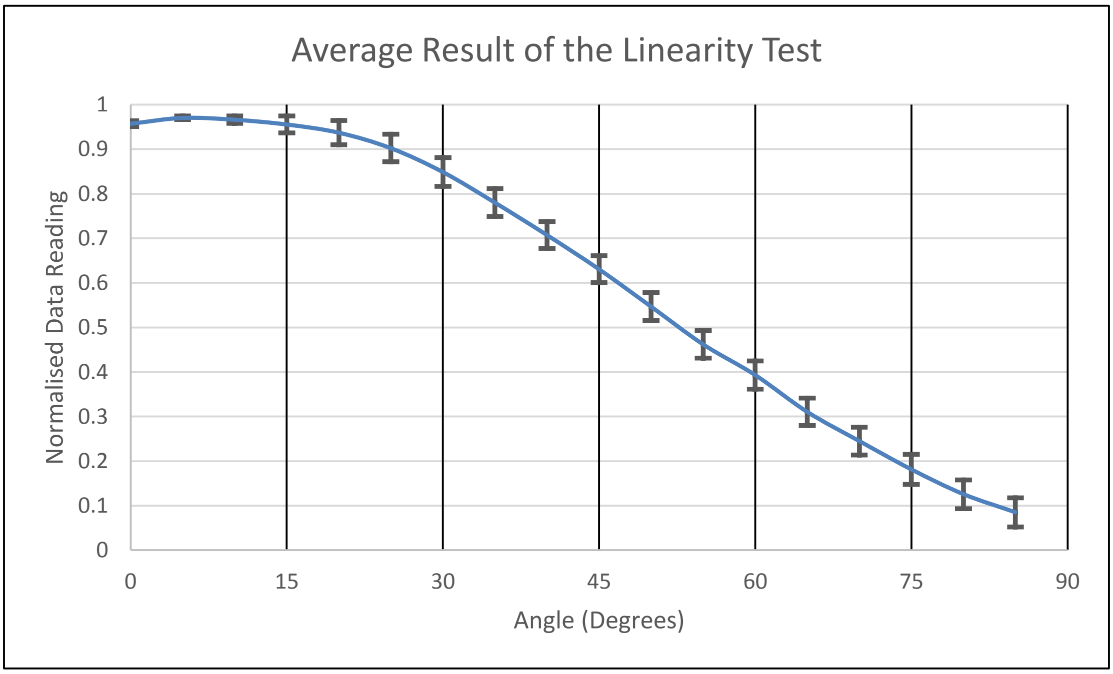


Figure 22: Average results of the linearity test with standard deviation shown

iii. ***Evaluation and Modifications***

The resistive sensors evaluated were the 2 inch flex sensors with over-laminate made by Flexpoint (Draper, UT, USA). Gentner and Classen [47] suggest two modifications in order to improve the results given by these sensors. They used the same brand of flex sensor but a slightly different model. The sensor they evaluated had no over-laminate while the flex sensors used in this study do have the over-laminate.

Modification 1: A thin unplaticised polyvinyl chloride (PVC) foil was fixed over the carbon layer of the front side of the sensor. The PVC had a thickness of 0.2mm and a width of 4mm. This was done in order to enhance stability of the sensor signal.

Modification 2: A resistor was placed in parallel to each flex sensor in order to improve the linearity of the sensor readings.

For the purpose of this study neither of the modifications mentioned were implemented.

Modification 1 does not need to be applied as the decay rate is negligible for the duration required for the recording and classification of static gestures. The over-laminate could be potentially be filling the role of modification 1. There are no specifications given about the over-laminate and so this cannot be confirmed.

While modification 2 may be useful in order to slightly improve the results of the linearity test, the sensors managed to achieve a greater resolution than was specified for this study and so it is unnecessary to implement the modification.

4.5 Glove Reliability and Repeatability

A test was developed by Wise *et al.* [48], and later expanded by Dipietro *et al*. [18], and became the standardised protocol for determining the repeatability and reliability of data glove devices.

The test was broken up into four separate tests which involved the participant placing their hands in a known grip and a flat position, as well as varying whether the glove must be removed or consistently kept on between tests. The main focus of the test is repeatability and reliability of the data glove. The data was recorded using the 10bit ADC that was available on the Arduino Pro Mini microcontroller. The data was sent to a PC via Bluetooth and recorded at a Baud Rate of 9600.

A known and unchangeable grip position is created for each participant. This is done using DAS modelling material and asking the participant to squeeze the modelling material until their hand can fit comfortably in the mould. The DAS would then harden and the participant would be able to grip in a repeatable and natural manner without using force. This is important as an unknown grip force could cause unnecessary measurement errors [18] [48].

Three healthy subjects, two male and one female, performed this series of tests using the prototype. Each participant was asked to sit in a comfortable, natural manner in order to reduce any unnecessary strain that could affect the test. They were then led through all of the tests in order to ensure that they understand all of the processes involved. This is done in order to try and minimise human error during the test.

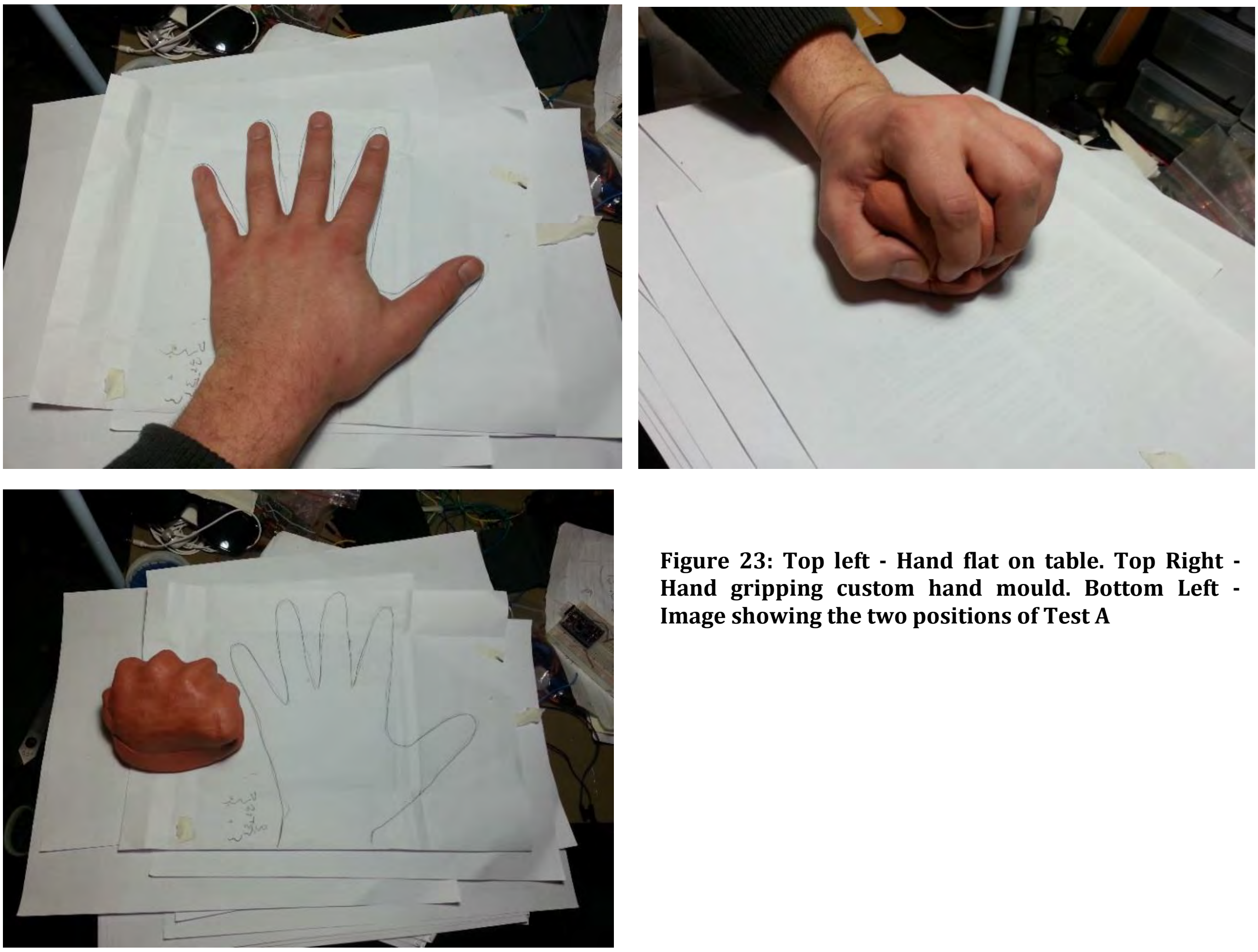


Figure 23: Top left - Hand flat on table. Top Right - Hand gripping custom hand mould. Bottom Left - Image showing the two positions of Test A

4.5.1 Test A

The participant was asked to grip the mould for 6 seconds and then place their hand on the flat table top for 6 seconds; this equates to one set of trial data. This process was repeated 10 times in order to gain one block of data. This in turn was repeated 10 times in order to capture a total of ten blocks of data (100 trial data). The glove remains on the participant's hand for the duration of the test.

4.5.2 Test B

Test B was the exact same process as Test A except the participant must remove and put on the glove again for each cycle (one cycle is two blocks of data). This means that each participant will remove and put on the glove 5 times in total by the end of the test. Removing and putting on the glove every cycle tests to see how it will affect the measurements.

4.5.3 Test C

The participant was asked to place their hand on the flat table within the outline of a drawn hand for 6 seconds and then close their hand into a light fist (maximum flexion) for 6 seconds, and then release their hand and place it back within the hand outline for 6 seconds. As with test A this process was repeated until 10 blocks of data is captured. The glove remained on for the duration of the test.

4.5.4 Test D

Test D is the exact same process as Test C except the participant must remove and put on the glove again each cycle (one cycle is two blocks of data). This means that each participant would remove and put on the glove 5 times in total by the end of the test.

4.5.5 Data Processing

According to the Dipietro et al. [7] the 100 data trials from each participant were stored in a 3D Matrix [X_{ijk}]. The subscripts represented the i-th trial (i = 1,...,10), the j-th data block (j=1,...,10) and the k-th sensor on the data glove (k=1,...,10).

The data was processed and the Range (in degrees), average Range and the Standard Deviations were obtained.

The range of the k-th sensor was obtained as follows:

$$R_k = max_j\left(\overline{X_{jk}}\right) - min_j\left(\overline{X_{jk}}\right) \quad (34)$$

$$\left(\overline{X_{jk}}\right) = \sum_{i=1}^{10} X_{ijk} \quad (35)$$

4.5.6 Previous Results

The following results were reported while using this testing protocol:

Researcher	Range (Degrees)	Standard Deviation
Wise et al [6]	5.6	2.3
Dipietro et al [7]	6.65	2.14
Gentner and Classen [8]	4.96	1.59
Li et al [9]	3.29	1.12
Simone et al [10]	3.36	1.05
Saggio [11]	3.25	1.07

Table 2: Overall test results of various researchers who followed the test methodology set out by Wise et al. [48]

4.5.7 Test Results

An example of the raw data, from Test A, of the sensors has been shown in Fig. 1. The graph shows the participant's hand starting at a flat position, gripping the mould and then back to the initial position. This movement can be seen ten times as the test progressed. The data is separable and shows very distinct peaks and troughs.

The data revealed that the PIP joints have a greater range of values compared to the MCP joints due to the sharper curvature of the PIP joints. The sensors are more stable when they are at a deflection of 0 degrees than when they are deflected to a large angle. The greater the flexion, the noisier the sensor readings become. The variation in the sensor readings is not likely to be caused by a fluctuation in participant grip strength as the tests were thoroughly explained to the participants in detail. There is short time-to-stability and so the data was slightly truncated in order to have a more accurate range calculation as the settling data would skew the results of the averages unnecessarily.

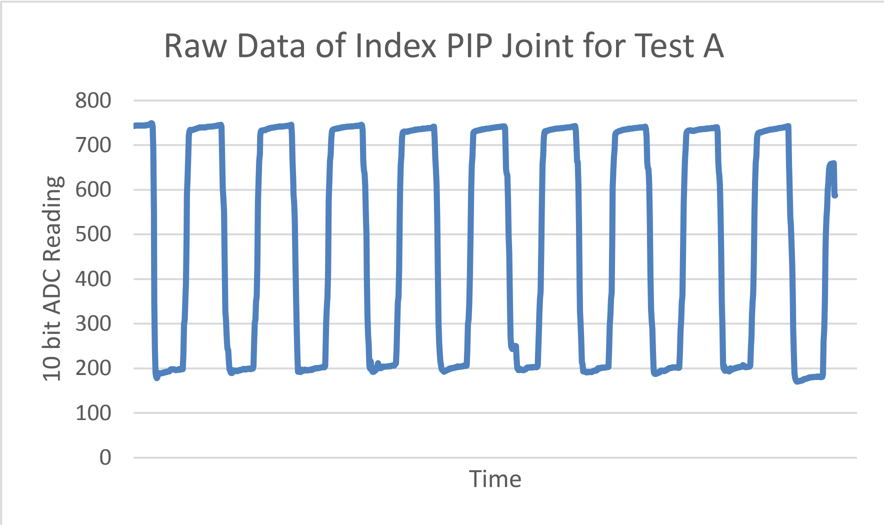


Figure 24: The raw data from the index PIP joint for Test A

The test data, stored in [Xijk], was used to calculate the various ranges and standard deviations and compared to various other researchers in Table 4 shows the results of the four tests for each of the three participants as well as the overall mean. The results from other researchers have also been tabulated for the sake of comparison. The "range" columns show the smallest repeatable angle that the glove can measure reliably. For example if the value is shown to be 3 degrees, this means that the sensor can reliably pick up 30 different positions reliably (90 degrees divided by 3). The SD column shows the standard deviation (in degrees) of the results.

Subject	**Test A**		**Test B**		**Test C**		**Test D**		**Total**	
	Range	**SD**	**Range**	**SD**	**Range**	**SD**	**Range**	**SD**	**Range**	**SD**
Participant 1	1.825	0.644	3.034	1.024	1.655	0.632	2.356	0.919	2.218	0.805
Participant 2	3.056	1.211	2.948	0.990	3.865	1.428	2.969	2.814	3.210	1.611
Participant 3	3.566	1.608	6.460	2.498	3.463	1.588	4.045	1.659	4.383	1.838
Overall Mean	**2.815**	**1.154**	**4.147**	**1.504**	**2.994**	**1.216**	**3.123**	**1.797**	**3.270**	**1.418**
Previous Results										
Wise et al [48]	6.5	2.6	6.8	2.6	4.5	1.6	4.4	2.2	5.6	2.3
Dipietro et al [18]	7.47	2.44	9.38	2.96	3.84	1.23	5.88	1.92	6.64	2.14
Simone et al [49]	5.22	1.61	n/a	n/a	1.49	0.5	n/a	n/a	3.36	1.05
Gentner and Classen [47]	6.09	1.94	7.16	2.26	2.61	0.86	3.98	1.28	4.96	1.59
Li et al [16]	4.56	1.57	n/a	n/a	2.02	4.56	n/a	n/a	3.29	3.06
Saggio [50]	3.76	1.5	5.35	1.69	1.64	0.51	2.26	0.58	3.25	1.07

Table 5: Summary of glove repeatability and reliability as well as results from other researchers doing the same test.

It can be seen that participant 1 had very good results; this is due to the fact the glove was measured and made for their hand. The glove was designed to be tight, so that the sensors would fit snuggly over the joints. Participant 2 had smaller hands and still gave good results. Participant 3 had larger hands and the glove did not fit well but managed to still get better results than a few of the other researchers. This shows that if each of the participants had custom gloves then their results would have been far better than the other researchers. Another approach would be to design three general sizes (S, M, L) and sacrifice some of the resolution.

4.6 Ergonomics

A series of nine questions was given to each participant. This was done as a means of generating feedback from participants on the ergonomics of the glove, see table 3. A rating of 1 – 7 could be given for each questions ranging from strongly disagree to strongly agree.

#	Question
1	I felt comfortable as the glove was put on.
2	I did not feel my fingers were put into any uncomfortable position as the glove was put on.
3	I did not feel a restriction to movement with the glove.
4	I felt comfortable performing the activities in this study.
5	The glove did not feel too tight.
6	I feel like I can bend my fingers just like I can without wearing the glove.
7	The glove did not feel too hot or too cold.
8	I did not feel like my fingers were put into any uncomfortable positions as the glove was removed
9	I felt comfortable as the glove was removed.

Table 6: User Feedback Questionnaire

The answers to the questionnaire have been tabulated below:

Participant	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9
#1	7	7	7	7	7	7	7	7	7
#2	7	7	7	7	7	7	7	7	7
#3	3	5	6	7	7	6	7	6	7
Average	**5.66**	**6.33**	**6.67**	**7**	**7**	**6.67**	**7**	**6.67**	**7**

Table 7: Results of the user feedback questionnaire

In general it can be seen that the participants did not have any issues with the ergonomics of the glove. The results were very positive; the glove was considered comfortable and did not restrict movement. The only issue was putting the glove on participant 3's hand and this is due to the fact that their fingers were wider and this hindered their putting on the glove. During the testing processes involved in this research, the participants had to wear the glove for an extended amount of time and none of them reported any discomfort due to the glove being too tight or hot. There were some reports that the glove felt like it would tear when it was being put on but this problem will be fixed by reinforcing the sewing when the device becomes a final product. Overall the ergonomic considerations for the glove prototype were deemed successful.

4.7 Hyperextension

Generally when people open their hands, their fingers can hyperextend, the amount varying from person to person. This can be a problem for the flex sensors as they are unidirectional (the resistance only varies for flexion in one direction). Fortunately the chosen Flexpoint sensors do continue to deliver the correct readings but the readings become less linear the greater the hyperextension.

4.8 Testing Protocol

There has been criticism directed at the data glove evaluation method developed by Wise *et al*. and later improved by Dipietro *et al*. The process has been called limited as well as not very precise due to external factors influencing the readings of the sensors. The main issue is the human factor that can affect the results of the tests. An example of this is that of grip strength which can vary the angle of the fingers. Steps have been taken in order to reduce this error by instructing the participant to try to keep their grip as consistent as possible as well as using customised grip moulds in order to stabilise the participant's hand. These measures help but cannot remove the human factor completely. Despite these criticisms it has been accepted as a valid method of evaluating the glove on a human hand and as a set benchmark to compare various other data gloves [18] [49].

4.9 Participants

Due to limited time and resources the number of participant was restricted to three for the evaluation of the data glove. This is not ideal as more participants would give greater confidence in the results achieved by the glove. While the results achieved were good, they should be validated in future by running the test again with more participants to ensure accurate results.

5. Final Glove Design

5.1 System Overview

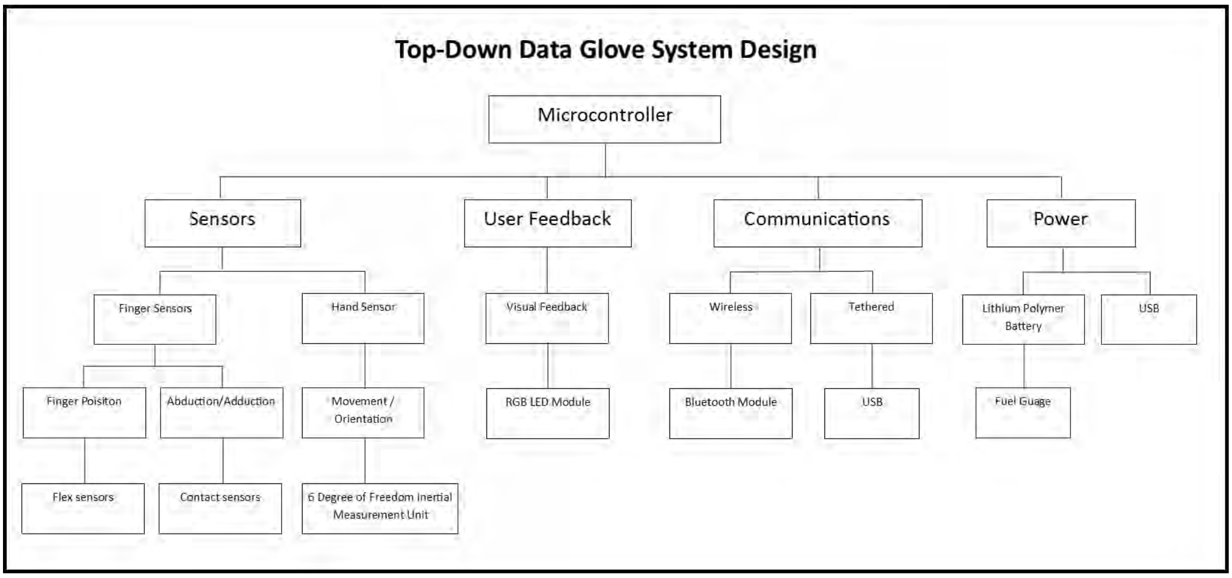


Figure 25: Top-Down Data Glove Block Diagram

A top down design approach was taken in order to determine all of the systems needed to develop the data glove. The diagram above gives an overview of all of the systems related to the data glove. Each of these systems were then prototyped on a breadboard separately in order to ensure that they worked. Once they worked correctly apart, they were added to one system.

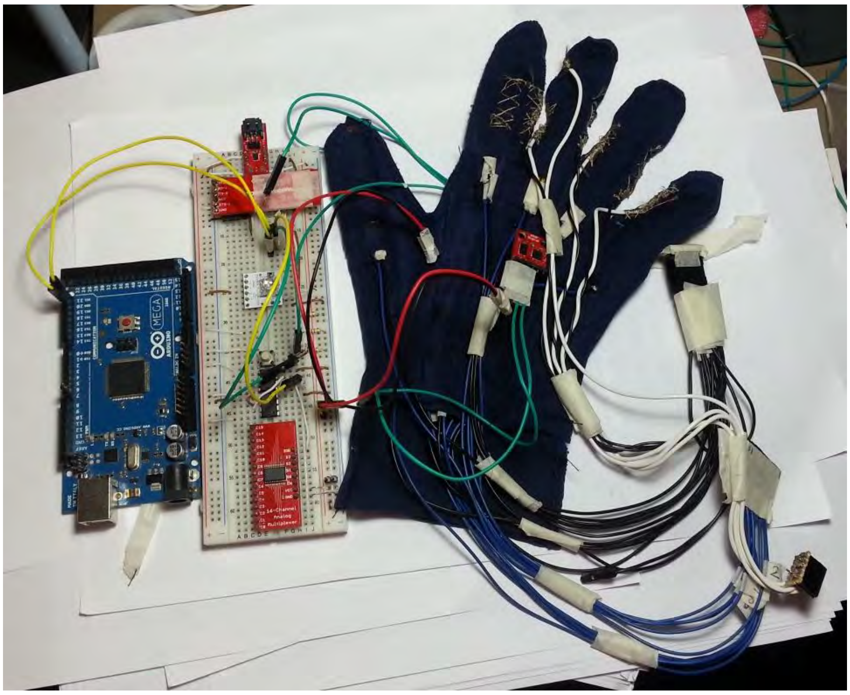


Figure 26: Breadboard Prototype

5.2 Glove Design

The glove (Figure 26) is made from Lycra and constructed to fit the user's hand. Lycra was chosen as it is a very robust material and has been used for many data glove devices. Lycra can stretch and does not lose its shape over time. This means that the glove will be able to fit on multiple hand sizes while not losing accuracy by becoming stretched.

The flex sensors are mounted on the dorsal side of the fingers using pockets that hold them close to the user's fingers. There are abduction/adduction sensors sewn to the side of the fingers and a contact sensor sewn on the dorsal side of the index finger. The wires of these sensors are carefully sewn to the glove in order to keep the glove neat but care is taken to ensure that the sensors are not affected and user's hand is not restricted by the wires. The electronics must be as compact as possible in order to make sure that the device is light and portable. To do this the rest of the circuitry has been soldered on Veroboard and connected in a manner that minimised the size of the overall control unit. An Inertial Measurement Unit (IMU) is mounted on the centre of the back of the glove. Care was taken to ensure that the glove could be put on and taken off easily. The glove is also designed so that it would not restrict movements as well as not be uncomfortable to the user especially over long periods of time. The end goal of this device is for the user to use it on a day to day basis.

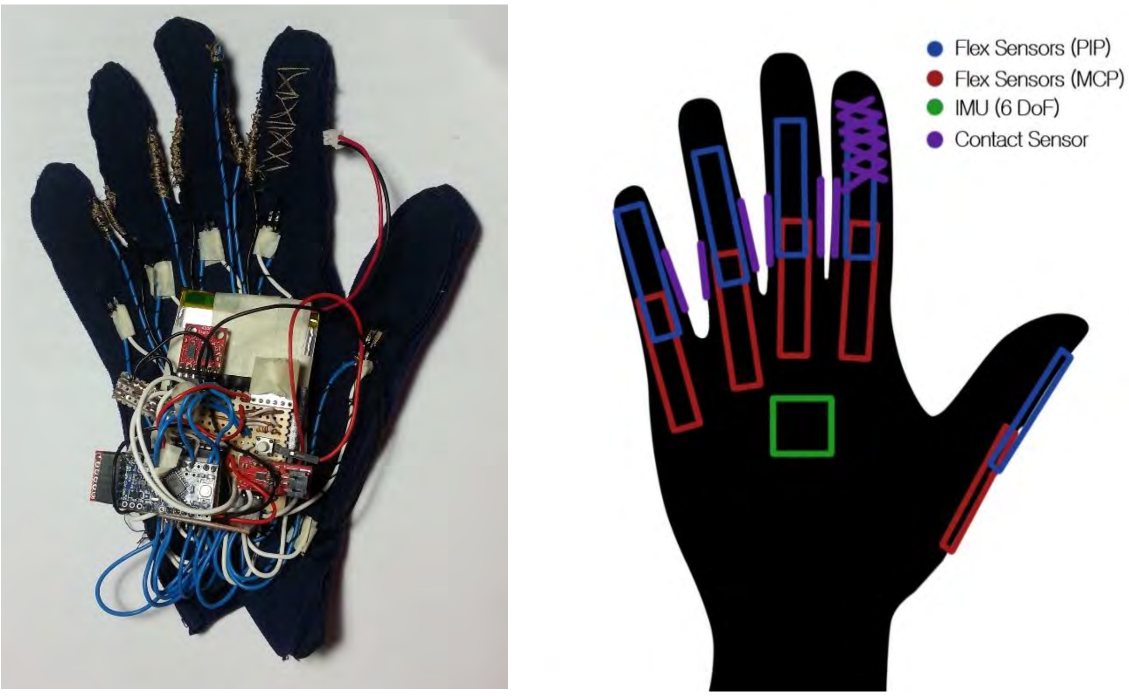


Figure 27: Left - Final Prototype. Right - Simplified Sensor Diagram

5.2.1 Device Dimensions

The control unit houses all of the electronics of the data glove device. The control unit's dimensions are 55 x 50 x 35 mm and weights 40g, not including the battery. Space has been made so that the battery can be mounted underneath the control unit. The battery's dimensions are 5.8 x 54 x 60 mm and weighs 36g. The entire device, glove and electronics, weighs 150g. The control unit would be able to be made smaller by using a custom PCB instead of using several breakout boards.

5.2.2 Microcontroller

The microcontroller chosen for the prototype is the Arduino Mini Pro as it is compact, available, affordable and light which is ideal for wearable devices. It has the necessary modules to power and control the devices circuitry as well as read the sensors. It has a 10-bit analogue to digital convertor (ADC) which allows for up to 1024 different values to be read. It also has serial peripheral interface (SPI), inter-integrated circuit (I2C) and serial communication capabilities which are used to communicate with the digital potentiometer (MCP4105), 6 degrees of freedom (DOF) IMU and Bluetooth module respectively. A 16-to-4 multiplexer (CD74HC4067) is used to ensure that all of the sensors could be read as the Arduino Mini Pro does not have enough analogue pins for all of the flex sensors required.

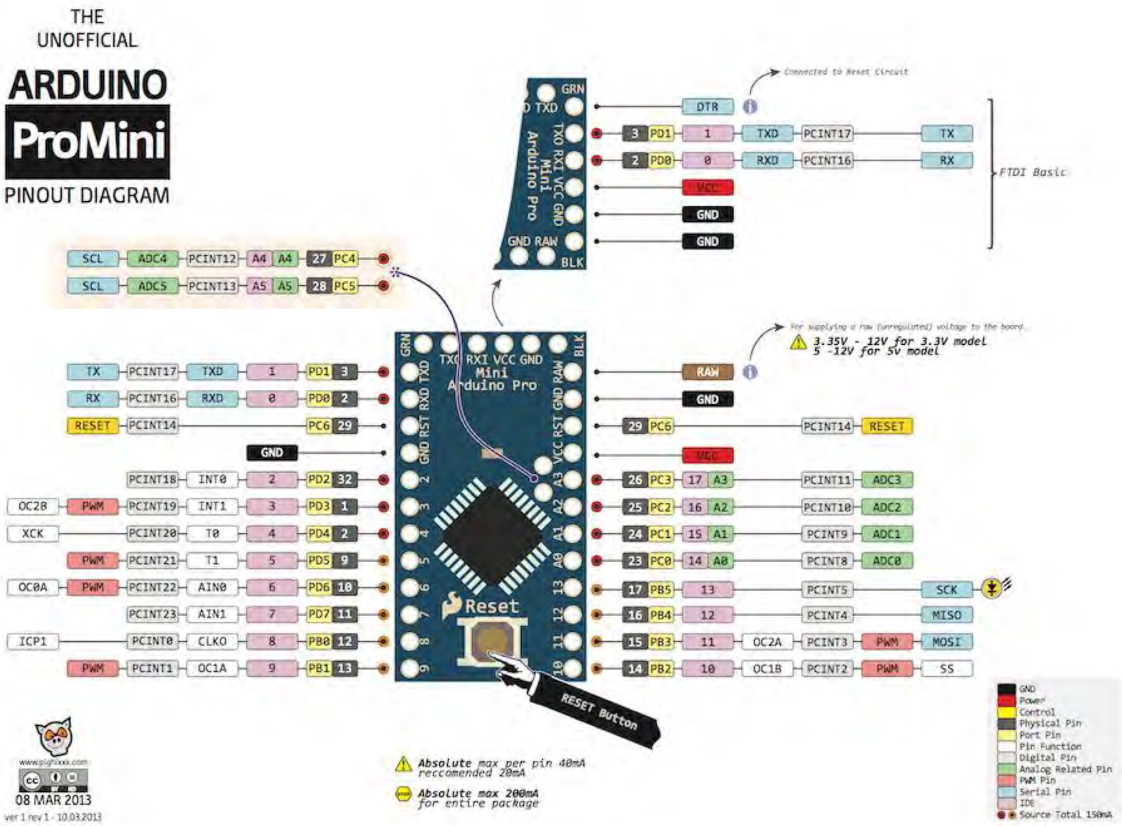


Figure 28: Pin out diagram of the Arduino Pro Mini showing all of the available peripherals (pighixxx.com)

5.2.3 Communications

The data glove uses the Bluetooth Mate Silver to transfer sensor data. This Bluetooth module is low power and has a range of 20m. The Bluetooth module communicates with the microcontroller using serial communications. This makes the implementation of the wireless communications very simple. It is able to send data at speeds up to 115200 baud. Bluetooth was chosen so that the user can wirelessly connect the device to a PC or to their smartphone, which could then use the data via a cell phone application or through a PC program.

Figure 29 : Bluetooth Mate Silver size comparison and labelled diagram (sparkfun.com)

5.2.4 Flex Sensors

Flex sensors were chosen to measure finger flexion over other sensors due to their availability, cost and simplicity to implement. Each flex sensor is used in conjunction with a voltage divider circuit, the digital potentiometer acts as the other resistor. The resulting voltage is then fed into a buffer circuit and finally read by the ADC on the microcontroller. The flex sensors also have valuable mechanical, electrical and time-to-stability characteristics [51] [52] [53]. Flexpoint flex sensors were chosen over the Spectra Symbol flex sensors as they are roughly 30% cheaper per sensor, impede movement less and are more robust due to the fact that they have a Polyimide laminate which protects the tracks from wear and tear.

The circuit for the flex sensors is very simple. The flex sensor is put in a voltage divider circuit with a digital potentiometer (which is used to calibrate and control the circuit). The result of the voltage divider is input into a buffer circuit which then feeds into one of the Arduino analogue pins. In order to save costs on components and reduce the number of pins required for the device, the voltage divider output is fed into a multiplexer circuit. The multiplexer output feeds into the buffer circuit which in turn feeds into the Arduino analogue pin.

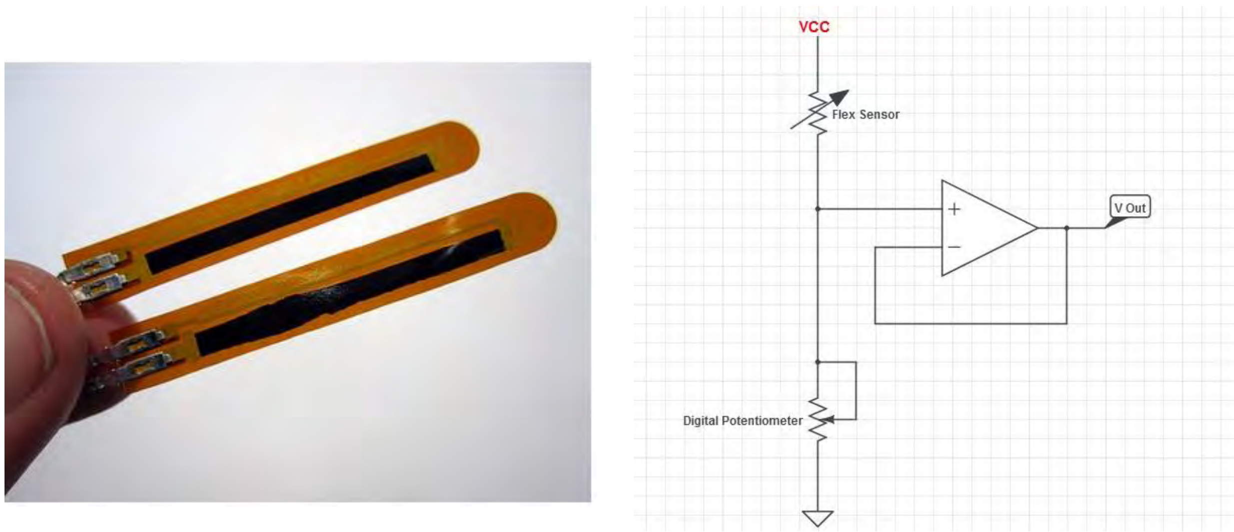


Figure 30: Left - Flexpoint Flex Sensors (plusea.at). Right - Flex Sensor Circuit Diagram.

5.2.5 Abduction/Adduction Sensors

In the interest of keeping costs low the abduction/adduction sensors were created by sewing conductive thread between each finger. The sensors work as simple switches, the sensor reads a one when the fingers are touching each other and a zero when the fingers are spread apart.

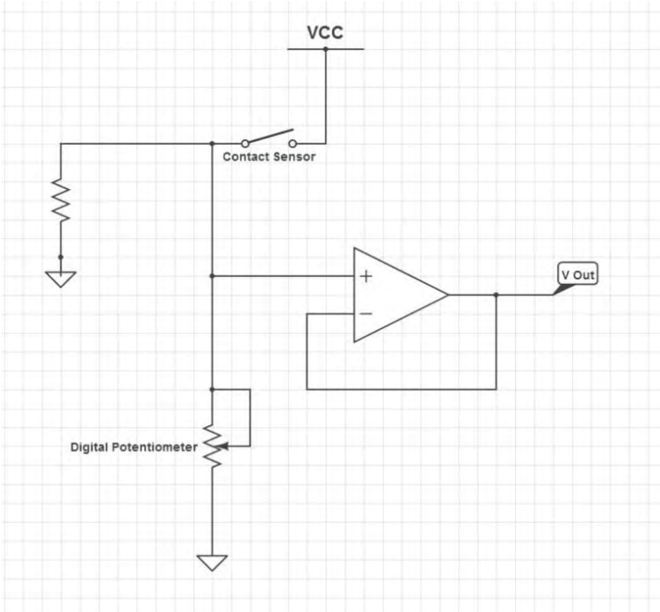


Figure 31: Contact Sensor Circuit Diagram

The circuit for the abduction/adduction contact sensor operated in exactly the same manner as the flex sensor circuit but it has a few minor changes. The sensor acts as a pull-up switch. When the contacts touch a voltage value is read by the circuit and fed into the Arduino analogue pin. When the contacts are apart the circuit is pulled to ground and a value of zero is read on the pin. The circuit is pulled to ground when the contacts aren't touching to ensure that the contacts aren't floating and that arbitrary values aren't read on the Arduino analogue pin.

5.2.6 Digital Potentiometer

An 8-bit 50kΩ digital potentiometer (MCP41050) is used to calibrate the voltage divider, used for the flex sensors. This was done in order to get the largest range of readings from the flex sensors. Using the digital potentiometer allows the device to be recalibrated to different users if necessary, instead of using a fixed bank of resistors. The integrated circuit (IC) is set using SPI and is controlled by the Arduino Pro Mini. The digital potentiometer can output 256 values (increments of roughly 195Ω until it reaches 50kΩ).

A method to ensure that the voltage divider had the output voltage with the largest dynamic was set out by Saggio [11] and used here because it is the most efficient means to calculate the resistor value needed. The R_{ref} was chosen as follows:

$$\frac{V_{out}}{V_s} = \frac{R_s}{R_{ref} + R_s} \tag{36}$$

R_s being the sensor being measured

$$\frac{\Delta V_{out}}{V_s} = \frac{R_{s\_max}}{R_{ref} + R_{s\_max}} - \frac{R_{s\_min}}{R_{ref} + R_{s\_min}} \tag{37}$$

$$\frac{\delta}{\delta R_{ref}} \frac{\Delta V_{out}}{V_s} = \frac{R_{s\_max}}{(R_{ref} + R_{s\_max})^2} - \frac{R_{s\_min}}{\left(R_{ref} + R_{s\_min}\right)^2} = 0 \tag{38}$$

$$R_{ref} = \sqrt{R_{s\_max} \times R_{s\_min}} \tag{39}$$

R_{s_max} = Resistance at 90 degree flexion
R_{s_min} = Resistance at 0 degree flexion

This is the formula used in order to automatically calibrate the device to its current user.

5.2.7 Inertial Measurement Unit

A 6 DOF IMU is used to determine the orientation of the user's hand. It uses a 3 DOF accelerometer and a 3 DOF gyroscope to do this. The data is transferred to the microcontroller via I2C and the orientation was calculated using an open source library called FreeIMU (varesano.net). There is a drift in the yaw axis over time but it is not a large issue as it can be easily reset by the user as necessary to keep measurements accurate for the various exercises. Tests will be carried out in order to determine the significance of the drift. While this sensor was added to measure the angle of the user's hand for static gestures, it can also be used in order to expand the device for use in dynamic gesture classification.

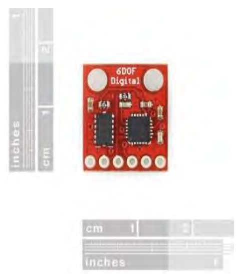


Figure 33: 6 Degree of Freedom IMU (sparkfun.com)

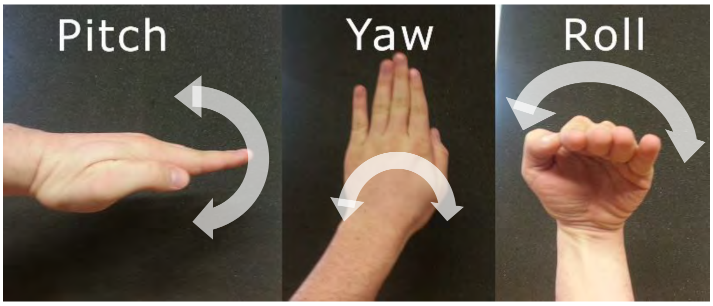


Figure 32: The movement of the hand with regards to pitch, yaw and roll.

5.2.8 Battery

A Polymer Lithium battery (2000mAh) was chosen to power the device as it will give the device a long battery life, it has a good power to weight ratio and it is rechargeable which is necessary for portable devices. While all of the peripherals and communications are running the device has an average current usage of 130mA. This means that the device has a battery life of roughly 14 hours which is ideal since the device can be used for an entire day between charges.

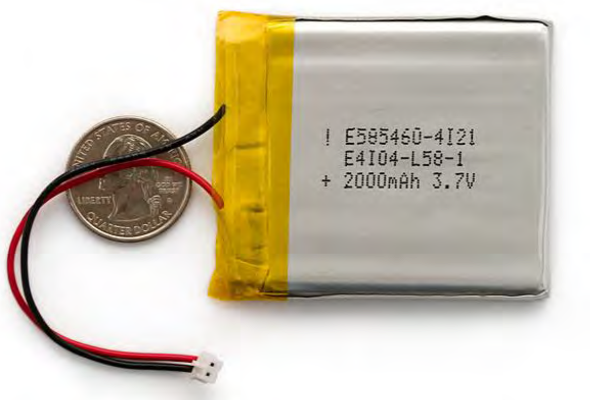


Figure 34: Polymer Lithium Ion Battery - 2000mAh (sparkfun.com)

5.2.9 Visual Feedback

The prototype makes use of the BlinkM RGB LED module. This module is useful for giving simple visual feedback to the user. This module is very useful as it is completely controlled using I2C and comes with a complete code library which makes it very simple and quick to implement. Using an I2C controlled module helps to save on the number of pins required and can allow for using a smaller microcontroller. The RGB LED was used to inform the user which mode the device is currently in and to alert the user to the status of the device. Low battery, wireless communication errors and loss off data confidence can be displayed using different colours. The current mode of the device can also be displayed. Modes such as sleep mode, calibration mode or active mode can be displayed for the users benefit.

Figure 35: BlinkM RGB LED Module (sparkfun.com)

5.3 Cost of Prototype

The final product will cost less than the prototype due to mass production and the economics of scale. Creating a dedicated custom board would remove unnecessary functionality and components from the costs. An example of this would be the cost of the IMU; purchasing the breakout board costs $39.95 whereas getting a surface mount chip such as the MPU-6000, which would be designed into a custom PCB board, only costs $12.95.

While the device is still quite costly at $201.50 per glove, it is still a lot cheaper than the available commercial gloves. Some of the more popular commercial gloves can cost over $800 just for the entry level devices (vrealities.com). According to Simone *et al*. [49], a 5 Sensor DataGlove would cost $2490, a 22 Sensor Cyber Glove would cost $14000 and their own Shadow Monitor would cost $300. This means that although the prototype developed in this research gave very similar results (the prototype had a slightly worse overall standard deviation on average) as the Shadow Monitor, it is roughly $100 cheaper to make. A breakdown of the costs to build a single glove prototype is tabulated below:

Component	Quantity	Price ($)	Total ($)
Arduino Pro Mini	1	9.95	9.95
Digital Potentiometer	1	2	2
6 DOF IMU	1	39.95	39.95
4 – to – 16 MUX	1	4.95	4.95
Flexpoint Flex Sensor 2"	10	7.25	72.50
Bluetooth Module	1	39.95	39.95
Lycra Material and Sewing	1	10	10
Conductive Thread	1	8.95	8.95
LM358 Op Amp	1	0.3	0.3
BlinkM Module	1	12.95	12.95
		Total	$201.50

Table 8: Break down of a single data glove prototype costs

5.4 Battery Life

The battery met the specification laid out by achieving roughly a 14 hour battery life. This is suitable because the device will be able to last longer than an average day and can be charged at night. This battery life is superior to that of some of the commercial data gloves that are available. It was reported that the CyberGlove II has a battery life of 3 hours and the Data Glove Ultra Wireless has more than 8 hours [49]. This means that the developed prototype has a significantly longer battery life than that of the commercial data gloves available.

5.5 Safety Specifications

The user comes into contact with two types of exposed wires when using the glove. The first is that of the flex sensors and the second is that of the contact sensors. In both cases the maximum current carried on the wire is well below the safely specification of 1mA. In the case of the flex sensors, the maximum current being carried is 0.5mA and this only occurs when the flex sensor is fully flexed which does not occur for extended periods of time. When the flex sensors are in the neutral state with no flexion the current is 0.2mA. The contact sensors have a current of 5nA when there is no contact and have a current of 0.5mA when they are in contact. Both of these sensors meet the requirement of being well below the safety specification of 1mA.

All other wiring that does not meet the less than 1mA criteria set out in the specification was well insulated so that it would not come into contact with the user of the data glove.

6. SASL Database Creation

6.1 Data Collection

As mentioned previously 31 static gestures were considered and recorded for the SASL database. In order to create a robust database the data was recorded in three different scenarios. The first was a simple static 'snapshot' of the sensor data for each gesture, the second was continuous data readings from a known position to the gesture and then back to the known position again and the third scenario involved continuous data readings from an unknown position to the gesture and then to a different unknown position. The three scenarios are referred to as type 1, type 2 and type 3 data respectively. The data was collected through a set of basic tests. The data glove was tethered to a PC in order to ensure that there is no loss or corruption of data.

Five healthy participants with no disabilities were chosen in order to record the SASL static gesture database. While these participants have no experience with SASL, it was acceptable due to the static nature of the gestures to be recorded. The naturalness of the gestures that comes with experience is not necessary when performing single static gestures. Each participant was briefed on the gestures until they could perform them with confidence. During the process the glove remained on the participant's right hand and the entire test was performed in one session in order to ensure sensor consistency between the gestures.

Participant	Handedness	Age Group	Gender	Comments
1	Left	20-30	Male	Hand fit the glove perfectly
2	Right	20-30	Female	Hand slightly too small for the glove
3	Right	20-30	Male	Hand fit the glove well
4	Right	50-60	Male	Hand slightly too wide for the glove and finger motion slightly limited
5	Right	50-60	Female	Slightly too small for the glove

Table 9: Basic information on the participants used to record the SASL database

6.1.1 Type 1

The participants placed their hand flat on a table with their fingers spread for 3 seconds, moved their hand into the static gesture and held it for 3 seconds. At the end of this 3 second interval one point of data was read from each of the sensors of the glove and then the participant placed their hand in the starting position. This process was repeated ten times. Once the ten repetitions were completed the participant moved on to the next gesture. This entire process was repeated until each of the 31 gestures had been recorded. This data was to be used in order to train the neural networks as it is clean, separable and easily labelled data.

Figure 36: Isolated SASL static gesture 'B'

6.1.2 Type 2

The participants placed their hand, in a known position, flat on a table with their fingers spread. The participant was signalled to begin which meant that they had a 10 second interval in order to place their hand in the gesture hold it briefly and then place their hand flat and open on the table again. The participant could do the gesture at any speed and hold their hand in the gesture as long as they wanted to as long as their hand was back in the known position again by the end of the 10 second interval. The data from the sensors was recorded continuously during the entire 10 second interval. This process was repeated 3 times for each of the 31 gestures.

This data was recorded so that methods of extracting a gesture data from continuous data can be tested and evaluated. Having the participant moving from a known position simplifies some of the problems mentioned in the literature review previously. Type 2 data is closer to a real life scenario than that of Type 1 data because it is continuous data and requires an algorithm to extract only the relevant data.

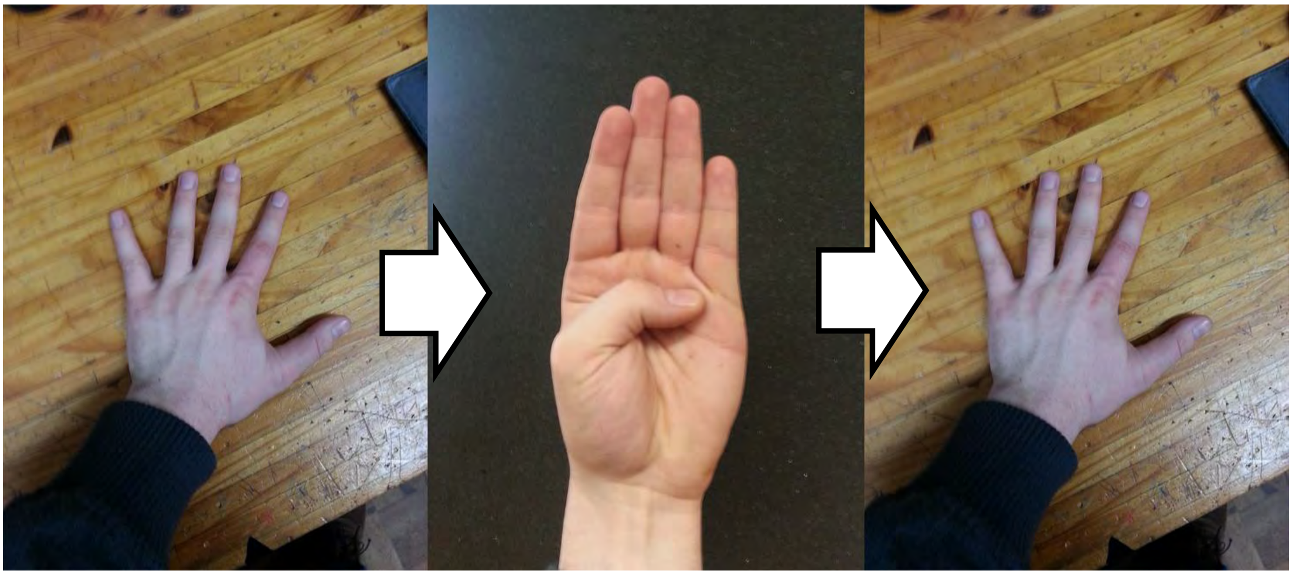


Figure 37: Hand showing the type 2 data recording process(without glove). The hand starts in a neutral position, moves to the gesture and then moves back to the neutral position.

6.1.3 Type 3

This test was performed in exactly the same manner as test used to collect the Type 2 data. The only difference is that the participant's hand starts and ends in different unknown positions of their choosing. The participants were cautioned to hold their hand steady in the unknown position in order to keep the data consistent.

While type 1 and type 2 data had five participants, type 3 data was limited to one participant. This was due to time and participant availability restrictions.

This data was recorded in order to make the database more robust and make it more similar to a real world scenario. Extracting the gesture from this data will be more complicated than that of the type 2 data as this data will suffer from most of the problems mentioned in the literature review. It is important to be able to extract gestures from continuous uncertain data as this is what the device would have to be capable of when used in day to day ASLT.

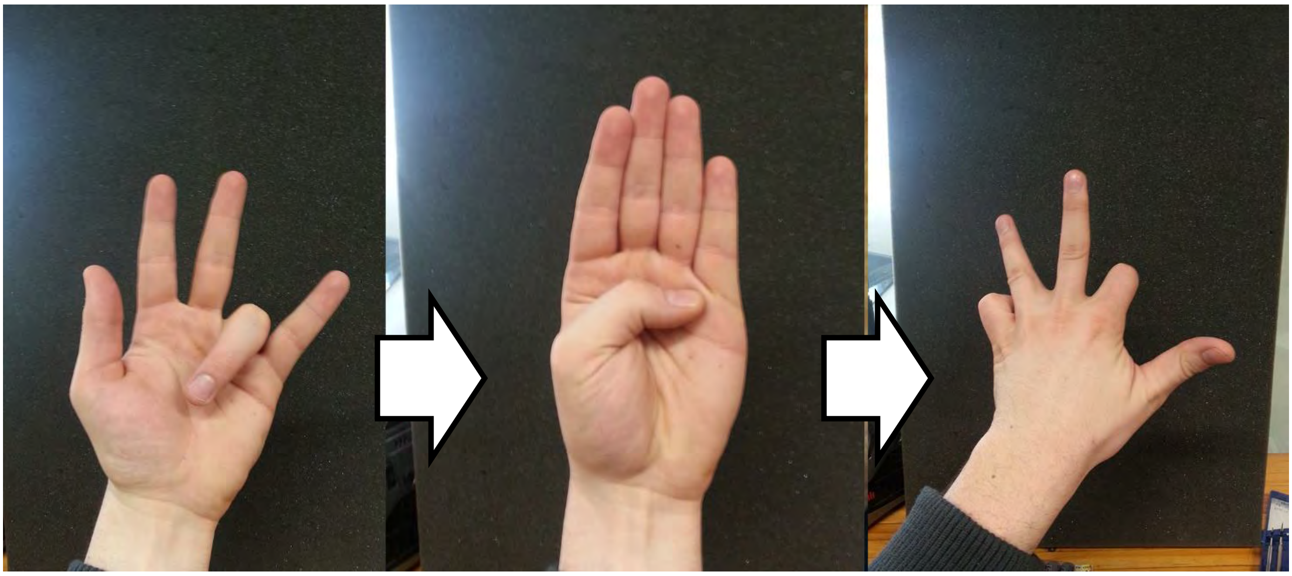


Figure 38: Hand showing the type 3 data recording process (without glove). The hand starts in an unknown position, moves to the gesture and then moves to a different unknown position.

6.2 Data Cleaning

6.2.1 *Data Glove Outputs*

The data glove can send a variety of sensor data to the workstation via cable or Bluetooth communications. When the device is turned on, the data glove simply transmits all of its data continuously. The data glove sends 28 data readings which are as follows:

- 1 – 10: Flex sensor data
- 11-14: Contact sensor data
- 15 – 20: IMU Angle data
- 21- 23: IMU raw accelerometer readings
- 24-26: IMU raw gyroscope readings
- 27: Push button
- 28: Lipo battery life percentage

This data can be read by opening the correct COM Port on a PC.

6.2.2 *Normalisation*

While all of the sensors are read in a range of 0-1023 on the 10 bit ADC they are not completely consistent. The scale between two sensors may vary due to the, roughly 10%, resistance tolerance in the flex sensors. This can cause problems in the machine learning phase of the project and the larger scale values may be given more weight than other sensors which could skew the gesture classification algorithm. To ensure that this does not occur, a simple normalisation was applied to all of the data:

$$x_{i\_norm} = \frac{x_i - S_{min}}{S_{max} - S_{min}} \tag{40}$$

Where

x_i is the ADC reading for the sensor

S_{min} is the overall minimum value read across all of the gestures for that particular sensor

S_{max} is the overall maximum value read across all of the gestures for that particular sensor

This normalisation method was applied to all three types of data after it was recorded. The raw data was kept as a backup so that another method of normalisation could be used if deemed necessary.

6.2.3 *Contact Sensor Thresholding*

While the flex sensors give a continuous reading between 0 and 90 degrees, the contact sensors give a binary reading of contact or no-contact. A threshold was introduced to the raw data of the contact sensors. If the ADC reading of the contact sensor was above 350, the output would be a 1. If the ADC reading is below 350, then an output of 0 would be given.

6.2.4 *Data encoding*

Since the Type 1 data was used for the training of the neural networks it needed to be encoded in a manner that can be understood by the neural network algorithm. When each gesture was performed and recorded a target value was assigned to it.

For example when the participant placed their hand in the 'A' handshape, the sensor data was recorded and the vector was labelled as gesture 'A'. When the gesture samples are passed into the neural network the character 'A' has no numerical value and so the neural network would not be able to use the target 'A'. To solve this problem the target data has to be encoded in order to be understood by the algorithm. This is done by creating a 1 x 31 vector and each static gesture will represent a value of 1 in its appropriate column. For clarification some examples have been given below:

$$A = [1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]$$
$$B = [0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]$$
$$9 = [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1]$$

A full list of the target vectors can be found in Appendix A

6.2.5 Feature Extraction

When considering the Type 2 and Type 3 data it is important to note that not all of the data is useful. In the 10 second interval that was recorded only a fraction of that holds the gesture data. This data must be extracted and reduced down to a single 1x14 row vector in order to classify the gesture correctly. The type 2 data will be easier to extract than that of the type 3 data since it involves known positions. Two methods were used to extract the relevant data.

i. Manual Extraction

This is a very basic and time consuming method for extracting data but it is accurate as the researcher could select the best data available from the recorded data. The data was examined manually in order to determine when the gesture had taken place. Once determined, a small subset of the data stream was chosen and averaged so that it becomes a 1 x 14 vector that can be passed into the neural network. This method was used mainly on the type 3 data but it was also used on one set of type 2 data in order to ensure that the best data was chosen for testing purposes and to prove that the dataset gives satisfactory results.

ii. Sensor Velocity Extraction

When trying to extract Type 2 data, the simplest method is to look at the overall movement of the sensor readings. Saengsri [37] used a sensor velocity summation equation to calculate when there was hand movement in the continuous data stream. This equation is used in order to determine the stop, start, gesture and the transition state. Once these states have been identified it is simple to separate the gesture state from the other states and extract the relevant gesture data. The finger flexion is calculated using the following equation:

$$S(t) = \sum_{i=0}^{13} |x_{i,t+1} - x_{i,t}| \tag{41}$$

Where,
$x_{i,j}$ is the value at timestamp t from the sensor

Consider Figure 40 and Figure 39, the sensor velocity graph clearly shows two large spikes which indicate movement of the participant's hand. These spikes coincide with the movement on the flex sensors shown in Figure 39. Figure 40 clearly indicates several points of interest. Sections A, C and F show states of very little movement, mainly sensor noise or minor movements, such as finger twitches, by the participant, these are the steady states. A and F show when the user has kept their hand in the neutral position and C shows the users hand holding a gesture for a certain interval. A is the start state, F is the stop state and this leaves C to be the gesture state. Sections B and E show two very large spikes on the graph and this indicates large movements such as moving a hand from the neutral position to a gesture or from a gesture to the neutral position again. These sections are known as the transition states. The last label D shows a spike in the gesture state which indicates erroneous movements that should not have occurred. These error velocities could have been caused by the participant adjusting their hand slightly once in position or a slight sensor malfunction.

The velocity graph effectively breaks up the continuous data stream into the various states and identifies the relevant gesture data. The two peaks are used to break up the data stream into the different states. Once the various states have been determined a threshold is applied to the gesture state so that only data below the threshold is extracted and used. The threshold removes erroneous movements and sensor spikes. In this case the threshold was chosen to be twice the average of the entire data stream. The steady data between sections B and E is extracted and averaged in order to get a single 14x1 vector that can be classified by the trained neural network.

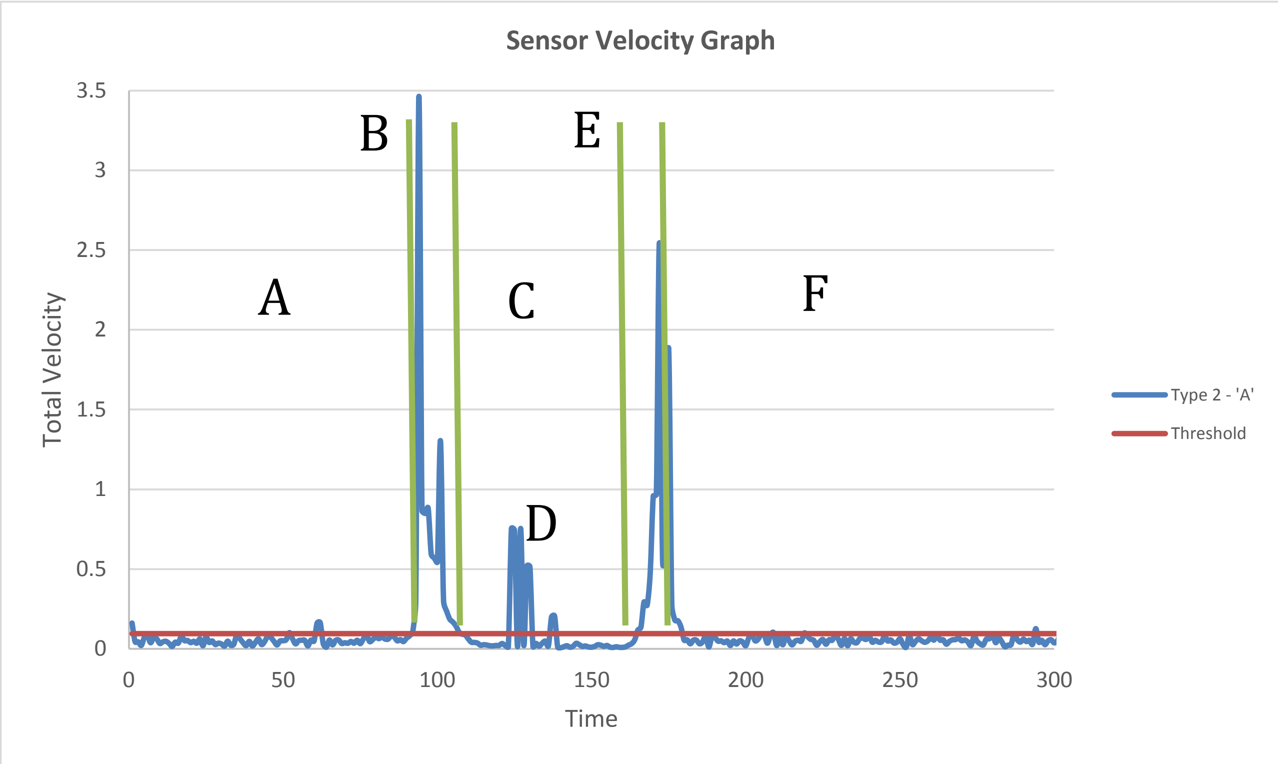


Figure 40: Sensor velocity graph of Type 2 data for gesture 'A'

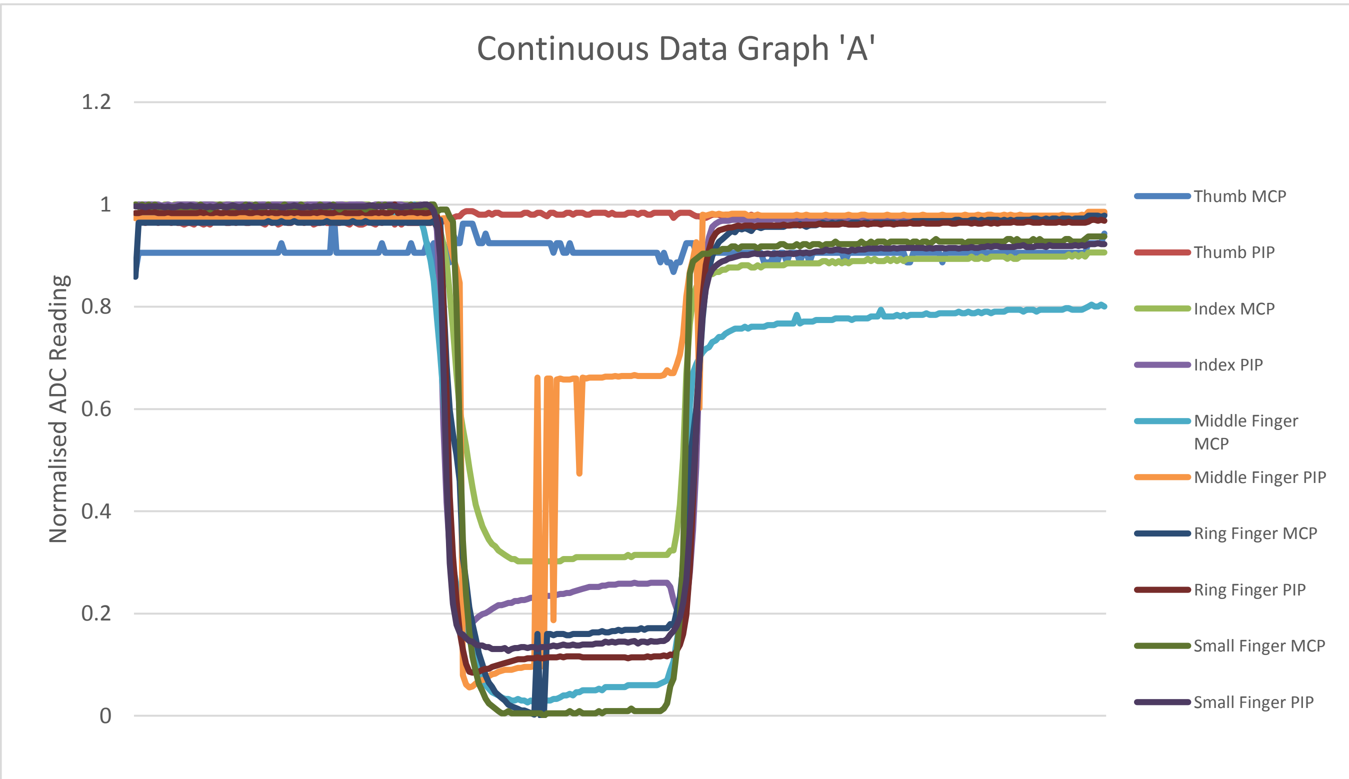


Figure 39: Continuous flex sensor data of Type 2 data for gesture 'A'

6.3 Data Results

6.3.1 *Type 1*

The data was recorded and cleaned as mentioned above. Each of the 31 gestures had 10 recordings taken from 5 participants and in total 1550 data points were collected. The data will be visualised and compared in order to determine the consistency between participants and the separability between gestures.

i. ***Histogram Comparison***

The SASL gestures for A, B, C and D are represented using histograms in Figure 41. The histograms show the normalised values from each of the 14 sensors used when recording the data. The normalisation is necessary as it places all of the readings on the same scale so that they are comparable. The tolerances of the sensors would otherwise skew the readings of the histograms.

Each histogram shows the averaged result of the first three participants' data. This was done in order to show the overall similarity of the sensor readings between different participants. Large variations between the participants could cause misclassifications and result in a less effective classification system. It is expected that there are some variations in the readings due to different hand sizes. Small variations between the participants are acceptable as they will allow for a more robust classification system.

It can be seen in the histogram of gesture 'A' that participant 2 has values that do not match that of participant 1 and 3. This is because participant 2's hand was small and the glove did not fit as tightly as the other participants. This led to a less accurate reading. Overall the values show a consistent trend in the data between the different participants.

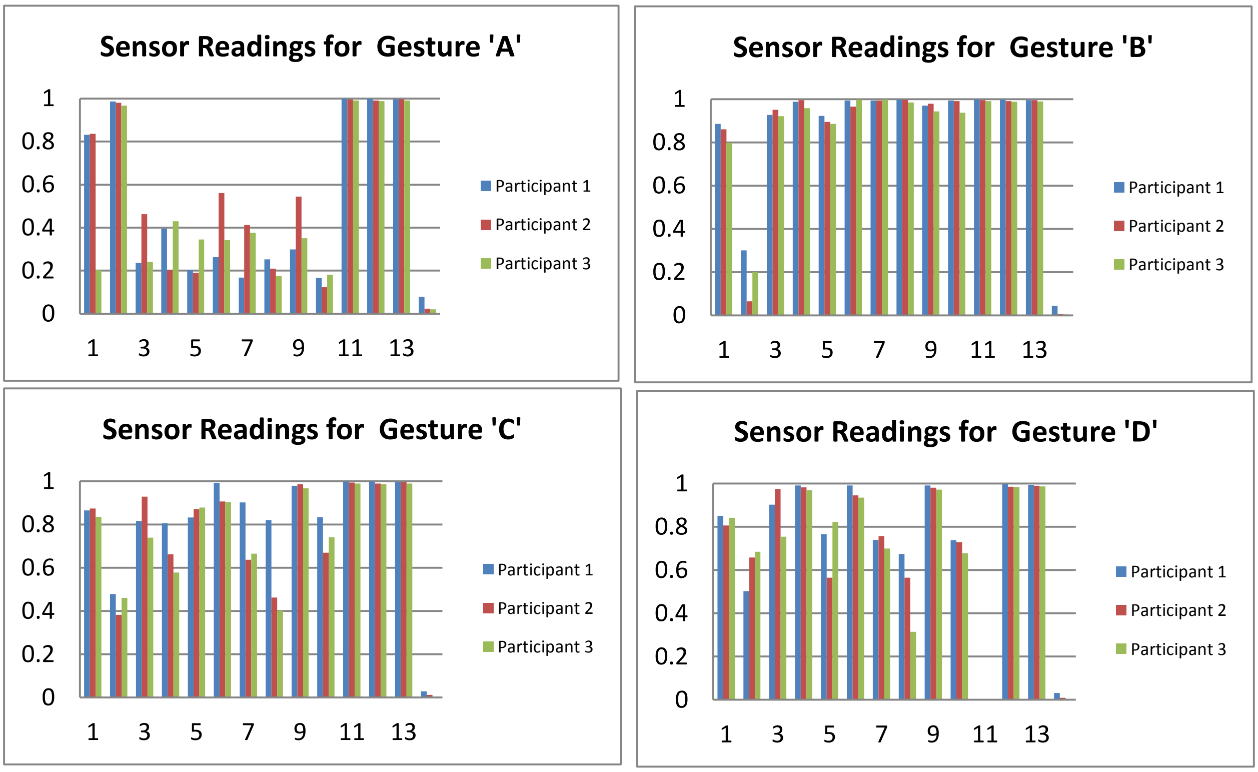


Figure 41: Histograms showing sensor readings of different letters in the SASL alphabet. Each histogram shows the averaged results from three different participants. The x-axis shows the 14 sensors used. Sensor 1 is the flex sensor on the thumb MCP joint, sensor 2 positioned on the thumb PIP joint, this is repeated for each subsequent finger until sensor 10 which is positioned on the small finger PIP joint. Sensor 11 – 13 are the abduction/adduction sensors and sensor 14 being the 'R' sensor.

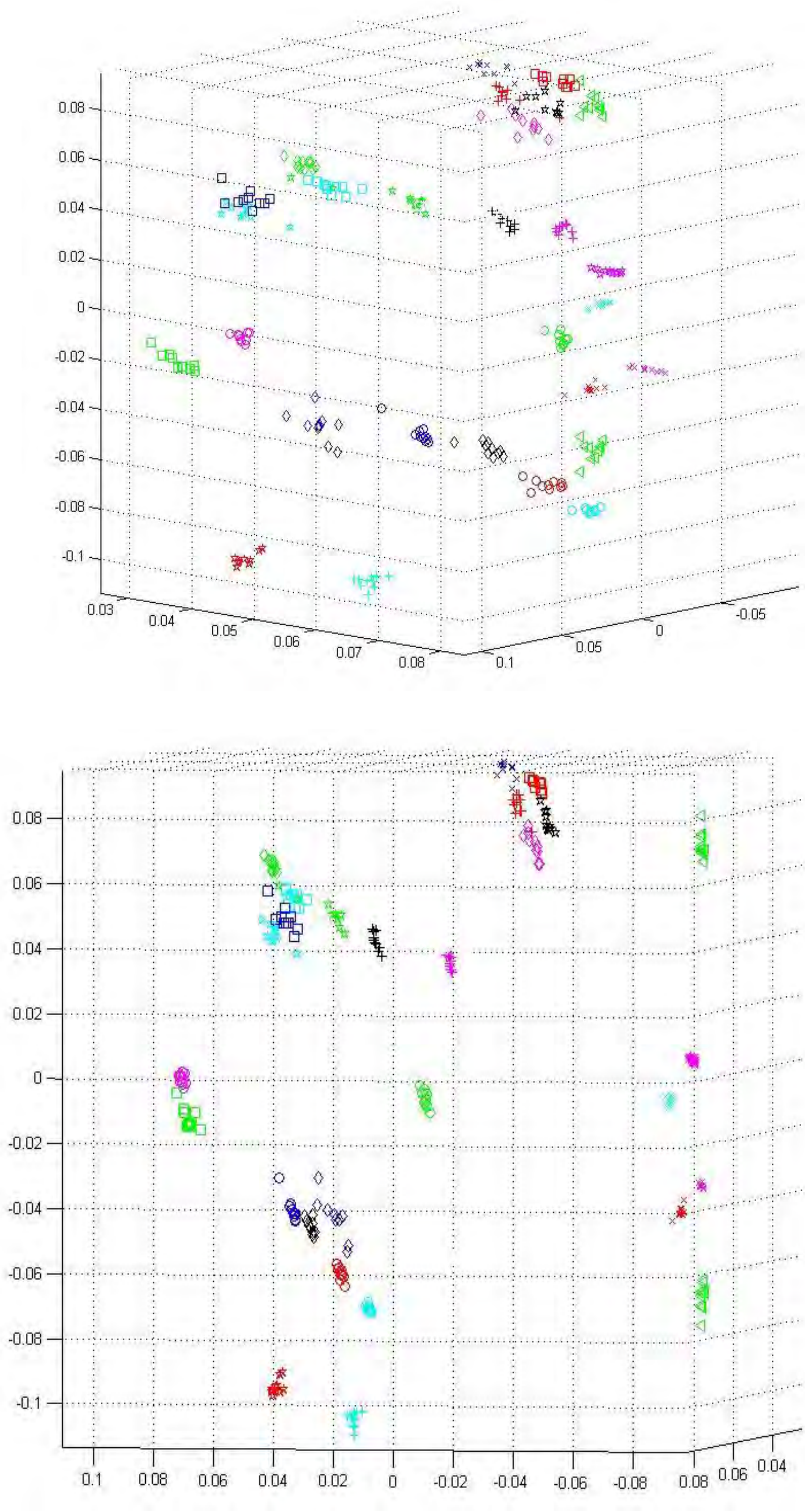


Figure 42: 3D Scatter plot of participant 1's Type 1 data using the three first principle components. Two different perspectives of the same 3D scatter plot are given in order to show the separability of the data.

ii. ***Scatter Plot***

The static data points for each gesture have 14 different variables which means that the data cannot be visualised for analysis. To solve this problem the dimension of the data must be reduced; this is done by breaking the data up into its three most significant principle components. This was done in MATLAB using singular value decomposition. The function was used to reduce the data points into three variables so that they could be plotted on a scatter plot.

The scatter plot (Figure 42) shows all of the 31 static gestures recorded by participant 1. Each of the 31 gestures was plotted with a different symbol and colour so that they could be differentiated easily on the scatter plot. It is clear that while there are few gestures that group close together, they are still separated from each other and they do not overlap. The points that are clustered closer together are the gestures that are most likely to be misclassified during the classification stage. Most of the gestures are well separated and are not close to other gestures.

6.3.2 Type 2 Classification

The type 2 data was recorded for each of the five participants and cleaned. Due to the nature of the data it is displayed using 5 graphs to show the results recorded by the different sensors. The first and second graph (Figure 43 and Figure 44) shows the data collected over the 10 second interval from the flex and contact sensors respectively. The three remaining graphs show the IMU data. Figure 45 shows the variation in hand angles, the raw acceleration data from the IMU and the raw gyroscope data during the 10 second interval. When considering the graphs, it is very clear when the gesture is taking place. The transitional state matches up across all five graphs, showing the movement from the neutral position, to the gesture and then back again.

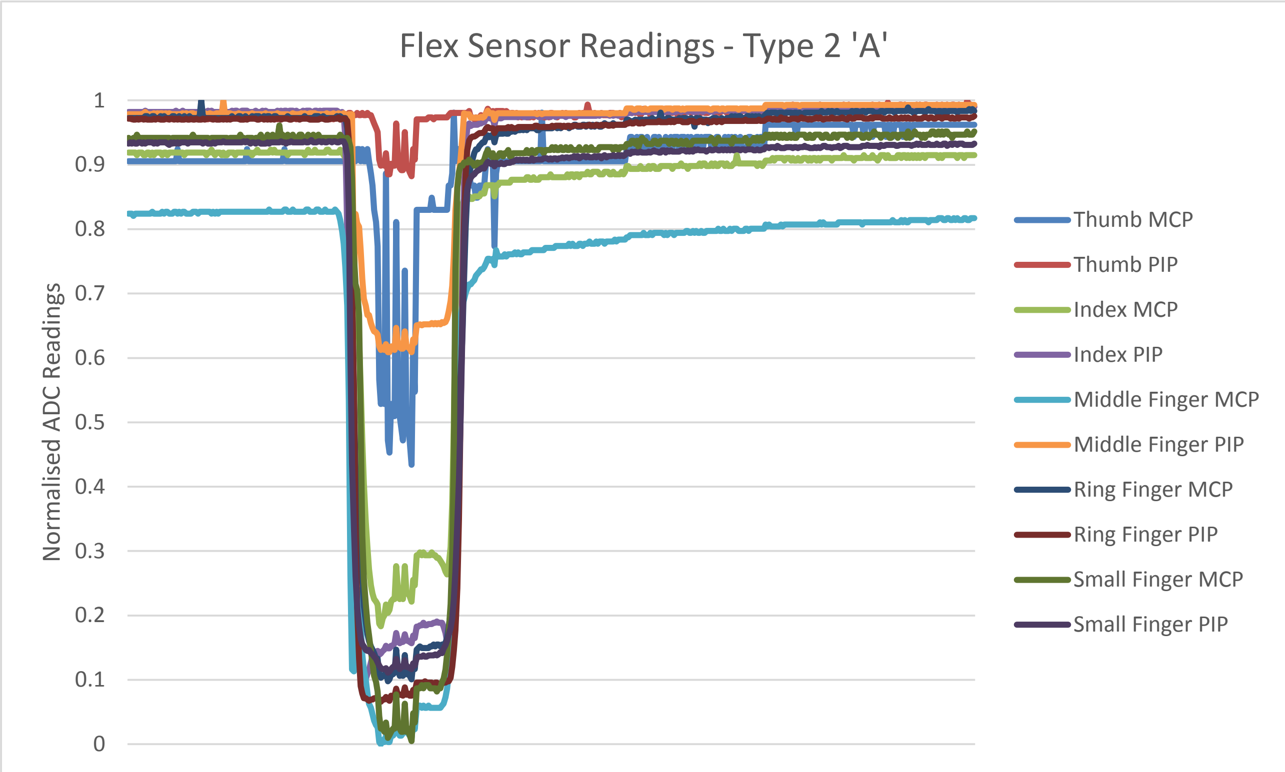


Figure 43: Continuous flex sensor data recorded while performing type 2 SASL gesture 'A'

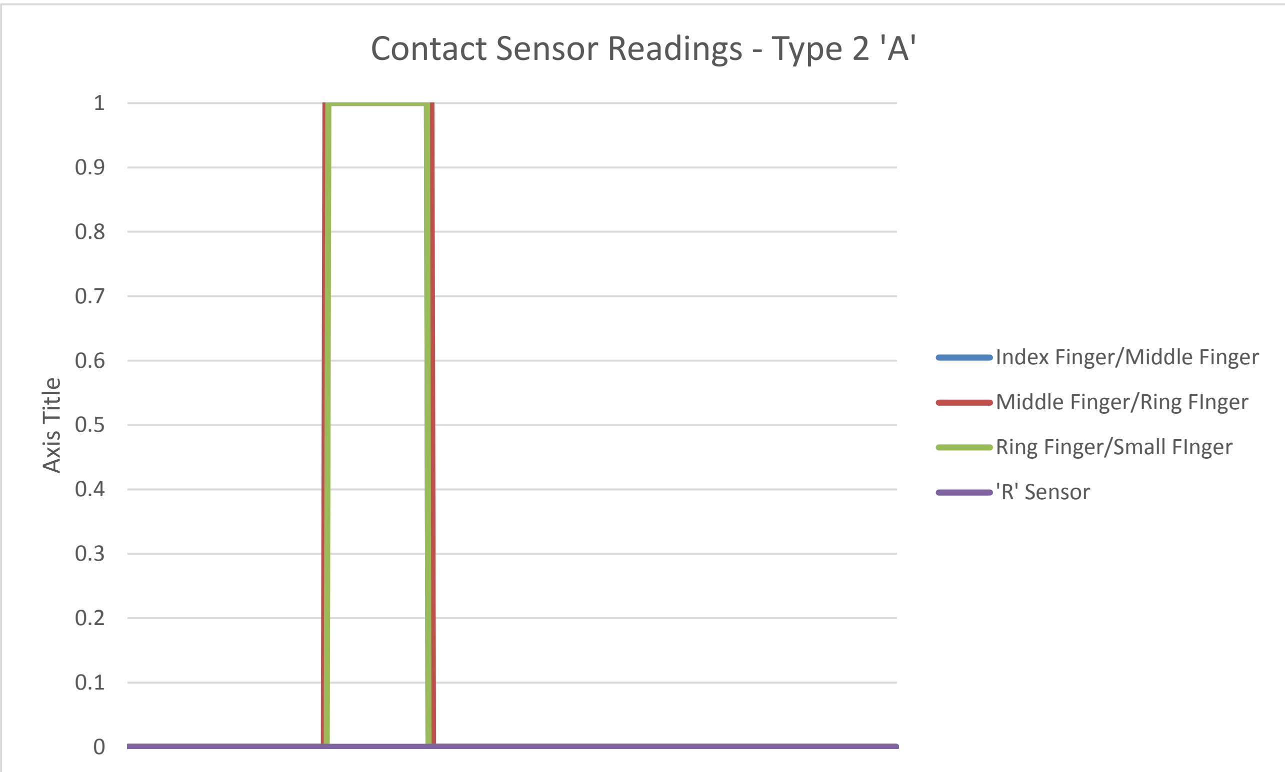


Figure 44: Continuous contact sensor data recorded while performing a type 2 SASL gesture 'A'.

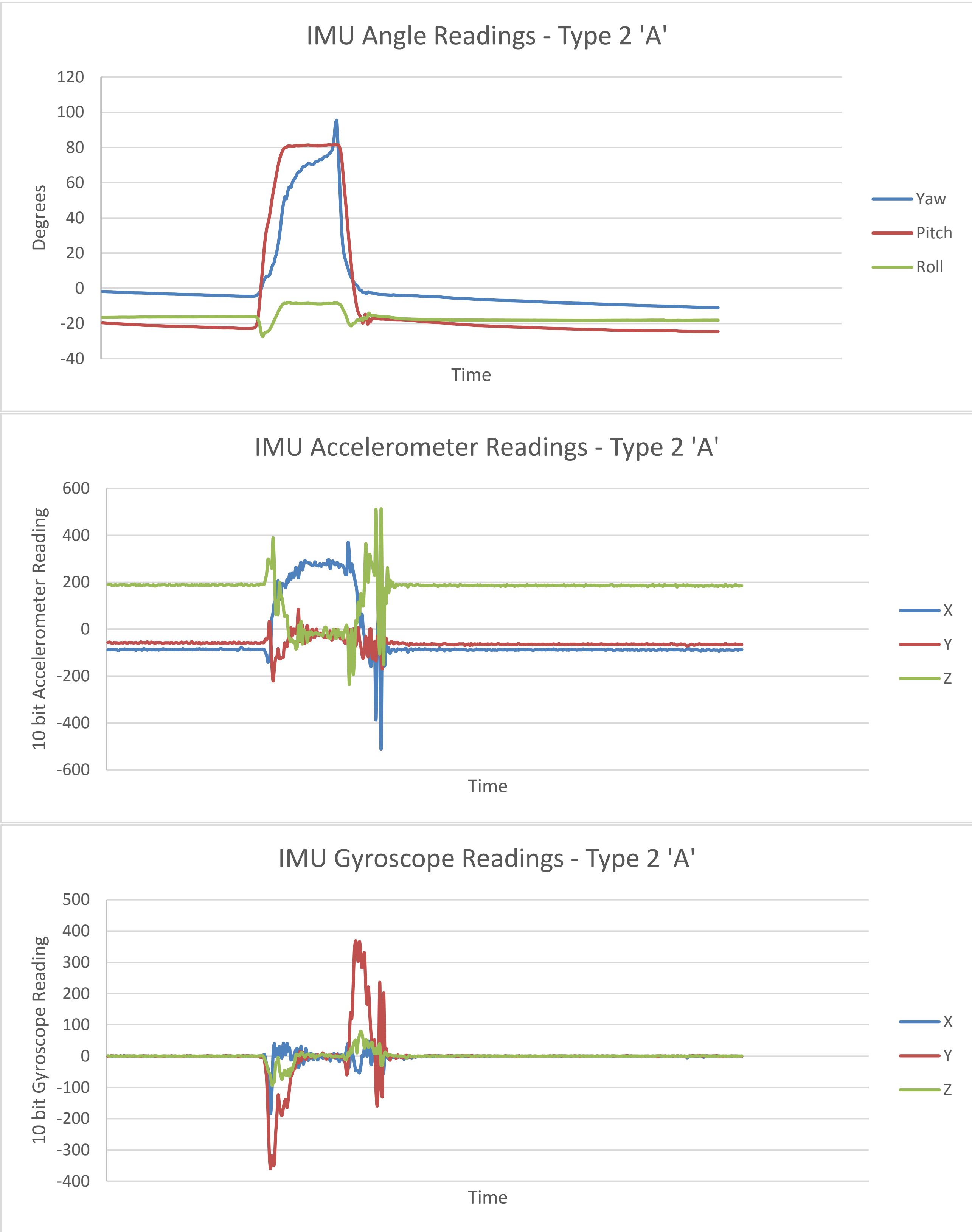


Figure 45: Top - Continuous IMU angle data recorded while performing a type 2 SASL gesture 'A'. Middle - Continuous IMU accelerometer data recorded while performing a type 2 SASL gesture 'A'. Bottom - Continuous IMU gyroscope data recorded while performing a type 2 SASL gesture 'A'.

6.3.3 Type 3 Classification

The type 3 data was only recorded for one participant. An example of the continuous data for the gesture has been shown below. Type 3 data moves from an unknown position to the gesture and then back to an unknown position again. The figures show this clearly when compared to the same gesture graphs shown for the type 2 gesture.

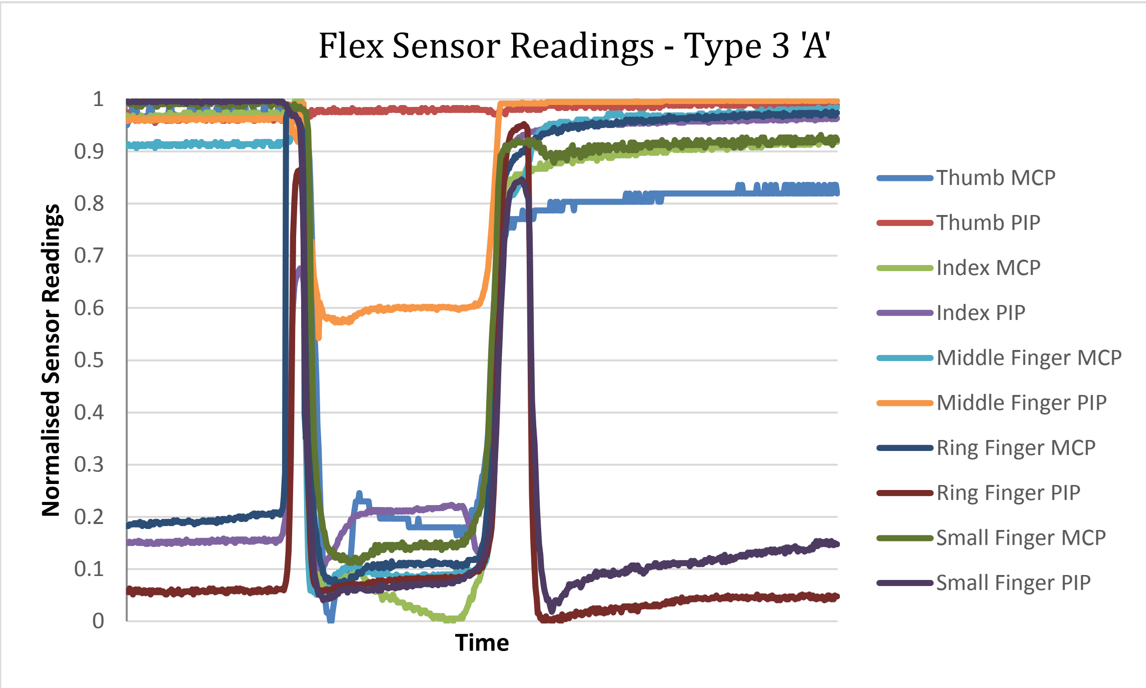


Figure 47: Continuous flex sensor data recorded while performing a type 3 SASL gesture 'A'.

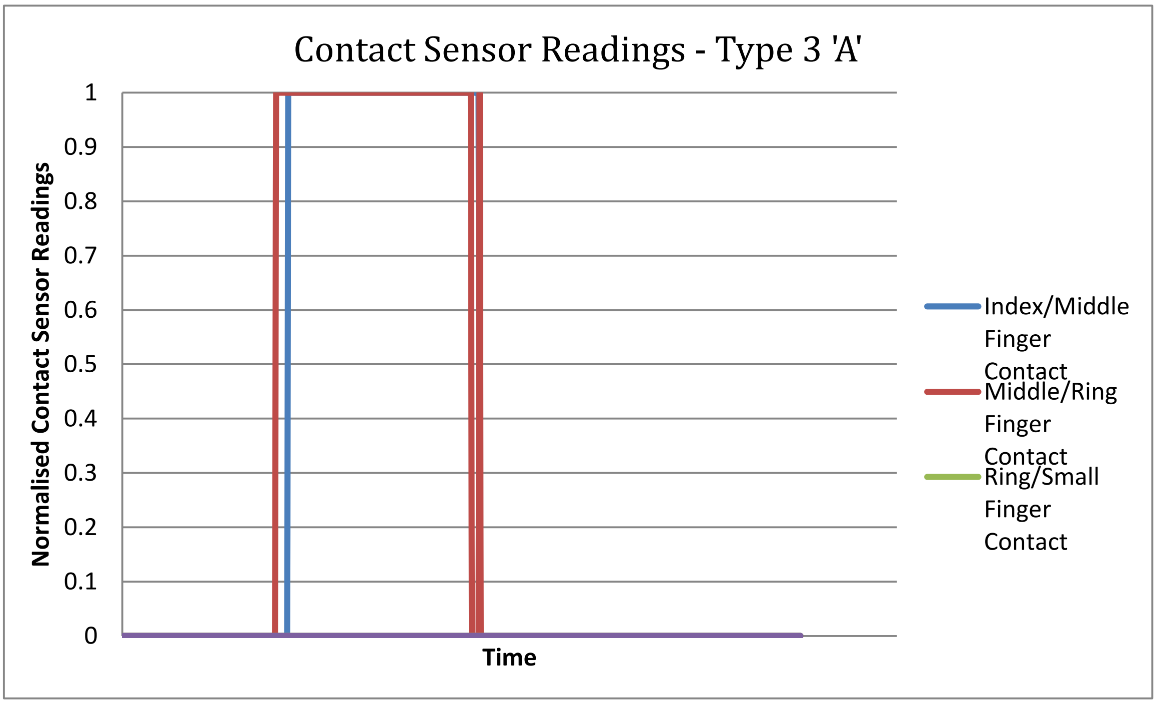


Figure 46: Continuous contact sensor data recorded while performing a type 3 SASL gesture 'A'.

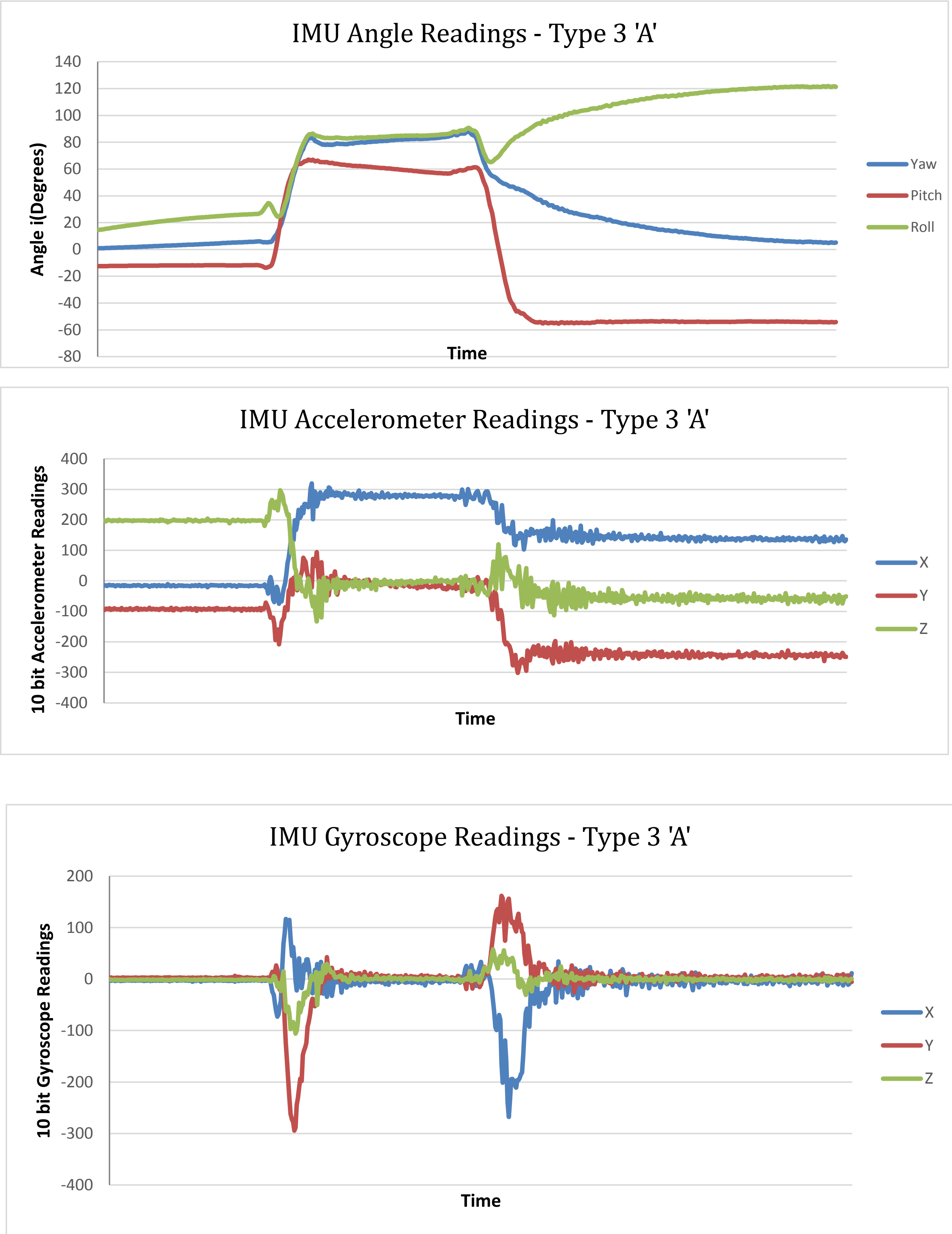


Figure 48: Top - Continuous IMU angle data recorded while performing a type 2 SASL gesture 'A'. Middle - Continuous IMU accelerometer data recorded while performing a type 2 SASL gesture 'A'. Bottom - Continuous IMU gyroscope data recorded while performing a type 2 SASL gesture 'A'

6.3.4 Data Evaluation

The three data types have been recorded and analysed individually, they must now be compared to each other in order to determine their consistency across the different scenarios. The relevant gesture features must be extracted and compared in order to determine the effectiveness of the extraction techniques as well as the reliability across the various scenarios.

i. Data types histogram comparison

It is important to see how the different types of data vary in relation to each other. The type 1, 2 and 3 data was taken from participant 1 and compared using histograms. It is clear that the data is consistent across the gestures and sensors except for sensor 2 which is the thumb PIP flex sensor for type 2 data. This variation could have been caused by the feature extraction method.

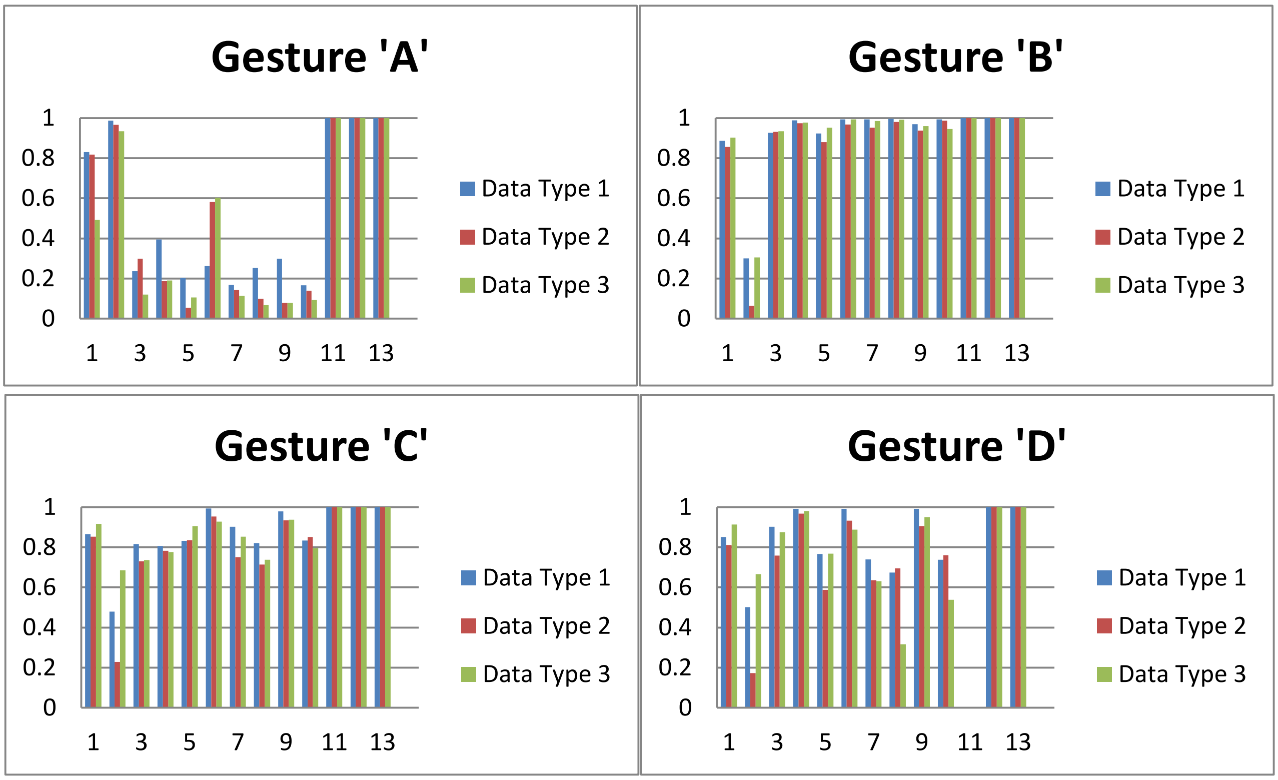


Figure 49: Histograms showing a comparison between the type 1 data and the extracted type 2 and 3 data for the gestures A, B, C and D

ii. Data types scatter plot comparison

A 3D scatter plot was created for Dataset B –Type 1 data. This dataset contains the static gesture information of gestures A, B, C and D for the first three participants. Each of the gesture points of each participant was averaged. This data was decomposed into its three most significant principle components using singular value decomposition. The data was then plotted on a 3D scatter plot in order to see how the data clusters.

It is clear that each gesture is well separated and none of the gestures overlap with each other. The data for gesture 'A' (magenta circles) is spread out in comparison to the other gestures. It is important to note that the scale of the scatter plot is small.

The scatter plot (Figure 50) effectively shows that in general the averaged gesture data for the first three participants is similar which means that the normalised sensor data is consistent when different subjects use the glove. The scatter plot shown in Figure 51 compares type 1, type 2 and type 3 data of the first three participants for the gestures 'A', 'B', 'C' and 'D'. It is clear that the data is well separated and there is little overlap. The main concern is that the type 2 data is quite far removed from the other two types of data on one axis. This will lead to a lower classification rate for the type 2 data.

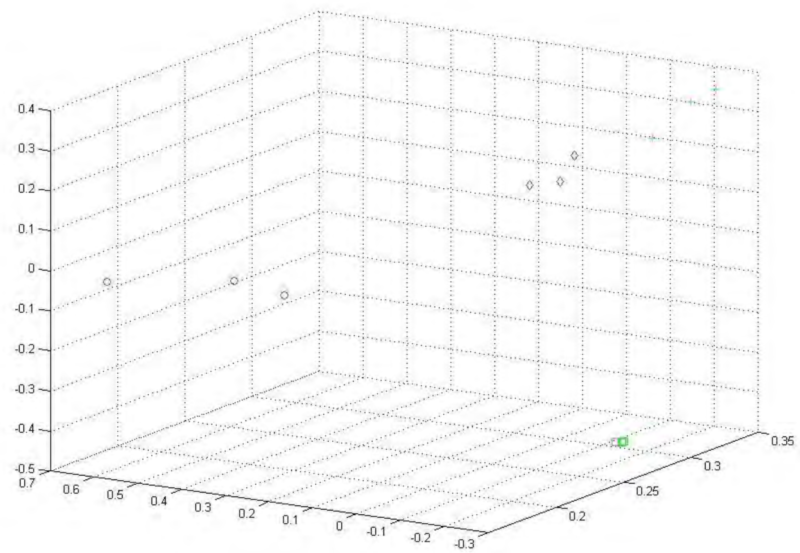


Figure 50: 3D Scatter plot showing a comparison between the first three participants type 1 data. The gestures A, B, C and D of each participants static data was averaged and plotted on the scatter plot. The magenta circles represent 'A', the cyan plus represent 'B', the red diamonds represent 'C' and the green squares represent 'D'.

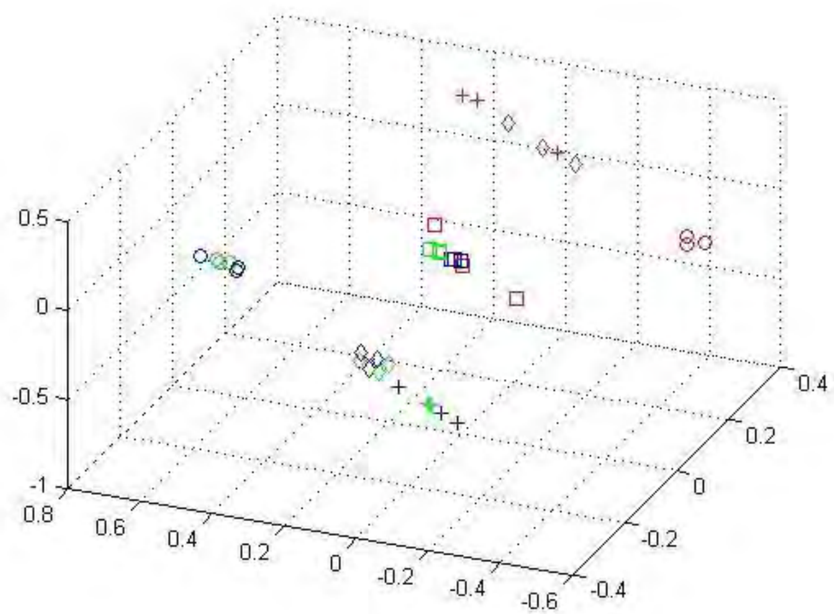


Figure 51: 3D Scatter plot showing a comparison between the first three participants' gesture data. The scatter plot shows type 1 (blue), type 2 (red) and type 3 of participant 1 (green). The gestures being shown are 'A' (circle), 'B' (plus), 'C' (diamond) and 'D' (square).

6.4 Neural Network Implementation

While the implementation of neural networks for the classification of SASL was not a main focus in this research they were used in order to prove that the SASL database collected in the previous section is effective and can be used.

MATLAB is a numerical computing environment that has been used for a huge variety of applications. It was chosen because it has a toolbox that allows the user to implement neural networks quickly and effectively. This toolbox allows the user to choose various parameters such as the number of hidden neurons and which training algorithm to use. It has a variety of training algorithms available for implementation and it also generates the code which allows the user to tweak and customise the algorithm even more.

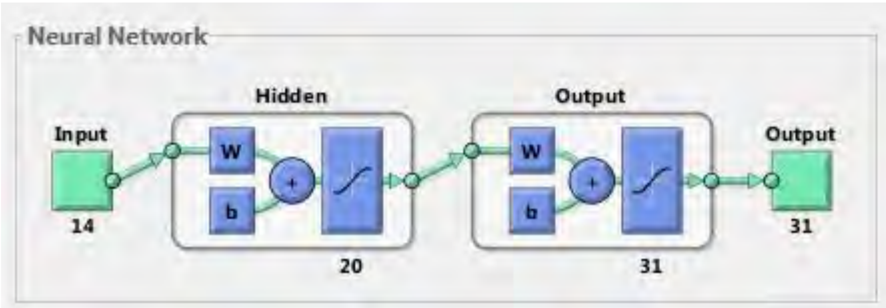


Figure 52: MATLAB Neural Network Diagram

Four training algorithms were chosen for their proven success with classification applications. These are Gradient Descent with momentum and an adaptive learning rate (Gdx), Resilient Back-Propagation (Rprop), Conjugate Gradient Decent (SCG) and Levenberg-Marquardt (LM).

Once the training algorithms were chosen the parameters for the network had to be determined. The default approach is to use one input layer, one hidden neuron layer and one output layer. The number input and output nodes are simple to choose as they match the data that is being used by the system. The number of input nodes was chosen to be 14 to match the number of input variables from the sensors and the number of output nodes was chosen to be 31 to match the number of gestures to be classified. There is no general rule to choose the number of hidden neurons other than it should not be less than the number of input neurons. An iterative approach was taken to determine the number of hidden neurons.

Within the MATLAB Neural Network Toolbox there are various performance indicators that can be set by the user. These indicators can be used to control the length of the training process. When an indicator is reached then the algorithm stops. This is to ensure that the algorithm does not waste time by running calculations that will have negligible or worse results that it has currently achieved.

These indicators used are as follows:
Iterations: Is the number of times the algorithm runs to minimise the cost function. The number of iterations was set to 1000 epochs.
Performance: This is linked to the overall Mean Squared Error for the outputs. The performance indicator was set to 0.
Gradient: This is the performance gradient, as the training algorithm approaches convergence the change of performance drops drastically and if it drops below a certain defined rate the algorithm will stop. The gradient was set to 1×10^{-6}.
Validation Checks: These checks ensure that the algorithm is improving with each iteration. If the

algorithm makes a user defined number of worse iterations then the algorithm will stop. The number of validation checks was set to 6.

Before the training starts the data must be broken up into three datasets; training data, validation data and testing data. A common ratio for this is 70% training data, 15% validation data and 15% test data. The training data is the actual data used in order to train the neural network and minimise the output error of the system. The validation data set is used to measure the generalisation of the network. The test data set is completely removed from the training process and acts as an independent performance indicator for the network once it is trained.

An iterative process was used to determine the most effective number of hidden neurons. This process was done using three out of the four algorithms. The Levenberg- Marquardt training algorithm takes a considerable amount longer to train than the other algorithms and so it was left out of this process but it will be considered for the Type 2 and 3 classification process. The performances of the three algorithms were compared and evaluated so that an appropriate number of hidden neurons can be chosen. The results were tabulated below:

	Rprop		**SCG**		**GDX**	
No. Hidden Neurons	**Misclassification (%)**	**MSE**	**Misclassification (%)**	**MSE**	**Misclassification (%)**	**MSE**
14	5.564	0.00384	59.03	0.02516	44.19333	0.01765
31	3.07	0.00236	**42.646**	**0.01418**	23.512	0.01044
45	2.606	0.00182	45.034	0.0153	19.238	0.0087
55	2.18	0.00156	58.334	0.0192	18.84	0.00852
70	2.116	0.0016	46.684	0.01548	18.722	0.00794
100	**1.484**	**0.00108**	77.548	0.02584	**11.02**	**0.00556**
120	2.772	0.0016	58.504	0.01926	11.792	0.00558
150	2.334	0.00162	74.258	0.0246	13.404	0.00642

Table 10 : Comparison of results by different neural network training algorithms for different numbers of hidden neurons. The minimum value for each algorithm is in bold.

Each of the training algorithms were run five times and the results were averaged for each number of hidden neurons. The minimum values were highlighted in bold. The scaled conjugate gradient (SCG) training algorithm achieved very poor and inconsistent results and will not be considered further. The GDX method improved as the number of hidden neurons increased but it only managed to attain a minimum misclassification percentage of 11% whereas the Rprop method managed to achieve a value of just under 1.5% which is nearly 7 times smaller in comparison. It is clear that the Resilient back-propagation algorithm is the best fit out of the three algorithms. Out of the three algorithms two had a reached a minimum value when using 100 hidden neurons and so this is the number of hidden neurons that will be used for the rest of this research.
Once the number of hidden neurons had been chosen a comparison between the LM and Rprop training algorithms could be done in order to determine the best algorithm for the classification process.

i. *Neural Network Algorithm Comparison*

All of the parameters for the neural network have been determined and so the final step was to choose the best algorithm for the application of South African Sign Language classification. The two algorithms that will be compared are the Resilient back-propagation and the Levenberg- Marquardt algorithms.

Each training algorithm was trained using all of the type 1 static data and then evaluated using 3 sets of data;
Dataset A: Contains the type 1 data of participant 1
Dataset B: Contains the type 1 data of participants 1 to 3
Dataset C: Contains the type 1 data of all 5 participants.

Dataset A should give the best results considering that the glove was designed to fit participant 1's hand and so the readings were most accurate in this dataset. This data set should show how accurate the algorithm can be using the most 'ideal' data available.
Dataset B contains the gesture data of two participants that the glove fits well (#1 and #3) and one participant who's hand was slightly too small. This would test the algorithms ability to generalise between various participants that gave good data.
Dataset C uses all of the participants' data and this will show how well the algorithm deals with a large amount of variation and in relatively bad data (participant 5) when compared to the overall data collected. The neural network was trained and the results have been tabulated below:

	Rprop			LM		
Dataset	Type 1 Gesture Classification (%)	Type 2 Gesture Classification (%)	Type 3 Gesture Classification (%)	Type 1 Gesture Classification (%)	Type 2 Gesture Classification (%)	Type 3 Gesture Classification (%)
A	99.68	59.14	78.49	99.68	77.42	81.72
B	98.92	54.84	-	99.68	65.23	-
C	98.32	52.04	-	99.61	60.43	-

Table 11: Comparison of gesture misclassification percentages between LM and Rprop training algorithms when inputting Type 1 and Type 2 data using datasets A, B and C

Both of the algorithms achieved successful results, nearly 100%, when considering the Type 1 data. The Rprop algorithms results decreased marginally when using more participants while the LM algorithm remained near constant. As expected the Type 2 data had lower classification accuracies due to the continuous nature of the data and the need for the data to be extracted. The LM algorithm achieved an18,28% higher classification percentage for Type 2 data than the Rprop algorithm when using Dataset A, 10.39% lower for Dataset B and 8.39% lower for Dataset C. It is clear that the LM training algorithm has superior classification accuracy. The classification accuracy could be improved by using data gloves that fit all of the participants well or improve the calibration of the sensors in order to ensure consistency between the participants. Another aspect to look into would be that of the feature extraction, effectively choosing the relevant data would improve the overall classification of the system. Collecting more data would also allow for better training of the neural network which would lead to a higher classification accuracy. Finally, a more specialised training algorithm could be developed for the purpose of classification as the algorithms used were the standard neural network implementations using MATLAB.

The main problem with the LM training algorithm is that it takes a considerable amount of time, just over 10 hours when using 100 hidden neurons, to train whereas the Rprop algorithm can be trained in under 5 seconds when using the same number of hidden neurons. The time it takes to train the training algorithm is not too important in this phase of the research. The gesture classification using the data glove prototype has been done in lab conditions and not in real life conditions. If this research is taken further, the training time of the algorithm will be a major factor to consider. For the purpose of this research the LM algorithm has been chosen as the final training algorithm for the neural network as it gives significantly better results than that of the resilient back-propogation algorithm.

ii. Confusion Matrix

The confusion matrix is used to visually determine which gestures were misclassified. This can be useful in discovering which gestures are similar in terms of sensor readings. This information can be used to improve the glove design for future improvements or help to implement changes to the classification algorithm. The confusion matrix was generated for the type 1 dataset C and the type 2 dataset C. The confusion matrices are shown below:

#	A	B	C	D	E	F	G	H	I	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	1	3	4	5	7	8	9
A	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
E	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
F	0	0	0	0	0	2	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
G	0	0	0	0	0	0	1	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
H	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
I	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
K	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
L	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
M	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0
N	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
O	0	0	2	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
P	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Q	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0
U	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
V	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0
W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0
X	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0
Y	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	2	0	0	0	0	0
4	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3

Figure 53: The confusion matrix shows the 21 misclassified gestures for type 2 gestures that make up Dataset A (which only contains participant 1). Dataset A only contains 93 gesture data points, 3 for each gesture. Dataset A was chosen as it gave the best results out of the three datasets and shows the best case scenario. The means that the misclassifications shown will most likely be the most commonly misclassified gestures for the other participants.

#	A	B	C	D	E	F	G	H	I	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	1	3	4	5	7	8	9
A	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
B	0	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
C	0	0	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D	0	0	0	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
E	0	0	0	0	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
F	0	0	0	0	0	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
G	0	0	0	0	0	0	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
H	0	0	0	0	0	0	0	49	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
I	0	0	0	0	0	0	0	0	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
K	0	0	0	0	0	0	0	0	0	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
L	0	0	0	0	0	0	0	0	0	0	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
M	0	0	0	0	0	0	0	0	0	0	0	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
N	0	0	0	0	0	0	0	0	0	0	0	0	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
O	0	0	0	0	0	0	0	0	0	0	0	0	0	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
P	0	0	0	0	0	0	0	0	0	0	0	0	0	0	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Q	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	50	0	0	0	0	0	0	0	0	0	0	0	0	0
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	50	0	0	0	0	0	0	0	0	0	0	0	0
U	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	50	0	0	0	0	0	0	0	0	0	0	0
V	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	50	0	0	0	0	0	0	0	0	0	0
W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	49	0	0	0	0	0	1	0	0	0
X	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	50	0	0	0	0	0	0	0	0
Y	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	50	0	0	0	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	48	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	50	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	50	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	49	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	50	0	0
8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	49	0
9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	50

Figure 54: The confusion matrix reveals the 6 misclassifications for type 1 gestures in Dataset C (which contains all 5 participants) that were collected and input into the neural network. Dataset C contains 1550 data points in total, 50 of each gesture.

7. Conclusion

7.1 Glove Development

7.1.1 Sensor Tests

The Flexpoint sensors were the correct choice for the prototype as they performed very well. They were linear, quick to settle, had little decay over short times and they were repeatable. The Flexpoint sensors achieved far better results than the 15 degree resolution required in the specification.
The IMU performed well on the roll and pitch axis but there were problems and inconsistencies with the yaw axis due to the lack of a magnetometer which could help to compensate on the yaw axis.
Overall the sensors met and exceeded the specification set out for the data glove design.

7.1.2 Data Glove Evaluation

A low cost data glove was designed and tested for the purpose of sign language translation. The glove proved successful as the results are far more accurate than the 15 degrees resolution set out in the beginning, even when the glove did not fit the participants hand well. The overall average results were a range of 3.270 degrees resolution and a standard deviation of 1.418 degrees. These results are as good as some of the best results published by researchers. If the participants use a glove that is made for them, then even higher resolutions could be achieved (For example participant 1 - Range: 2.218, SD: 0.805). The ergonomics of the glove were satisfactory as can be seen by the results of the questionnaire.
The overall results could be improved by using custom made gloves for each participant or in the interest of keeping costs down then three general sized gloves could be made to potentially increase the repeatability and reliability of the device.

The device that was developed performed well and has many advantages over those of commercial data gloves. The material used allows for many hand shapes and sizes while not compromising results. The device has a long battery life and would only need to be charged once a day where as some of the commercial devices would require two or more recharges which can be inconvenient for the user. One of the main objectives of this research was to develop a low-cost device and this goal has been achieved. The device is cheaper by $100 or more when compared to other research data glove devices and hundreds of dollars cheaper than commercial data gloves.

While many of the prototypes disadvantages will be remedied when it becomes a final product, they should be mentioned for sake of completeness. The user must be careful when putting the device on as it is handmade and could potentially rip. This increases the time that it takes to put on the glove which can be an inconvenience. Due to the handmade nature of the device, some of the sensors do not align as perfectly as if it was made by machine. This causes the sensors to not give the most effective results which reduce the overall performance of the device. The IMU used in the prototype has a large amount of uncertainty and drift on the yaw axis.

There are many improvements that can be made in future research. The IMU could be upgraded to 9 degrees of freedom in order to compensate for the yaw drift. The control unit could be designed on a PCB in order to reduce costs as well as size. Methods could be developed in order to increase battery life, more energy efficient components or more effective programming should be researched. The device could be professionally made in order to reduce human error as much as possible which will lead to more accurate sensor readings. Development of a smartphone application would be useful to the user as it would allow them to be more mobile on a day to day basis.

7.2 Database Development

The largest advantage of this database is that it is the first database to record SASL with the use of a data glove. None had existed previously, which led to this research. While this database was limited to 31 static gestures, it is the first step towards recording a full SASL database that includes both static and dynamic gestures.
Most research into creating sign language databases limits the number of participants to one or two, this database used five. This led to a more robust database as there is a variety of different hand sizes. The glove prototype fit two of the participants very well, two of the participants' hands were slightly too small and the fifth participants' hand was too wide.

The database also recorded three different scenarios, each a step closer to real life scenarios. Type 1 data was the isolated static gesture, Type 2 data was the continuous gesture data that used known positions and Type 3 data was continuous gesture data that used unknown positions. Type 1 was used to train the neural network while Type 2 and 3 were test cases that were used to evaluate the algorithm for more 'real' continuous scenarios.

While using a professional SASL translator is more important when recording dynamic gestures than static gestures it would have been advantageous to use a SASL translator in order to ensure perfect gestures and high quality data.

Future research should focus on expanding the database towards dynamic gestures with the help of professional SASL translators. This will help to create a more complete SASL database and take one step closer moving this research into the real world to break down the communication barriers between the hearing and Deaf communities.

7.3 Classification Implementation

While the classification aspect of this research was not a main focus, it was implemented in to show that the database can be used effectively. Resilient back-propagation and the Levenberg- Marquardt training algorithms were considered and it was determined that the LM algorithm achieved significantly higher classification results. Three types of data were used; type 1, type 2 and type 3. The type 1 data achieved a very high classification rate, 99.61%, when Dataset C was input.
The type 2 and 3 data did not achieve a classification rate as high, 77.42% and 81.72% respectively when considering Dataset A (Participant 1), but the results are sufficient to prove that the dataset is of a good quality. The classification rate drastically decreases when more participants are considered. This is to be expected when considering the amount of variation that can occur during the continuous recording process. The accuracy will also drop if the extracted data is not a good representation of the gesture state.

While the Levenberg- Marquardt training algorithm managed to attain successful results it took just over 10 hours to train. This may be acceptable for offline testing in a laboratory environment but it will would not work on a day to day basis, especially if the device is expected to learn and calibrate itself to the user over time.

A more in depth approach into the classification aspect of this research should be undertaken in order to improve the classification accuracy, the time it takes to train the network and to move the classification process to an embedded system so that the user does not have to be near a PC.

8. References

[1] Statistics South Africa, "Census 2011," 2011.

[2] Deaf Federation of South Afirca, "DeafSA," 2013. [Online]. Available: http://www.deafsa.co.za/sasl/. [Accessed 30 July 2014].

[3] Cyber Glove Systems, March 2013. [Online]. Available: www.cyberglovesystems.com.

[4] Fifth Dimension Technologies, *5DT Data Glove,* 2013.

[5] World Federation of the Deaf, "World Federation of the Deaf," Visuallife, [Online]. Available: http://wfdeaf.org/. [Accessed 30 July 2014].

[6] SLED, "Sign Language Education & Development," 2011. [Online]. Available: http://www.sled.org.za/. [Accessed 30 July 2014].

[7] S. Howard, Finger Talk: South African Sign Language (SASL) Dictionary, 1st ed., T. Chowles and L. Reynolds, Eds., Fulton School of the Deaf, 2008.

[8] Sign Language Education and Development (SLED), A South African Sigh Language Dictionary for Families with Young Deaf Children, Sign Language Education and Development (SLED).

[9] L. A. Jones and S. J. Lederman, Human Hand Function, New York: Oxford University Press, 2006..

[10] J. Lin, Y. Wu and T. S. Huang, "Modeling the Constraints of Human Hand Motion," Beckman Institute, Urbana-Champaign, 2000.

[11] T. . L. Kunii and J. Lee, "Model-Based Analysis of Hand Posture," *IEEE Computer Graphics and Applications,* pp. 77-86, 1995.

[12] D. J. Sturman and D. Zeltzer, "A Survey of Glove-based Input," *IEEE Computer Graphics and Applications,* pp. 30-39, 1994.

[13] G. J. Grimes, "Digital data entry glove interface device". United States of America Patent US 4414537 A, 15 September 1981.

[14] T. G. Zimmerman, J. Lanier, C. Blanchard, S. Bryson and Y. Harvill, "A hand gesture interface device," in *Human Factors in Computing Systems and Graphics Interface*, New York, 1986.

[15] L. Dipietro, A. M. Sabatini, and P. Dario, "A Survey of Glove-Based Systems and Their Applications," *Systems, Man, and Cybernetics,* vol. 38, no. 4, pp. 461-482, July 2008.

[16] K. Li, I.-M. Chen and S. H. Yeo, "Design and validation of a multi-finger sensing device based on Optical linear encoder," *Robotics and Automation,* pp. 3629 - 3634, 2010.

[17] C. I-Ming, S. H. Yeo, I.-M. Chen and Z. Luo, "A virtual reality system for arm and hand rehabilitation," *Frontiers of Mechanical Engineering,* vol. 6, no. 1, p. 23–32, 2011.

[18] L. Dipietro, A. M. Sabatini and P. Dario, "Evaluation of an instrumented glove for hand-movement acquisition," *Journal of Rehabilitation Research and Development,* vol. 40, no. 2, p. 179—190, 2003.

[19] J. L. Hernandez-Rebollar, N. Kyriakopoulos and R. W. Lindeman, "The AcceleGlove: A Whole-Hand Input Device for Virtual Reality," in *ACM SIGGRAPH*, New York, 2002.

[20] H. Alirezaei, A. Nagakubo and Y. Kuniyoshi, "A highly stretchable tactile distribution sensor," *Humanoids,* 2007.

[21] D. P. J. Cotton, I. M. Graz and S. P. Lacour, "A Multifunctional Capacitive Sensor for Stretchable Electronic Skins," *IEEE SENSORS JOURNAL,* vol. 9, no. 12, pp. 2008-2009, December 2009.

[22] R. S. Dahiya, G. Metta, M. Valle and G. Sandini, "Tactile Sensing—From Humans to Humanoids," *IEEE TRANSACTIONS ON ROBOTICS,* vol. 26, no. 1, pp. 1-20, February 2010.

[23] F. Parvini, D. McLeod, C. Shahabi, B. Navai, B. Zali and S. Ghandeharizadeh, "An Approach to Glove-Based Gesture Recognition".

[24] J. L. Hernandez-Rebollar, N. Kyriakopoulos and R. W. Lindeman, "A New Instrumented Approach For Translating American Sign Language Into Sound And Text," in *Sixth IEEE International Conference on Automatic Face and Gesture Recognition*, 2004.

[25] G. t. Holt, P. Hendriks and T. Andringa, "Why Don't You See What I Mean? Prospects and Limitations of Current Automatic Sign Recognition Research," *Sign Language Studies,* vol. 6, pp. 416-437, 2004.

[26] S. C. Ong and S. Ranganath, "Automatic Sign Language Analysis: A Survey and the Future beyond Lexical Meaning," *PATTERN ANALYSIS AND MACHINE INTELLIGENCE,* vol. 27, no. 6, pp. 873 - 891, June 2005.

[27] P. Dreuw, J. Forster, Y. Gweth, D. Stein and H. Ney, "SignSpeak – Understanding, Recognition, and Translation of Sign Languages," 2010.

[28] G. T. Holt, P. Hendriks and T. Andringa, "Why Don't You See What I Mean? Prospects and Limitations of Current Automatic Sign Recognition Research," *Sign Language Studies,* vol. 6, no. 4, pp. 416-437, 2006.

[29] M. W. Kadous, *Australian Sign Langueage Dataset.*

[30] D. Floreano and C. Mattiussi, Bio-Inspired Artificial Intelligence, 1st ed., R. C. Arkin, Ed., London: The MIT Press, 2008.

[31] S. Haykin, Neural Networks and Learning Machines, 3rd ed., Ontario: Pearson, 2009.

[32] K. Murakami and H. Taguchi, "Gesture Recognition Using Recurrent Neural Networks," *Human Factors in Computing Systems,* pp. 237-242, 1991.

[33] P. Vamplew and A. Adams, "Recognition of Sign Language Gestures Using Neural Networks," *Intelligence Information Processing Systems,* vol. 5, no. 2, pp. 94-102, 1998.

[34] M.-C. Su, "A Fuzzy Rule-Based Approach to Spatio-Temporal Hand Gesture Recognition," *Systems, Man, and Cybernetics,* vol. 30, no. 2, pp. 276-281, 2000.

[35] M. B. Waldron and S. Kim, "Isolated ASL Sign Recognition System for Deaf Persons," *Rehabilitation Engineering,* pp. 261-271, 1995.

[36] C. Oz and M. C. Leu, "American Sign Language word recognition with a sensory glove using artificial neural networks," *Engineering Applications of Artificial Intelligence,* vol. 24, no. 7, pp. 1204-1213, 2011.

[37] S. Saengsri, V. Niennattrakul and C. A. Ratanamahatana, "TFRS: Thai finger-spelling sign language recognition system," *Digital Information and Communication Technology and it's Applications,* pp. 457-462, 2012.

[38] G. Fang, W. Gao, X. Chen, C. Wang and J. Ma, "Signer-Independent Continuous Sign Language Recognition Based on SRN/HMM," *Gesture Workshop,* pp. 76-85, 2001.

[39] C. Wang, W. Gao and S. Shan, "An Approach Based on Phonemes to Large Vocabulary Chinese Sign Language Recognition," *Automatic Face and Gesture Recognition,* pp. 393-398, 2002.

[40] C. Vogler, "American Sign Language Recognition: Reducing the Complexity of the Task with Phoneme-Based Modeling and Parallel Hidden Markov Models," University of Pennsylvania, Pennsylvania, 2003.

[41] Y. Chen, W. Gao, G. Fang, C. Yang and Z. Wang, "CSLDS: Chinese Sign Language Dialog System," in *Analysis and Modeling of Faces and Gestures (AMFG)*, Nice, 2003.

[42] M. H. Beale, M. T. Hagan and H. B. Demuth, "Mathworks," 2014. [Online]. Available: http://www.mathworks.com/help/pdf_doc/nnet/nnet_ug.pdf. [Accessed July 2014].

[43] A. A. Alhamud, "The use of a neural network to recognize placental insufficiency from blood flow velocity waveforms in the umbilical cord," Cape Town, 2004.

[44] R. Harris, "Youtube.com," 7 January 2012. [Online]. Available: https://www.youtube.com/watch?v=aVId8KMsdUU. [Accessed July 2014].

[45] Willamette University, "Willamette University," [Online]. Available: http://www.willamette.edu/~gorr/classes/cs449/momrate.html. [Accessed July 2014].

[46] O. Kisi and E. Uncuoglu, "Comparison of three back-propagation algorithms for two case studies," *Indian Journal of Electrical Engineering and Material Science,* vol. 12, pp. 434-442, October 2005.

[47] R. Gentner and J. Classen, "Development and evaluation of a low-cost sensor glove for assessment of human finger movements in neurophysiological settings," *Journal of Neuroscience Methods,* vol. 178, no. 1, p. 138–147, 2009.

[48] S. Wise, W. Gardner, E. Sabelman, E. Valainis and Y. Wong, "Evaluation of a fiber optic glove for semi-automated goniometric measurements," *Journal of Rehabilitation Research,* vol. 27, no. 4, pp. 411-424, 1990.

[49] L. K. Simone, N. Sundarrajan, X. Luo, Y. Jia and D. G. Kamper, "A low cost instrumented glove for extended monitoring and functional hand assessment," *Journal of Neuroscience Methods,* vol. 160, no. 2, p. 335–348, 2007.

[50] G. Saggio, "A novel array of flex sensors for a goniometric glove," *Sensors and Actuators A,* vol. 205, pp. 119-125, January 2014.

[51] G. Saggio, "Electrical Resistance Profiling of Bend Sensors adopted to Measure Spatial," in *University of Tor Vergata*, Rome, 2010.

[52] G. Saggio, "Mechanical model of flex sensors used to sense finger movements," *Sensors and Actuators A,* vol. 185, pp. 53-58, 2012.

[53] G. Orengo, A. Lagati and G. Saggio, "Bend Sensors Modeling for Fast Signal Recovering in Human Motion Analysis," SENSORDEVICES, Rome, 2012.

[54] Infinite Curiosity, June 2014. [Online]. Available: http://infinite-curiosity.com/.

[55] J. D. Kramer and C. Jacky, "How to write biblos," vol. 1, no. 1, 2006.

[56] Thibologa Sign Language Institution, July 2014. [Online]. Available: http://thibologa.co.za/.

9. Appendices

9.1 Appendix A – Gesture Targets

There are 31 static gestures that have been considered and used in order to develop the database. Each of these gestures have data that comes from the sensors. This is the input data that is used to train the neural network. The second component is the target information which lets the neural network know which gesture the data represents. This information has to be encoded using numbers so that it can be understood by the neural network. Each target was encoded as a single 1 in a vector of 31 zeroes. Each column represents one of the 31 gestures starting with the letter ‘A’ and ending with the number 9. The full list of encoded target values is given below:

$$A = [1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]$$
$$B = [0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]$$
$$C = [0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]$$
$$D = [0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]$$
$$E = [0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]$$
$$F = [0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]$$
$$G = [0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]$$
$$H = [0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]$$
$$I = [0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]$$
$$K = [0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]$$
$$L = [0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]$$
$$M = [0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]$$
$$N = [0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]$$
$$O = [0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]$$
$$P = [0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]$$
$$Q = [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]$$
$$R = [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0]$$
$$S = [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0]$$
$$T = [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0]$$
$$U = [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0]$$
$$V = [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0]$$
$$W = [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0]$$
$$X = [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0]$$
$$Y = [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0]$$
$$1 = [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0]$$
$$3 = [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0]$$
$$4 = [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0]$$
$$5 = [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0]$$
$$7 = [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0]$$
$$8 = [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0]$$
$$9 = [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1]$$

9.2 Appendix B - How to use the data glove prototype

This research project was focused on the back end of the system and no GUI was created for ease of use. The data glove prototype has been programmed to constantly stream the sensor data from the moment that it is powered up. There are two simple methods in order to gain access to the streamed data:

9.2.1 USB Tether

The data glove can be powered via a 5V FTDI. The FTDI cable must be connected in the correct orientation in order to power the Arduino Pro Mini. This cable acts as a means of communication for the device. The sensor data is streamed over serial communication. To view the data you must connect to the correct COM port. The simplest method of doing this is using a lightweight Telnet client such as PuTTY or one can use the Ardunio IDE and click on the Serial Monitor after selecting the appropriate COM port. Ensure that the baud rate is set to 115200 in order to view the data correctly.

9.2.2 Bluetooth

To wirelessly communicate the data the user must plug the LiPo battery into the fuel gauge. This will power the board. The data glove must now be linked to your PC via Bluetooth. Use your PC to search for Bluetooth devices in the area. The glove device will be found. Choose to connect to the glove using a password. When prompted enter the password of ‘1234’ and accept. The data glove should now be linked to your pc.

Once this is done the data glove can send the sensor data to your PC via Bluetooth. To view the data you must connect to the appropriate COM port in a similar manner to that of the USB tether method. The simplest method is to use PuTTY to view the data. Ensure that the baud rate is set to 115200 in order to view the data correctly.

The code for the data glove and all of the necessary libraries can be found on the attached CD (See Appendix C)

9.3 Appendix C – CD Contents

The following content can be found on the attached CD:

- Digital Copy of Master's Dissertation (PDF)
- Arduino IDE Software
- PuTTY
- Arduino Code used for during the research project
- Libraries used for this research project
- SASL Database raw and cleaned data
- MATLAB code and workspace

10. EBE Faculty: Assessment of Ethics in Research Projects

Any person planning to undertake research in the Faculty of Engineering and the Built Environment at the University of Cape Town is required to complete this form before collecting or analysing data. When completed it should be submitted to the supervisor (where applicable) and from there to the Head of Department. If any of the questions below have been answered YES, and the applicant is NOT a fourth year student, the Head should forward this form for approval by the Faculty EIR committee: submit to Ms Zulpha Geyer (Zulpha.Geyer@uct.ac.za; Chem Eng Building, Ph 021 650 4791).Students must include a copy of the completed form with the final year project when it is submitted for examination.

Name of Principal Researcher/Student: Ben McInnes **Department:** ELECTRICAL ENGINEERING

If a Student: YES **Degree:** Mechatronics **Supervisor:** Mohohlo Tsoeu

If a Research Contract indicate source of funding/sponsorship:

Research Project Title: South African Sign Language Dataset Development and Translation: A Glove-based Approach

Overview of ethics issues in your research project:

Question 1: Is there a possibility that your research could cause harm to a third party (i.e. a person not involved in your project)?	YES	**NO**
Question 2: Is your research making use of human subjects as sources of data? If your answer is YES, please complete Addendum 2.	**YES**	NO
Question 3: Does your research involve the participation of or provision of services to communities? If your answer is YES, please complete Addendum 3.	YES	**NO**
Question 4: If your research is sponsored, is there any potential for conflicts of interest? If your answer is YES, please complete Addendum 4.	YES	**NO**

If you have answered YES to any of the above questions, please append a copy of your research proposal, as well as any interview schedules or questionnaires (Addendum 1) and please complete further addenda as appropriate.

I hereby undertake to carry out my research in such a way that

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

Signed by:

	Full name and signature	**Date**
Principal Researcher/Student:	**Ben McInnes**	13 November 2014

This application is approved by:

Supervisor (if applicable):	**Mohohlo Tsoeu**	13 November 2014
HOD (or delegated nominee): Final authority for all assessments with NO to all questions and for all undergraduate research.	**Janine Buxey**	13 November 2014
Chair : Faculty EIR Committee For applicants other than undergraduate students who have answered YES to any of the above		

ADDENDUM 1:

10.1 Application to the EBE Ethics Committee for Ethics Approval 2014

Project Title: South African Sign Language Dataset Development and Translation: A Glove-Based Approach

10.2 Applicant Details

Surname:	**McInnes**	**Name:**	**Ben**	**Student #:**	**MCNBEN003**
Email Address:	**mcinnes.bs@gmail.com**			**Contact Number:**	**0721162406**
Qualification:	**BSc Eng Mechatronics**			**Date Registered:**	**10/02/2013**
Supervisor:	**M. Tsoeu**			**Department:**	**Electrical Engineering**

10.3 Research Background

According to the South African 2011 Census [1] there are over 51 770 people in South Africa that are completely deaf and 258 800 people with severe hearing loss. It has been recorded that roughly 258 800 people use South African Sign Language (SASL) in their homes. While South African Sign Language is not one of the Official South African languages, it has been formally recognised by the government. *"Sign Language has the status of an official language for the purposes of learning at public school"* (South African School Act 1996).

A digital South African Sign Language database currently does not exist. There are printed dictionaries and a small library of SASL videos but there are no available comprehensive databases available online. Without a database it is impossible to create a standard device that could be used as an Automatic Sign Language Translation for SASL and all of the research that would accompany such a device.

A secondary problem has also been noted, that data gloves, while available, are expensive. In this study we propose the development of a low-cost data glove device as well as the generation of the associated SASL database. This data glove must sense enough data to create a South African Sign Language Database from the ground up for research purposes. This device must then be able to be used for the reversed purpose of classifying and translating SASL when a SASL user gestures with the glove.

10.4 Testing Procedure

All of the testing procedures will be carried out in the Department of Electrical Engineering at the UCT Faculty of Engineering and the Built Environment. There are three main sets of data collected: (a) a volunteer questionnaire, (b) static gesture data capture and, (c) dynamic gesture data capture (if time allows). These tests will only take place once the participant has signed the consent form and completely understands the process.

10.4.1 Participants

Five to ten participants will be required for this test, with varying fluency in South African Sign Language in order to create a more robust dataset. The goal would be to have an even split of male and female volunteers as well as right and left-handed people. This would ensure variety in the data and would lead to a more robust system being developed. Depending on the budget allocated and the availability of the South African Sign Language translators it would beneficial to have more than one in the study to ensure the quality of the data and to set a benchmark to compare the other data to. It is not completely necessary to have Deaf people in this study as long as there is an even spread of participants' fluency and skill in South African Sign Language. A maximum of two professionals will be considered due to budget constraints, the rest will be volunteers.

The time required of each participant will roughly be 20 minutes. The participant will be required to wear a data glove in order to record the position, orientation and movement of their hands while performing a list of set gestures (both static and dynamic).The goal is to have a complete set of data gloves in order to record two handed dynamic gestures. Both left and right gloves will be used in capturing the data for the database.

10.4.2 Questionnaire

A short questionnaire will be given for basic information including hand and finger measurements and level of fluency of sign language. This is to be used as a reference in case of outliers within the data and to give explanations as to why certain data may be left out of the final dataset.

10.4.3 Static Gesture

Two sets of static gestures will be given to the participant; the first being a list of all of the South African Sign Language alphabet gestures and the second set will be a list of the most common hand shapes that make up the South African Sign Language gestures. The static gestures will be measured by a mixture of sensors. Commercial Flex Sensors will measure the movement and position of the participant's fingers and a 6 degree of freedom IMU will measure the orientation of the user's hand.

Once the participant feels confident in their ability to perform each of these gestures and hand shapes they will begin the test which will proceed as follows:

1. Open hand
2. Make the first static gesture on the list
3. Hold for 1 second
4. Release static gesture
5. Repeat 10 times
6. Move on to next gesture

Examples of static gestures can be seen below:

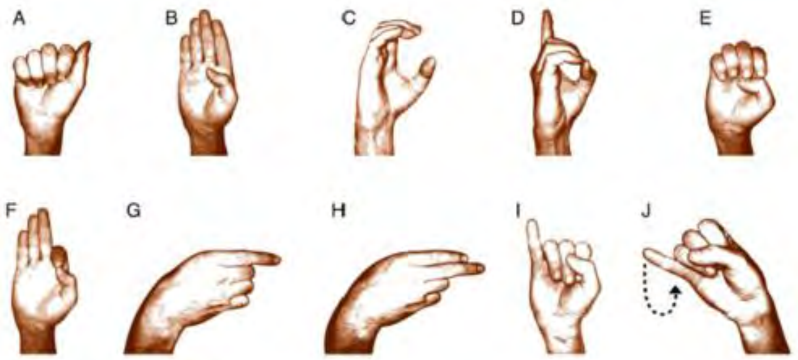


Figure 55: South African Sign Language Alphabet from A - J (Source: http://www.thibologa.co.za)

10.4.4 Dynamic Gesture

Two sets of dynamic gestures will be given to the participant. The first set will be made of simple hand movements (up, down, left and right), more complicated hand movements (short arcs, long arcs, zig-zags and repetitive motions) and finally basic shapes (circle, square and spiral). The second set of dynamic gestures will be the most commonly used South African Sign Language words.

Once the participant feels confident in their ability to perform each of these gestures they will begin the test which will proceed as follows:

1. Move hand to “home” position in front of them
2. Hold for 2 seconds
3. Perform the first dynamic gesture
4. Hold for 3 seconds
5. Move back to “home” position
6. Repeat 10 times
7. Move on to next gesture

Examples of dynamic gestures shown below:

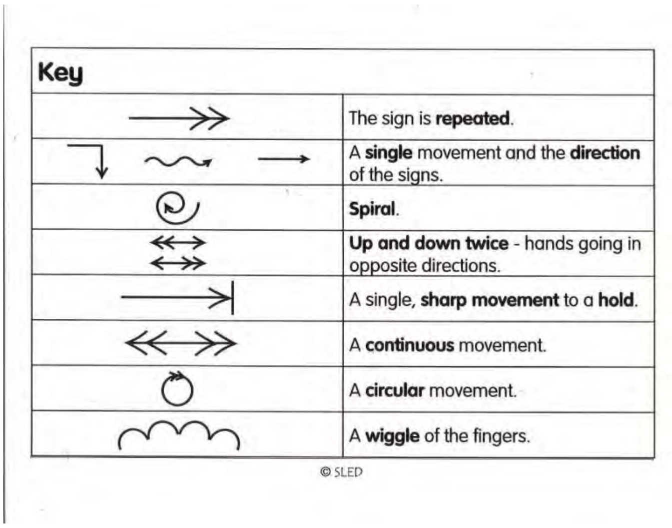


Figure 56: Source: Sign Language Education and Development, A South African Sign Language Dictionary for Families with Young Deaf Children. South Africa: Sign Language Education and Development.

10.5 Possible Risk

The recording equipment will be isolated from mains electrical supply; the system will be connected to a battery powered laptop during testing. The device will adhere to biomedical standards and the maximum electrical current used in the device will be well below the threshold (1mA) that could cause harm. Worst case scenario is that the participant will feel a slight tingle on their skin. This will be very unlikely because the design of the glove will insulate the user from the device and the user will not come into contact with any exposed wires during the test.

10.6 Data Usage

The data recorded from the aforementioned tests and the resulting database will be made public and distributed under the GPL (General Public Licence), so that other researchers may benefit from it. The recording of this data is purely to improve gesture recognition and communication between Deaf and hearing people. These datasets will be property of UCT, without any prejudice to the participants. The volunteers will be kept strictly confidential and will only appear as a number in classification analysis as well as any public documents resulting from this study.

10.7 Monetary incentive

There could monetary incentive for two reasons during this testing process:
The first being if professional South African Sign Language translators are not willing to donate their time to the testing procedure in which case a fee may be paid in order to compensate them for their time.

The second would be in the event that not enough participants have volunteered to create a comprehensive database in which case a small gift may be given to compensate for the participants time. Gifts such as chocolates or food vouchers to show appreciation for help with this study.
If any monetary incentives are required for this research it will purely be used to compensate for the participants time. It will not be used to coerce the participant into giving “better” data or any similar types of methods to gain better results.

10.8 References

[1] Statistics South Africa, "Census 2011," Statistics SA, Census Results 2011.

10.9 Statement of understanding and consent:

I confirm that I am 18 years of age or older and that the exact procedure, techniques and the possible complications of the above tests have been thoroughly explained to me. I am free to withdraw from the study at any time should I choose to do so. I understand that I may ask questions at any time during the testing procedure. I know that the personal information required by the researchers and derived from the testing procedure will remain strictly confidential and will only be revealed as a number in classification analysis. I have carefully read this form and understand the nature, purpose and procedures of this study. I agree to participate in this research project conducted by the Electrical Engineering Department of UCT.

Name of volunteer / guardian (if necessary): ______________________________

Signature: ______________________________

Name of Investigator: ______________________________

Signature: ______________________________

Date: ______________________________

10.10 Research Team:

Principal Investigator:	Mr B. McInnes (MSc Electrical Engineering UCT)
Supervisor:	Mr M. Tsoeu (Lecturer, UCT)

ADDENDUM 2: To be completed if you answered YES to Question 2:

It is assumed that you have read the UCT Code for Research involving Human Subjects (available at http://web.uct.ac.za/depts/educate/download/uctcodeforresearchinvolvinghumansubjects.pdf) in order to be able to answer the questions in this addendum.

2.1 Does the research discriminate against participation by individuals, or differentiate between participants, on the grounds of gender, race or ethnic group, age range, religion, income, handicap, illness or any similar classification?	YES	**NO**
2.2 Does the research require the participation of socially or physically vulnerable people (children, aged, disabled, etc) or legally restricted groups?	YES	**NO**
2.3 Will you not be able to secure the informed consent of all participants in the research? (In the case of children, will you not be able to obtain the consent of their guardians or parents?)	YES	**NO**
2.4 Will any confidential data be collected or will identifiable records of individuals be kept?	**YES**	NO
2.5 In reporting on this research is there any possibility that you will not be able to keep the identities of the individuals involved anonymous?	YES	**NO**
2.6 Are there any foreseeable risks of physical, psychological or social harm to participants that might occur in the course of the research?	YES	**NO**
2.7 Does the research include making payments or giving gifts to any participants?	YES	**NO**

If you have answered YES to any of these questions, please describe below how you plan to address these issues:

The issues have been addressed in the Ethics Application.

ADDENDUM 3: To be completed if you answered YES to Question 3:

3.1 Is the community expected to make decisions for, during or based on the research?	YES	NO
3.2 At the end of the research will any economic or social process be terminated or left unsupported, or equipment or facilities used in the research be recovered from the participants or community?	YES	NO
3.3 Will any service be provided at a level below the generally accepted standards?	YES	NO

If you have answered YES to any of these questions, please describe below how you plan to address these issues:

ADDENDUM 4: To be completed if you answered YES to Question 4

4.1 Is there any existing or potential conflict of interest between a research sponsor, academic supervisor, other researchers or participants?	YES	NO
4.2 Will information that reveals the identity of participants be supplied to a research sponsor, other than with the permission of the individuals?	YES	NO
4.3 Does the proposed research potentially conflict with the research of any other individual or group within the University?	YES	NO

If you have answered YES to any of these questions, please describe below how you plan to address these issues: