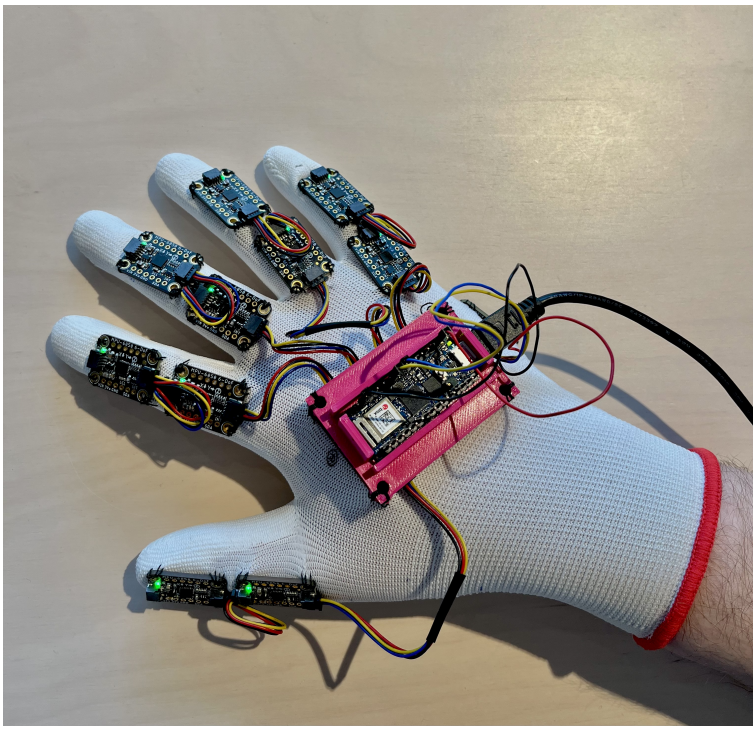
CHALMERS



Wearable sensor based system for measuring hand motions

Constructing a data glove capable of collecting motion data in real time

Bachelor's thesis in Systems and control

KALLE FUNCK, IDA HÄGERYD, MALIN JOHANSSON, JAKOB MALMER GÖRANSSON, JONATAN SVENSSON, LUDVIG SWALA

DEPARTMENT OF SOME SUBJECT OR TECHNOLOGY

CHALMERS UNIVERSITY OF TECHNOLOGY

Gothenburg, Sweden 2023

www.chalmers.se

Bachelor's thesis 2023

Wearable sensor based system for measuring hand motions

Constructing a data glove capable of collecting motion data in real time

KALLE FUNCK
IDA HÄGERYD
MALIN JOHANSSON
JAKOB MALMER GÖRANSSON
JONATAN SVENSSON
LUDVIG SWALA

CHALMERS

Department of Electrical Engineering
Division of Systems and Control
EENX16-23-14
Chalmers University of Technology
Gothenburg, Sweden 2023

Wearable sensor based system for measuring hand motions
Constructing a data glove capable of collecting motion data in real time

Supervisor: Rikard Karlsson, Department of Electrical Engineering
Examiner: Emmanuel Dean, Department of Electrical Engineering

Bachelor's Thesis 2023
Department of Electrical Engineering
Division of Systems and Control
EENX16-23-14
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Picture of the finished prototype.

Typeset in LaTeX
Printed by TeknologTryck
Gothenburg, Sweden 2023

Wearable sensor based system for measuring hand motions
Constructing a data glove capable of collecting motion data in real time
Kalle Funck, Ida Hägeryd, Malin Johansson, Jakob Malmer Göransson,
Jonatan Svensson, Ludvig Swala
Department of Electrical Engineering
Chalmers University of Technology

Abstract

The hand's incredible flexibility, dexterity and powerful grip makes it essential for performing most daily activities. This in turn means that suffering an injury to the hand that limits its fine motor skill is not just hindering, but affects the overall quality of life. A very important part of recovering from a hand injury is rehabilitation, as exercises often must be performed to regain functionality. In this thesis, a sensor glove capable of measuring the movement of the fingers was created. For this purpose, the possibility of using inertial measurement units to track the movement of the thumb was tested while also exploring if there are other more suitable sensors for tracking finger movements. The final prototype created in this project managed to track the movement of the fingers with varying accuracy. Ultimately, inertial measurement units were used for all fingers to achieve this.

The prototype works by sending data from the 6-DoF inertial measurement units to an Arduino microcontroller. The microcontroller then filters the data which can then be used to either calculate the angle between sensors, thus estimating the angles of the finger joints, or for visualization of the glove in a 3D environment. A calibration sequence was implemented in order to improve accuracy. When measuring the accuracy of the angle calculations from the prototype, it reached an accuracy of about 82%. The accuracy is based on the result when comparing the prototype to the validation system, which was not perfectly accurate. This leaves plenty of room for improvement in the form of implementing better sensor fusion filtering, a more accurate validation system, deployment of a magnetometer and more. This project concludes that inertial measurement units, combined with relevant software, are a viable option for detecting finger movements. However, the overall accuracy of this prototype was not improved compared to last year's project when looking at the results of performing sign language.

Keywords: IMU, system, glove, hand, motion, hand movement, wearable.

Sammandrag

Handens otroliga flexibilitet, fingerfärdighet och kraftfulla grepp gör den väsentlig för att utföra de flesta dagliga aktiviteter. Detta innebär i sin tur att en handskada som begränsar dess finmotoriska förmåga inte bara hindrar, utan påverkar den övergripande livskvaliteten. En mycket viktig del av att återhämta sig från en handskada är rehabilitering, då övningar ofta måste utföras för att återfå funktionalitet. I detta kandidatarbete skapades en sensorhandske som kan mäta fingrarnas rörelse. För detta ändamål undersöktes möjligheten att använda tröghetsmätenheter för att spåra tummens rörelse, samtidigt undersöktes andra sensorers lämplighet för syftet. Den slutliga prototypen som skapades i detta projekt lyckades spåra fingrarnas rörelse med varierande noggrannhet. I slutändan användes tröghetsmätenheter för alla fingrar för att uppnå målet.

Prototypen fungerar genom att skicka data från 6-DoF-tröghetsmätenheter till en Arduino-mikrokontroller. Mikrokontrollern filtrerar sedan datan som därefter kan användas för att beräkna vinkeln mellan sensorer, och därmed uppskatta vinklarna på fingerlederna, eller för visualisering av handsken i en 3D-miljö. En kalibreringssekvens har implementerats för att förbättra noggrannheten. Vid uträkning av noggrannheten för vinkelmätningarna från prototypen nåddes en noggrannhet på omkring 82%. Noggrannheten baseras på resultatet då prototypen jämförs med valideringssystemet, som inte heller har perfekt precision. Detta lämnar utrymme för förbättringar i form av implementering av bättre filtrering, ett mer exakt valideringssystem, användning av magnetometrar med mera. Detta projekt drar slutsatsen att tröghetsmätenheter, i kombination med relevant programvara, är ett gångbart alternativ för att detektera fingerrörelser. Den övergripande noggrannheten hos denna prototyp förbättrades dock inte jämfört med förra årets projekt när man tittade på resultaten av att utföra teckenspråk.

Nyckelord: IMU, system, handske, hand, rörelse, handrörelse, bärbar.

Acknowledgements

The members of this year's project group would like to thank the group working with this project last year for giving us the inspiration, as well as Rikard Karlsson and Emmanuel Dean for providing us with necessary help and guidance. Thank you.

List of Acronyms

Below, the list of the acronyms used throughout this thesis is listed in alphabetical order:

ASPRS	American Society for Photogrammetry and Remote Sensing
CM	Carpometacarpal
CNT	Carbon Nanotube
DIP	Distal interphalangeal
DMP	Digital Motion Processor
DoF	Degree of freedom
EKF	Extended Kalman filter
FFCB	Flexible composite board
HP	High pass
IDE	Integrated Development Enviroment
IMU	Inertial Measurement Unit
IP	Interphalangeal
IR	Infrared
KF	Kalman filter
LED	Light-emitting diode
LMC	Leap Motion Controller
LP	Low pass
MC	Microcontroller
MUX	Multiplexer
PIP	Proximal interpahalangeal
SCL	Serial Clock
SDA	Serial Data
ROM	Range of motion
sEMG	Surface electromyography
UI	Ultrasonic imaging

Contents

List of Acronyms **xi**

List of Figures **xv**

List of Tables **xvii**

1 Introduction **1**
1.1 Background . . . 1
1.2 Purpose . . . 2
1.3 Limitations . . . 2
1.4 Problem Specification . . . 2

2 Theory **5**
2.1 Anatomy of the hand . . . 5
2.2 Existing Solutions . . . 6
2.2.1 Machine learning solutions . . . 7
2.2.2 Carbon Nanotube-Based Strain Sensors . . . 8
2.3 Inertial Measurement Unit . . . 8
2.3.1 Gimbal lock . . . 9
2.3.2 Quaternions . . . 9
2.4 Sensor fusion . . . 10
2.4.1 Complementary filter . . . 10
2.4.2 Kalman Filter . . . 11
2.4.3 Madgwick . . . 13
2.4.4 Mahony . . . 13
2.5 IMU system overview . . . 14
2.5.1 Serial-chain network . . . 14
2.5.2 Multipoint network . . . 15
2.5.3 Number of IMUs . . . 16

3 Methods **17**
3.1 Requirements . . . 17
3.2 Hardware . . . 17
3.2.1 IMU . . . 18
3.2.2 Microcontroller . . . 18
3.2.3 Communication . . . 19
3.2.4 System overview . . . 19

3.3 Microcontroller software . . . 21
3.3.1 Sensor fusion filter . . . 21
3.3.2 Rotational errors . . . 21
3.3.3 Calibration . . . 23
3.4 Simulation software . . . 23
3.5 Validation system . . . 24
3.5.1 Photogrammetry . . . 25
3.5.2 Detection of camera parameters . . . 25
3.5.3 Stereo Calibration . . . 26
3.5.4 Landmark detection . . . 27
3.6 Testing . . . 28

4 Results **29**
4.1 Assembly . . . 29
4.1.1 Visual results . . . 31
4.2 Accuracy and speed . . . 31
4.2.1 Validation system . . . 33

5 Discussion **35**
5.1 Accuracy . . . 35
5.1.1 Testing . . . 35
5.1.2 Hand modeling . . . 36
5.1.3 IMU selection . . . 37
5.2 Opposition of the thumb . . . 37
5.3 System speed . . . 37
5.4 Flexibility of the glove . . . 38
5.5 Validation system . . . 39
5.6 Accessible software . . . 40
5.7 Ethics . . . 40

6 Conclusion **43**

Bibliography **45**

A Appendix 1 **I**

B Appendix 2 **III**

List of Figures

2.1 Schematic of the hand including the phalanges and different joints [11]. 5
2.2 Schematic of finger and thumb movement [13]. 6
2.3 Illustration of the different rotations on the IMU 10
2.4 Simplified overview of a complementary filter 11
2.5 Visualization of the statistical model of the Kalman filter 12
2.6 Diagrams comparing the different filters [46]. 14
2.7 Serial chain network 14
2.8 Multipoint network 15
2.9 Multiplexer 16

3.1 Arduino Nano RP2040 19
3.2 System overview. Describes placement of IMUs numbered from 1-10, microcontroller and multiplexer. 20
3.3 Image of the glove showing the twisted placements of the IMUs . . . 22
3.4 Backside of the designed hand in Unity 24
3.5 Triangulation with two cameras 25
3.6 Pictures of checkerboard in different positions 26
3.7 Explanation in which order the finger joints are stored [72]. 27
3.8 Landmark detection with glove on. 27
3.9 The Swedish sign language [82]. 28

4.1 Two MPU6050 placed on the middle finger 29
4.2 View of the IMU placements on the thumb 29
4.3 Plastic pocket above MUX with MC inside 30
4.4 The finish sensor glove with all of its components 30
4.5 Measurements from Arduino and camera validation system, no outer glove 32
4.6 Measurements from Arduino and camera validation system, with outer glove 32
4.7 Cardboard with known angle to find out most accurate position. . . . 33

A.1 Requirement Specification for Hand Motion Sensor Glove I

B.1 Preformed letters from the Swedish sign language, except y, å, ä and ö. Note that M was preformed incorrectly, it should be done using three fingers instead of four. VIII

List of Tables

4.1 The result of the accuracy test for the glove, performed without an outer glove. 31
4.2 The result of the accuracy test for the glove, performed with an outer glove. 31
4.3 Runtimes, both IMUs and also total system 33
4.4 Accuracy test on the validation system 33

1
Introduction

Healthcare plays an essential part in society and has a responsibility to preserve public health. In order to achieve this, many different kinds of physical and digital tools are used by medical staff. These can be as simple as a scalpel used to cut skin or a computer for updating patient records. In recent years, different technical appliances have become more common among healthcare equipment. Though these devices vary in how they are made and what purpose they fulfill, many rely on sensors [1].

Sensors are used to measure and keep track of a large number of different values such as blood pressure, oxygen saturation, muscle activation, and more [2]. It is with this information that medical staff can analyze a patient's health and take appropriate measures to ensure maintained well-being. These kinds of sensor-based systems have also started to branch over to rehabilitation [3]. This allows for collection of data that can be used to make rehabilitation more precise and personalized. However, there is a lack of equipment on the market that can produce data to be used for rehabilitation. There is much ongoing research in the area of rehabilitation and there is great potential for growth in addition to an unmet demand for rehabilitation services [2][4].

1.1 Background

The hand is a fundamental part of everyone's life as it allows for easy manipulation of objects in our surroundings. Its wide functionality begins at the 27 bones and 35 muscles that work together to bring about 25 degrees of freedom (DoF) for volitional movement [5]. The hand's incredible flexibility, dexterity and powerful grip make it essential for performing most daily activities. This in turn means that suffering an injury to the hand that limits its fine motor skill is not just hindering, but affects the overall quality of life [6]. It is not surprising that hand surgery has become its own specialty as it requires a multidisciplinary approach that combines knowledge from several medical fields [7]. This includes general surgery, plastic surgery, orthopedic surgery, vascular surgery and neurosurgery.

A very important part of recovering from a hand injury is rehabilitation as exercises must be performed to regain functionality [8]. However, knowing how much effort is enough can be hard. In the case of hand surgery involving repairment of the tendons, too much effort during exercises can rupture those tendons. For the purpose of assisting with rehabilitation, there is research that explores the utilization of hand rehabilitation devices [5][8]. These devices are gloves or exoskeletons

that collect hand movement data and assist movement. The data can be used by professionals to more closely follow the patient's progress and give assurance that the exercises are being done correctly. If proven effective, such an instrument could be used to streamline the entire rehabilitation process. But for this to be possible, the product must be accurate, cost-effective and user-friendly.

In a previous Bachelor's thesis exploring hand motion sensors, flex sensors were used to create a wearable system [9]. This system was capable of measuring the angles of the fingers, but had issues measuring the entire range of motion (ROM) of the thumb. They discussed the possibility of using Inertial Measurement Units (IMUs) to measure the movement of the thumb. This possibility, and the possibility of using IMUs on all fingers, will be further explored in this project. Other motion detection sensors will also be examined in order to investigate if there are other more suitable solutions than IMUs for the fingers.

1.2 Purpose

The purpose of this project is to, with last year's project as background, create a new sensor glove based on inertial measurement units for the thumb, while also exploring if there are other, more suitable sensors for tracking finger movements. Furthermore, this year's system will be evaluated, and then compared to last year's in order to ultimately be able to draw a conclusion about which type of sensor system works better for measuring hand motion. The end goal for this project is to have a solution that produces data from the fingers and thumb accurately enough to be an asset for occupational therapists.

1.3 Limitations

Since the project is time limited (18 weeks) and has a limited budget (5000 SEK), all possible solutions will not be fully researched or tested. This also implies that solutions that demand a lot of knowledge can not be examined. The sensor system will be tested in normal rooms, therefore there will be no tests on how the product copes with temperature, humidity, power consumption or durability. Also, there will not be a focus on making the solution comfortable or usable on different subjects. For instance, it is not guaranteed that the solution will work for every hand size.

1.4 Problem Specification

With the end goal described in section 1.2 it is important to develop a system that is accurate and fast while also not hindering hand movements. To be able to go through with this project the following questions below will serve as a guide for the project's progress:

- Are sensor systems that are based on IMUs, capable of detecting the complex movement of the thumb?

- What is a good and accurate way to validate the accuracy of the angles of the fingers and thumb?
- What can be done to ensure that the precision of the developed prototype can be deemed adequate?
- How can hardware and software be leveraged to be able to track at a satisfactory pace?
- How should the sensors be placed in order to minimize obstruction of hand movement?

2
Theory

To create a basic understanding of how to track the hand with IMUs, and if necessary in combination with other methods, the anatomy of the hand needs to be reviewed. The problem of tracking fingers has already been researched, and therefore existing solutions of hand tracking systems will be covered. Systems that are based on IMUs will be covered, but also different solutions that utilize other sensors. This will allow for this project to determine if solely IMUs will be utilized or if they should be combined or exchanged with other sensors. How an IMU-based network is built up will also be examined in order for this project to be able to create a prototype based on earlier research.

2.1 Anatomy of the hand

In order to understand this project it is crucial to have a solid understanding of the anatomy of the hand. First off, supported by the carpals, are the metacarpals which form the base of the fingers [10]. There are five metacarpal bones that make up the digits and each digit is then comprised of proximal phalanges, middle phalanges, and distal phalanges. An exception is the thumb that only has two phalanges, the distal and proximal. There are interphalangeal (IP) joints between the phalanges allows the finger to bend, acting like hinges, which can be seen in Figure 2.1.

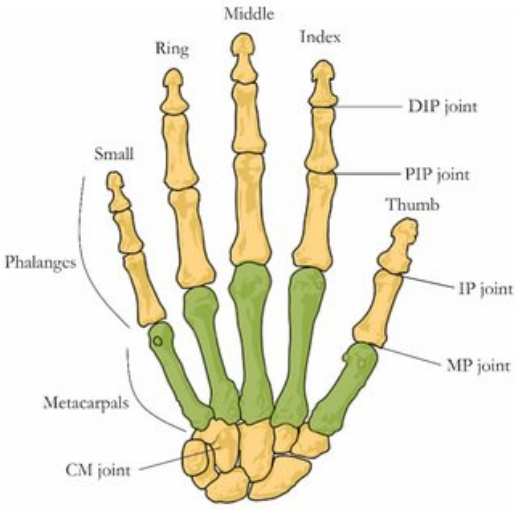


Figure 2.1: Schematic of the hand including the phalanges and different joints [11].

To allow for finger movement, including flexion and extension, two main muscles are in place, the flexor and extensor muscles [12]. These are located in the forearm connected to the flexor and extensor tendons. The flexor muscles allow the fingers to bend at the IP joints leading to a contraction and gripping motion of the fingers whilst the extensor muscles allow for extension.

The thumb is important for full hand functionality allowing us, for example to pinch and grasp objects and is supported by a significant portion of the intrinsic muscles [10]. There are four intrinsic muscles connected to the thumb, namely the adductor pollicis, abductor pollicis brevis, flexor pollicis brevis, and opponens pollicis and these combined with the carpometacarpal (CM) joint gives the thumb a wider ROM compared to the other digits. This includes abduction, adduction, flexion, extension, opposition and retroposition, which can be seen in Figure 2.2.

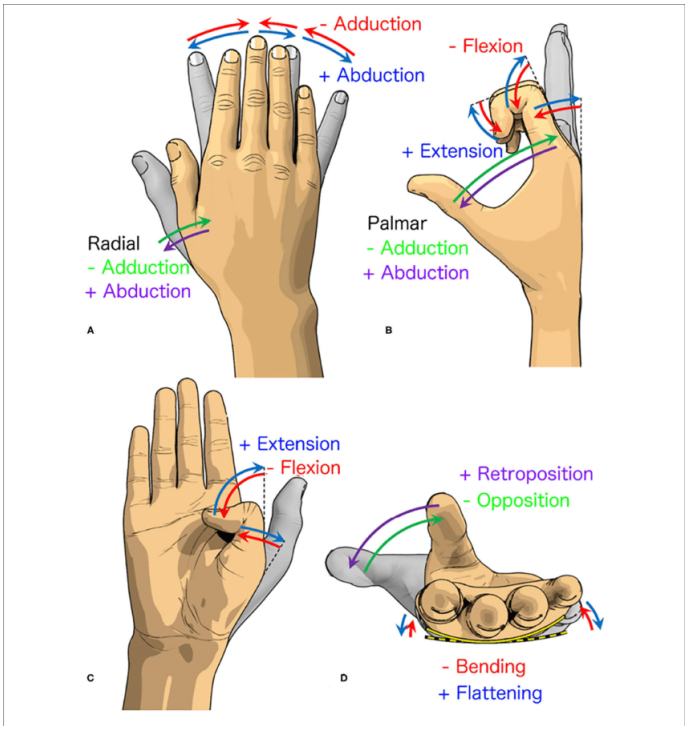


Figure 2.2: Schematic of finger and thumb movement [13].

2.2 Existing Solutions

In this section, possible solutions that could be used instead of IMUs on the fingers will be covered. Hand motion detection is not a new concept and there are several existing solutions already [14]. Examples of what a system can be based on are vision, bio-signals and sensors [15].

2.2.1 Machine learning solutions

There are several solutions to trace the hand that utilizes machine learning to later classify different gestures [16]. This then causes the system to classify gestures such as a closed fist or open fist. However, some of these solutions can not individually detect where the fingers are because they are not implemented for that purpose. Therefore the amount of gestures that a system can detect is limited [17]. The gestures are predetermined in order for the machine learning to be able to classify and the number of gestures varies between different solutions. It also requires lots of data processing which can produce a slow classification-based system [18]. These systems can utilize different sensors such as cameras and different bio-signals.

There are solutions where computer vision are preferred. The big upside is that it will not obstruct any movements [19]. It does nevertheless require movements to occur in front of the user and movements that do not involve hand motions in front of the camera will lower the accuracy [20]. However, there are visual solutions that can track individual coordinates of points of fingers. This is a big upside and therefore the amount of gestures is not limited. These solutions have been implemented with the help of machine learning [21].

Surface electromyography (sEMG) can also be an interesting way of classifying gestures using machine learning. It works by placing a surface EMG sensor around your wrist which measures electric signals produced by the muscles. Then machine learning is used to classify the movement that is done [22]. One example of this was done, but got a response time of about 227.76 ms while also receiving a gesture classification accuracy of 98.7%. EMG signals generally include noise and the produced data is limited, generating overfitting problems [23]. Also, for each patient new electrodes would have to be placed. This increases modularity, as everyone's hands are different, but limits simplicity and practicality.

Another way of using machine learning is by using Ultrasonic imaging (UI). It works by placing a ultrasonic probe on the forearm or wrist. The probe sends out ultrasounds and then measures the sound bouncing back which is then used to paint a picture of the muscles and bones. Machine learning can then use these images to classify the gestures done by the hand [24]. UI can be used to perform continuous classification and reach a precision of 96.5%. It also provides a good potential with finger gestures in the future [25]. UI is also safe and does not constrain hand movements which makes UI comparable to EMG. A difference between these is that UI only requires a small contact area in contrast with EMG which requires several electrode placements on the forearm [24]. UI has also proven to outperform EMG when comparing wrist movements [26]. A problem with UI is that the probe requires a medium between the skin and itself. This medium is most often a gel that might not be suitable for daily usage [24]. Nevertheless, there are solid coupling mediums that do exist and even outperform liquid gel while still being cost-effective [27]. Small portable ultrasound devices do exist and can image with preferable quality [28]. However, these can become quite costly. One of the cheapest ones found cost 1600 GBP and had an extra 300 GBP annual cost [28]. The price is expected to decrease with time as the cost of individual components is rather cheap [29].

Because of that classification-based systems can become quite slow and the number of gestures is limited these solutions will not be examined further during this

project.

2.2.2 Carbon Nanotube-Based Strain Sensors

One main issue with last year's glove was the precision of the thumb. Studies has shown that a Carbon Nanotube-Based Strain Sensors (CNT-based strain sensor) could improve the measures of the oppositions of the thumb [30]. This kind of sensor base its measurements on the changes in electric resistance, allowing to evaluate the length. Since the CNT is soft and elastic, it can be placed on surfaces where the skin stretches, such as curved and movable parts. Furthermore, it is small, thin, and accurate. Therefore, the device does not limit the hand movement. The CNT is also low cost and easy to use, which makes it affordable and available for people with limited knowledge and experience to use this system. Using CNT has been found to be sufficiently accurate, with a moderate accuracy.

However, there are some limitations [30]. First of all, this type of sensor only measures single-axis movements, and as the pronation angle of the thumb moves in three axes, it cannot be measured with only one of these sensors. Moreover, changes of dimensions of the hand affects the sensor signal.

Although, this type of sensor has some significant limitations, because of the low price and easy use, it can still be interesting to study together with other sensors to be able to measure all the angles. However, because this sensor is at the research stage, it is not possible to purchase and will therefore not be tested.

2.3 Inertial Measurement Unit

Inertial Measurement Unit (IMU) is the sensor proposed to be used for the thumb from last years project. It is a type of electronic sensor that returns the position and angular velocity of the joint measured in three dimensions [31][32]. The IMU typically consists of three gyroscopes, three accelerometers, and sometimes also three magnetometers [33]. The accelerometer measures the translational acceleration, while the gyroscope measures the angular acceleration [32]. IMUs are often used for motion detection of different parts of the body, for example hand motion detection, but also generally for tracking and navigation [34][35].

For data gloves, IMUs have been found to be one of the most practical sensors to use because of their lightness and great precision at catching motions of the fingers [36]. In 2017, a data glove was constructed by using a 6-axis IMU-sensors to keep the flexibility and modulation of the glove and keep the magnetic environment from affecting the measurement as these IMUs do not use magnetometers [14][36]. However, a problem with not using the magnetometers is that the ROM only can be measured at specific settings which increases the errors over time [36]. Furthermore, a problem found in the same research was that if only one IMU was placed at the back of the hand, the ROM of the metacarpophalangeal joints of the thumb and the little finger were not as accurate as the other fingers. Another study done in 2018 suggested 9-axis IMUs to overcome the limitations of the data glove previously mentioned. This glove was possible to make both flexible and modular, while having a mean error of under $\pm 3^\circ$ in motion.

The ROM calculation depends on two sensors placed proximal and distal from the measured joint [14]. This constraint makes it necessary to use multiple sensors at a time, which creates a complex and challenging IMU calibration procedure. One disadvantage with the IMU is that it has accumulated errors, caused by the fact that it constantly measures changes that are subject to floating-point precision problems [37]. When this precision problem happens for longer periods, the errors accumulate and are no longer neglectable. This depends on the inertial sensors, and the only ways to improve this accuracy to choose a good inertial sensor or by using an effective filter [37][38]

2.3.1 Gimbal lock

There are several methods for calculating angles in a 3D-space, where one of them is by using Euler Angles [39]. The principle behind the Euler angles is quite simple, but using them might cause an extensive problem called Gimbal lock. This happens when two of the three axes in the system align [40]. When this occurs the movement around one of the axis is lost, the movement of the whole system gets limited and one degree of freedom gets lost. The problem with gimbal lock can be solved for specific movements by changing the order of the movements of the axis, for example movement of the x-axis before the z-axis instead of the other way around. This does not make the gimbal lock go away and it will always appear when the middle axis rotates 90 degrees. To avoid this problem, a mathematical construct referred to as quaternions can be used instead [41].

2.3.2 Quaternions

Quaternions have the possibility for measuring spatial rotations [42]. A quaternion is referred to as $\mathbf{q}$ and is a type of hypercomplex number that consists of a real part and three complex parts. The way a quaternion is constructed can be seen in equation 2.1

$$\mathbf{q} = q_0 + q_1\mathbf{i} + q_2\mathbf{j} + q_3\mathbf{k} \tag{2.1}$$

q_0, q_1, q_2 and q_3 are real numbers, whereas $\mathbf{i}$, $\mathbf{j}$ and $\mathbf{k}$ are imaginary vectors placed orthogonal from each other. To be able to use the quaternions to calculate the ROM, they are needed to be converted to Euler Angles [14]. This is done by using three different equations, depending on which type of rotation that is desired. The different rotations are called yaw, pitch and roll. The calculations for yaw, pitch and roll can be found in equation 2.2, 2.3 and 2.4 [43]

$$\text{Roll} = \operatorname{atan2}\left(2\left(q_0q_1 + q_2q_3\right), 1 - 2\left(q_1^2 + q_2^2\right)\right) \tag{2.2}$$

$$\text{Pitch} = \operatorname{asin}\left(2\left(q_0q_2 - q_3q_1\right)\right) \tag{2.3}$$

$$\text{Yaw} = \operatorname{atan2}\left(2\left(q_3q_0 + q_1q_2\right), 1 - 2\left(q_2^2 + q_3^2\right)\right) \tag{2.4}$$

Yaw is the movement around the z-axis, pitch is along the y-axis, and roll around the x-axis. In Figure 2.3 the different rotations of the IMU are shown.

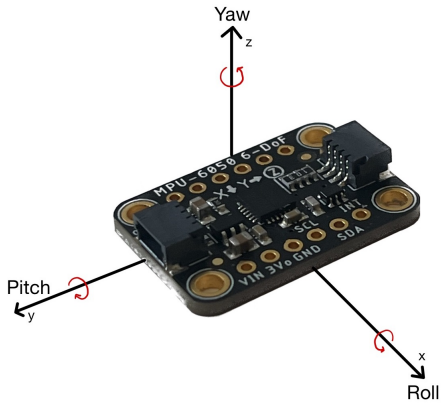


Figure 2.3: Illustration of the different rotations on the IMU

The quaternions q of the IMUs current location is calculated with equation 2.5, where ω_t is the angular velocity, Δ_t represent the sampling period, and q_{t-1} is the previous estimated quaternion [14].

$$q = q_{t-1} + \left(\frac{1}{2} q_{t-1} \times \omega_t\right) \cdot \Delta_t \tag{2.5}$$

To rotate a quaternion with another quaternion, quaternion multiplication can do it [44]. To rotate q_1 around q_2 the following equation is used.

$$q_{1,rot} = \frac{q_1 * q_2}{|q_1||q_2|} \tag{2.6}$$

To calculate a relative quaternion, equation 2.7 can be used. In combination with 2.2, 2.3 and 2.4, the angles between quaternions can be obtained.

$$q_{relative} = q_2 q_1^{-1} \tag{2.7}$$

2.4 Sensor fusion

When multiple sensors are used there is a need for an algorithm that can fuse their data together. A successful algorithm opens up the possibility to turn the data from the gyroscope, accelerometer and possibly magnetometer into a model of orientation [45]. There are multiple sensor fusion algorithms that can determine the orientation of an object measured with IMUs [46]. Examples of these are the Kalman filter (KF), the extended Kalman filter (EKF), the Mahony filter, the Madwick filter, and a complementary filter. There are multiple papers comparing these filters, and to get a better understanding of what filter will suit this project best it is also necessary to know how they are implemented.

2.4.1 Complementary filter

This method utilizes a high-pass (HP) filter on the gyroscope and a low-pass (LP) filter on the accelerometer. It allows the accelerometer to not output sudden spikes

and therefore the accelerometer will track the long-term changes in the orientation. The high-pass filter does the complete opposite, low changes are filtered out and this is to prevent the gyroscope drift that occurs otherwise. The gyroscope tends to have a lot of noise that wants to be filtered out, which is achieved with a HP-filter. Adding together the estimations from the LP-filter and HP-filter then results in an estimation of the angular speed of the plane. The Euler step method is then used to go from the speed estimate to a total estimation of the angle, which can be seen in equation 2.8.

$$\theta_{k+1} = \theta_k + \dot{\theta}dt \tag{2.8}$$

Figure 2.4 visualize how the complementary filter looks as a block diagram.

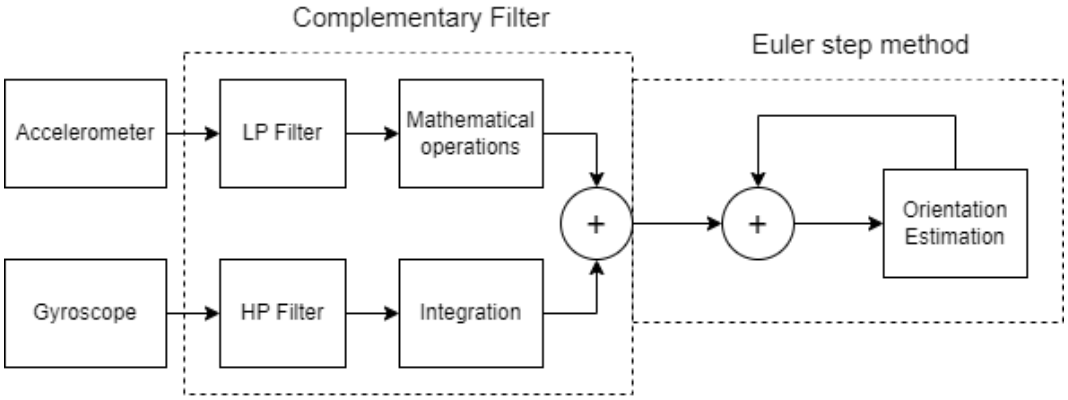


Figure 2.4: Simplified overview of a complementary filter

This filter has a low computation burden but requires the user to have experience with it when determining the filter constants [47]. Without a magnetometer, it performs poorly in terms of precision when trying to estimate a floating point compared to the filters mentioned above [46].

2.4.2 Kalman Filter

The Kalman filter (KF) is one of the most common sensor fusion algorithms [48]. The algorithm is used to predict the future by utilizing both old estimates and new information which include uncertainties and noise. For each new prediction of the future, the present state is used to aid the new estimate, which means the algorithm is recursive. The KF modeling manages to take system dynamics into account for linear systems, but for a non-linear system, the filter needs to be expanded into an extended Kalman filter (EKF) [49]. Two main components of the KF algorithm are the prediction step and the update step.

A KF takes two or more predictions of what the next state will be [50]. These predictions are then compared and fused together to create a statistically optimal prediction of the real state. This causes KF to be useful when there are multiple sensors since it can predict the status from multiple sources. Kalman filters tend to outperform the others in terms of precision but require more processing power and memory [51]. A visualization of the statistical model can be seen in Figure 2.5.

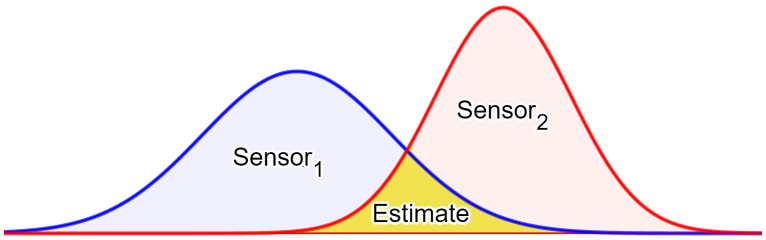


Figure 2.5: Visualization of the statistical model of the Kalman filter

The process model in equation 2.9 describes how the state evolves from time k-1 to k [49].

$$x_k = Fx_{k-1} + Bu_{k-1} + w_{k-1} \tag{2.9}$$

Here, x_k is the state vector and x_{k-1} is the previous state vector [49]. F is the state transition matrix applied to the earlier state and B is the control input matrix. u is the control vector and w is the process noise vector that is presumed to be Gaussian with the covariance Q.

The measurement model from equation 2.10 describes the relationship between our states and our measurements.

$$z_k = Hx_k + v_k \tag{2.10}$$

Here, z_k is the measurement vector, H is the observation model and v is the observation noise.

The KF needs to be initialized by inferring an initial state estimate and error covariance matrix [49]. The initial state is typically represented by a Gaussian-distributed random variable with a known mean and covariance [52]. If there is no covariance data, a common method is to choose a covariance matrix such that the variance initially becomes large. This will provide a quicker convergence toward more accurate estimates [49]. The method is called “initial ignorance”. There are also methods to initialize the KF without any grasp of the initial state and with no needed guesswork which could be helpful if only a few measurements are available [53].

$$\hat{x}_k^- = F\hat{x}_{k-1}^+ + Bu_{k-1} \tag{2.11}$$

$$P_k^- = FP_{k-1}^+F^T + Q \tag{2.12}$$

In the prediction step, the state is estimated with 2.11 and the covariance is predicted with 2.12 [49].

$$\tilde{y}_k = z_k - H\hat{x}_k^- \tag{2.13}$$

$$K_k = P_k^- H^T (R + H P_k^- H^T)^{-1} \tag{2.14}$$

$$\hat{x}_k^+ = \hat{x}_k^- + K_k \tilde{y} \tag{2.15}$$

$$P_k^+ = (I - K_k H) P_k^- \tag{2.16}$$

The measurement residual is given by 2.13. The residual is the difference between the actual measurement and the estimated measurement. The Kalman gain, a mean squared error minimizer, is calculated in 2.14 and later multiplied with the residual in 2.15 to obtain an updated state estimate [54]. In 2.16, the error covariance is corrected.

2.4.3 Madgwick

The Madgwick filter was proposed in 2010 and is specifically designed for IMU orientation tracking [55]. It is an extension of a complementary filter that takes the use of gradient descent. This in turn compensates for the drift of IMUs. Firstly a unit quaternion is generated and is then altered for each measurement [56]. The gyroscope provides an estimation for rotation after integration. Then the accelerometer measures the gravity vector. The two evaluations are subtracted from each other to generate one estimated gradient and one measured gradient. These are fused and weighted together to give a final estimation of orientation. Mathematically it can be represented as follows [57].

$$q_k = q_{k-1} + (\dot{q}_{\omega,k} - \beta \dot{q}_{\epsilon,k}) \Delta t \tag{2.17}$$

In equation 2.17 k represents the discrete time unit. Quaternions are represented by q. The estimated change, $\dot{q}_{\omega,k}$ is subtracted with the error gradient $\dot{q}_{\epsilon,k}$. The convergence rate is decided by the constant β. An increase of β causes the accelerometer to be more prominent compared to the gyroscope. Since the accelerometer is responsible for long-term changes the short-term changes will not be taken into account if β is large.

Wearable robotics and other systems with limited processing power are examples of suitable implementations of the Madgwick filter [58]. Compared to other open-source AHRS algorithms, the Madgwick filter has been revealed to be the best one.

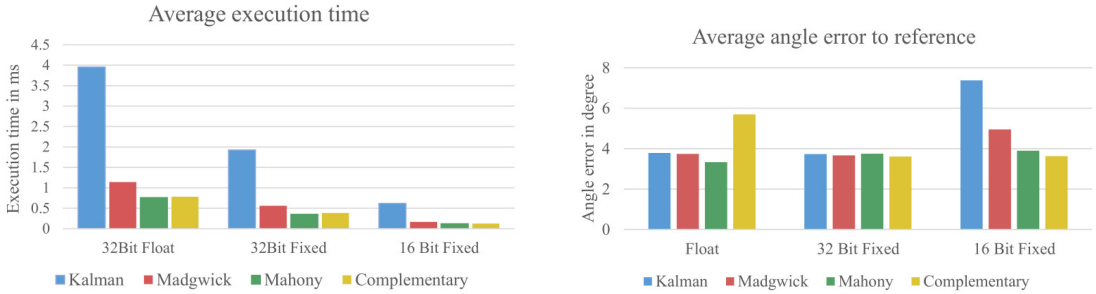
2.4.4 Mahony

The Mahony and Madgwick filter behaves and performs quite similarly in terms of processing power required and accuracy. [59]. The Mahony filter is also based on a complementary filter like the Madgwick filter. The one big difference between these two filters is that the Mahony takes in integral errors while the Madgwick filter only uses a proportional error compensation. The filter can be viewed as a discrete PI controller which mathematically is described by the following equations [46].

$$e_i = \sum e_k \, K_i \, dt \tag{2.18}$$

$$\theta_{corrected} = \theta + Kpe_k + e_i \tag{2.19}$$

The upside with both the Mahony and Madgwick filter is that they do not require as much computational capability while also having comparable precision to the Kalman filter. The Madgwick filter tends to have a slightly longer computation time compared to the Mahony filter as can be seen in figure 2.6a. The performance differences are illustrated in figure 2.6b.



(a) Execution time.

(b) Error estimation.

Figure 2.6: Diagrams comparing the different filters [46].

.

2.5 IMU system overview

To decide how this project's hardware should be implemented, existing systems are examined. When implementing a prototype it is necessary to know how the network should be implemented and where the IMUs should be placed. There are numerous ways to gather data from multiple sensors simultaneously. Two fundamental ways of doing this are via a serial-chain network or a multi-point network [60]. The system can also be a combination of these types of networks.

2.5.1 Serial-chain network

A serial-chain network is a network where each individual unit has its master and slave. This means that they all share the same communication line. In figure 2.7 an example of a serial-chain network is displayed.

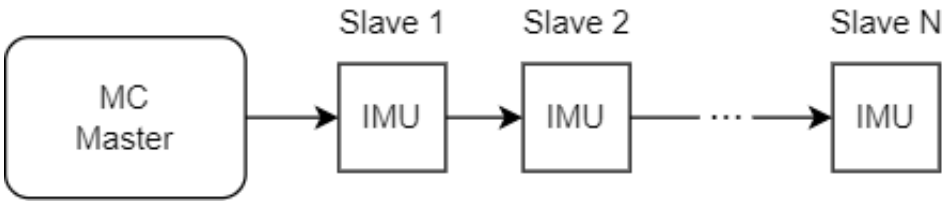


Figure 2.7: Serial chain network

There are examples of IMU-sensor systems that purely rely on a serial chain network [61]. The system traces arm movements but a serial-chain system has great adaptability. This system can easily add and move devices around the network which also means that IMU modules can be used for tracing hand movements in this network. A serial chain network on the hand would however have one problem with how the IMUs would be connected to one another since there would be a need to connect wires between fingers. This could lead to wires obstructing hand movements. The IMUs also need to have separate addresses to allow this to be possible. This is not the case with all IMUs on the market [62].

In comparison to figure 2.7 each slave in this particular system has a microcontroller (MC) connected to it [61]. An MC is a chip that combines a microprocessor, a specific quantity of memory, and various peripheral interfaces into a single structure [63]. Connected to the MC, the system does not have to send all data through the communication line since calculations can be done in every unit. The system has the ability to expand to track the entire range of body movements while also maintaining a high tracing frequency [61].

2.5.2 Multipoint network

A pure multipoint network is also possible to implement to trace the hand [39]. It can be able to track joint angles in real time with high precision. There are many examples of successful multi-point networks that trace the hand [36][64]. In Figure 2.8 an example of a multi-point network is displayed.

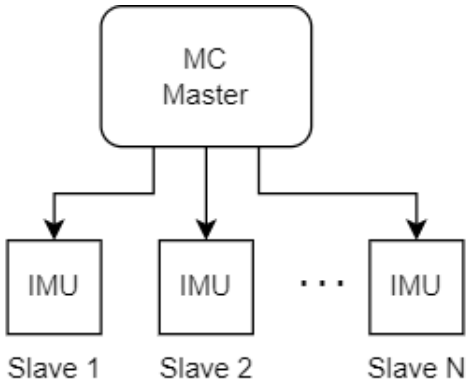


Figure 2.8: Multipoint network

A common way to implement this network type is by utilizing a multiplexer (MUX) [36]. A multiplexer gathers multiple sensor outputs and can then select which of the data to send down the output depending on the control signal's binary value. Therefore it is possible to look at multiple signals through the same communication link, which can be useful if there are more inputs than the microcontroller has I/Os. The basic structure of a multiplexer can be seen in Figure 2.9.

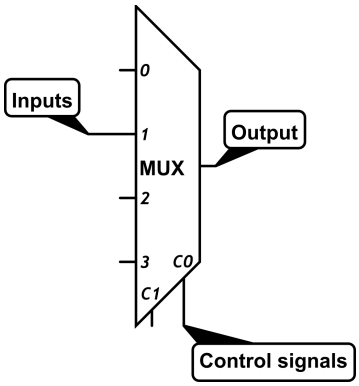


Figure 2.9: Multiplexer

2.5.3 Number of IMUs

The number of IMUs used varies quite a lot between different solutions. One prototype uses only five IMUs in total [64], while another prototype had one IMU placed on each distal phalange and one reference IMU on the back of the hand. The latest example causes the prototype not to be able to measure all joint movements, but since the purpose of the glove was to be able to classify French sign language letters it did not matter. The prototype got a 92% classification accuracy.

On the other hand, there are solutions that uses up to 18 IMUs [36]. In this particular example, three IMUs were placed on each finger between each joint and three on the back of the hand. The purpose of this glove was to do real-time tracking of the hand and also succeeded in calculating the angles of the fingers, the knuckles and the wrist. Depending on what each implementation wanted to achieve the number and placement of IMUs differed.

The distal interphalangeal (DIP) joint and proximal interpahalangeal (PIP) joint are correlated and therefore if the orientation of one of them is known the other could be estimated [65]. Hence it might not be necessary to have IMUs placed on both the distal and intermediate phalanges of the fingers while still managing to estimate the orientation of distal phalanges.

3
Methods

In this chapter, the requirements that the product must meet will be described, as well as all the hardware and software used to assemble it into a working prototype. The resulting validation system created will also be described.

3.1 Requirements

When creating the prototype the limitations and requirements derive from the problem specification in 1.4. For a final product to be used by occupational therapists, three main things must be kept in mind. It has to be accurate, have a decent response time and also be user-friendly. To ensure that the final product encapsulates the desired aspects, a full requirement list was defined and can be found in appendix A. Out of these requirements the most essential ones for this prototype are shown below.

- A response time less than 100 ms.
- A weight on the hand less than 100 grams.
- An emergency stop for cutting off power.
- Mean accuracy $< 5°$.

3.2 Hardware

The glove works by combining different electronic hardware, such as IMUs, microcontrollers and multiplexers. These components and their implementation are described below.

3.2.1 IMU

Based on last year's suggested improvements and the theory presented in chapter 2, it has been decided that solely IMUs will be used for the prototype. The different existing solutions that were explored were not better suited for this project compared to IMUs with the purpose and limitations of this project. There are IMU-based gloves and because of that it is possible to create one. This project's focus lies on improving the thumb but since it is possible to track the whole hand with IMUs this will also be done. The decision to track all fingers will save time since there will not be a need to merge different systems and sensors, which in turn allows this project to investigate IMUs more thoroughly. There is also a chance that IMUs perform better on the other fingers compared to the combination of flex sensors and potentiometers that was used last year [9]. This makes way for an interesting investigation if this is the case.

When choosing the IMU, several aspects needed to be taken into consideration. First of all the circuit board of the IMU needed to be small enough to fit on both the intermediate- and proximal phalanxes of every finger. That restricted the length and width to around 20 millimeters. The next aspect was the price. The total budget was set to 5000 SEK and therefore the price of the IMUs could not be too expensive. The plan was to use 10 individual IMUs and 1 IMUs that are combined with the microcontroller which meant that IMUs in a price range of 100 - 200 SEK were examined.

There are both IMUs with 6 and 9 degrees of freedom, and they have different pros and cons. Because of the higher price and the complexity of the 9 DoF IMUs, 6 DoF IMUs were chosen. This causes our sensors not to be able to tell absolute orientation around the yaw axis since there is no reference to how much it has moved. It then purely relies on the gyroscope when performing this rotation which causes drifting errors. The decision to skip out on a magnetometer might also speed up the system since less data is handled and fewer calculations are done when sensor fusion is applied [66].

The two IMUs examined were the MPU-6050 6-DoF Accelerometer and Gyroscopic Sensor Breakout from Adafruit, and the ST LSM6DSOXTR 6-axis IMU on the Arduino Nano RP2040 Connect. The MPU6050 has two STEMMA QT connectors which makes it easy to connect two MPU6050 with each other without soldering [67]. They also have a size of 25x17mm which in reality is a little bit too big, forcing the IMU to stick out of the finger, but it was found out that this was not a big problem since it did not hinder movements. The fact that the IMUs are able to be serially connected was a big benefit because this simplifies the process of designing the system.

3.2.2 Microcontroller

To be able to read and handle data, a microcontroller was needed. The Arduino Nano RP2040 is the model that was chosen for the controller since it has an integrated IMU, is reasonably priced, and is compact. The size of the components is crucial since they will be positioned on the glove, limiting the wearer's range of motion. The Nanos use two pins for I2C communication, one for SDA (Serial Data)

and one for SCL (Serial Clock). They are also compatible with the Arduino IDE (Integrated Development Environment). The Arduino Nano is displayed in Figure 3.1.

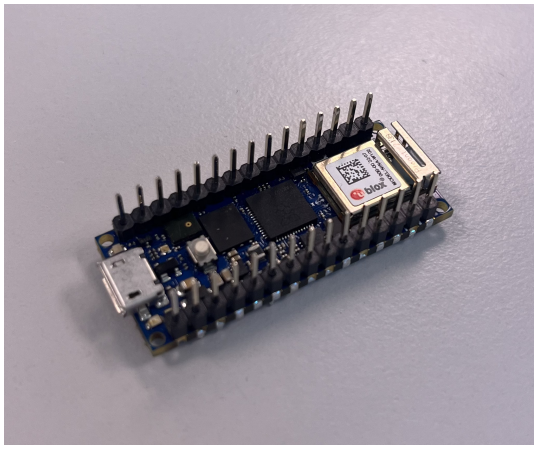


Figure 3.1: Arduino Nano RP2040

3.2.3 Communication

I2C connection is a method used for communication between multiple chips connected to at least one controller chip [68]. It is a useful protocol since it only requires two wires that can be shared between multiple devices limiting the amount of microcontroller pins necessary. However, one problem with I2C is that the devices have to have unique addresses.

To combat the issue of limited I2C addresses for IMUs, a multiplexer (MUX) was implemented in the system to collect the output from the sensors. The "SparkFun Qwiic Mux Breakout - 8 Channel (TCA9548A)" was used for this purpose. Most important was the fact that it had Qwiic connections since that is what is used for the IMUs. The MPU6050 had two possible addresses but the use of a MUX allowed for interfacing with more than two serial-connected IMUs in the Arduino IDE, which is essential for the final system. One drawback with wired communication is that cables did break quite often since the IMUs were very close to each other forcing the cables to bend.

3.2.4 System overview

In order to get a fully functioning system the placement of each component was carefully considered. The prototype was designed as figure 3.2 shows.

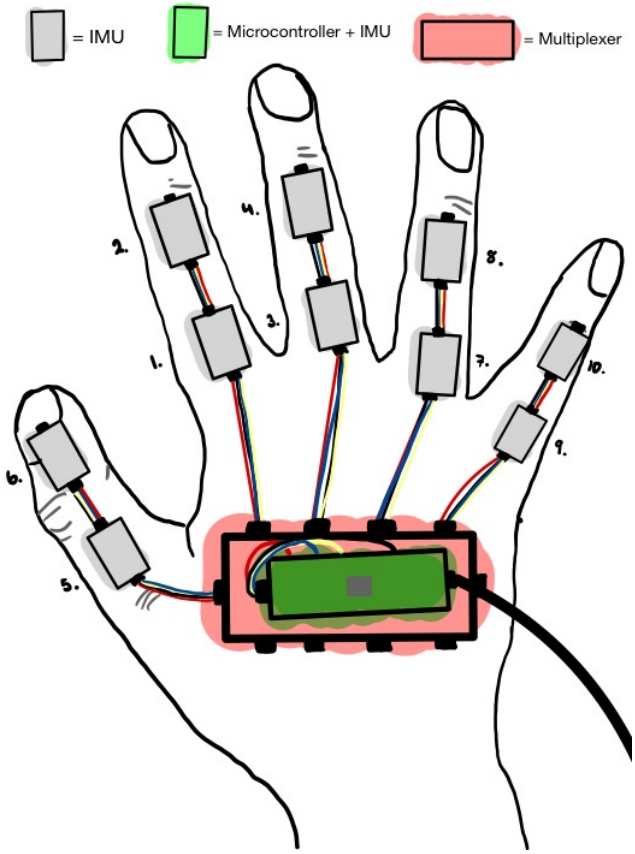


Figure 3.2: System overview. Describes placement of IMUs numbered from 1-10, microcontroller and multiplexer.

Firstly, two serial-connected IMUs were placed on each finger according to Figure 3.2. The placement did not hinder the user from moving the fingers and allowed for the detection of movement at the joints. The reason why the distal phalanges do not have any IMUs placed on them is because of that it is possible to estimate their orientation when the angle of the DIP joint is known, which is possible to calculate with IMUs placed on both intermediate and proximal phalanges [69]. Increasing the number of IMUs would also increase the iteration time and price of the system. With the number of IMUs placed on the hand the system should be able to track all movements of the fingers.

Secondly, the IMUs were connected to the MUX, which is connected to the microcontroller. This, as described previously, allowed for interfacing with all IMUs. Continuing, the chosen microcontroller had a built-in IMU which was used to detect the movement of the back of the hand. This is also the reference IMU meaning that other IMUs were looked at in how they related to the reference.

The drawbacks of this system are that it is not able to distinguish arm movements from wrist movements since there is no reference IMU on the forearm. There is also only one reference IMU on the back of the hand which is not optimal as presented in section 2.3. It was however decided to move on with the presented system design because it was easier to implement with the acquired hardware. The main hardware limitations that caused this were components being too big and not leaving enough

room on the back of the thumb. Also, cables would be forced to bend when IMUs were close to each other causing them to break. Therefore the place between the back of the thumb was not enough to place an IMU without having the cable rip often. Worth noting is that it is not possible to serially connect another MPU6050 on the thumb since two were already there. Forcing cables to also overlap if there is placed another IMU there which is not desirable.

3.3 Microcontroller software

Arduino IDE was used to write code scripts for the Arduino MC. The main purpose that the MC fulfilled was to process the sensor data from the IMUs and convert it to quaternions. This opens up the possibility to tell the orientation of each IMU.

3.3.1 Sensor fusion filter

For filtering, two different sensor fusion filters were utilized. The Arduino Madgwick library was downloaded for filtering the sensor data coming from the IMU on the Arduino Nano RP2040. This filter takes in the gyroscope and accelerometer values and returns yaw, pitch and roll. For visualization and constraint reasons, accessing the quaternions was necessary. The library did not have any functions for this purpose so the library had to be modified to fulfill this need. This was done by simply changing these quaternion variables from private to public variables.

Madgwick was chosen over the other algorithms because it has been proven to work well with low computing power and requires few tuning of parameters. For one microcontroller, there are up to 10 IMUs connected to this. Therefore a filter that could work with less computing power was desirable and because of that KF and EKF were not used. The Mahony filter seems like a reasonable choice but when working with both filters implemented in Arduino there was simply fewer problems and bugs occurring when working with Madgwick compared to Mahony. The complementary filter was not applied because its accuracy was too low.

It was decided to continue with the orientation algorithm contained in the MPU6050 library in Arduino for those IMUs. This algorithm is developed by Invensense and is not publicly disclosed. However, the digital motion processor (DMP) which exists on the MPU6050 does these orientation estimation calculations and then sends back quaternions to the microcontroller. Utilizing this processor allows the master MC to handle fewer calculations since the DMP takes care of some of the processing. Which in turn can speed up the process. The filter that is implemented in the MPU6050 DMP does have good accuracy of 0.1° in the roll and pitch measurements [70]. The filter also had no drift. Because of the high accuracy and the ability to split the computations into different sections the DMP filter was used.

3.3.2 Rotational errors

When wearing the glove it is impossible to hold it flat towards the earth. Partly, due to the anatomy of the hand, since the placement of the IMUs does not allow the hand to stay completely in line with one another. One more reason why it is

impossible to hold it flat is that the IMUs are not perfectly placed on the top of each phalanx. This can be viewed from figure 3.3 below.

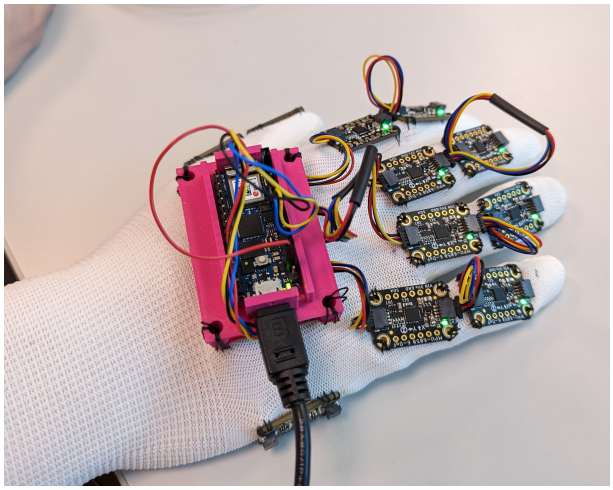


Figure 3.3: Image of the glove showing the twisted placements of the IMUs

This causes problems when trying to calculate the angles for validation since all have their own different coordinate systems based on how the hand holds when it starts. Meaning there is a lack of a global coordinate system and how they relate to one another. To compensate for this, two implementations were made.

Firstly the accelerometer got used to calculate the starting position. The user of the glove is forced to keep the hand still for a short period of time to make sure the accelerometer values are consistent. This gives pitch- and roll estimates through the following equations:

$$\begin{aligned} \theta &= arctan2(ay, az) \\ \phi &= arctan2(-ax, ay^2 + az^2) \\ \psi &= 0; \end{aligned} \tag{3.1}$$

These are then converted into quaternions through the inverse equations of 2.2 and 2.3. Yaw is impossible to measure without a magnetometer and is therefore set to zero during this starting phase. This is not optimal since the angle of the abduction will not get a starting value and therefore this type of correction for this dimension will get lost. However, for the reference IMU this does not matter since this project has no interest in what cardinal direction the hand points in.

The second thing that was done to correct the rotational errors was to calculate the rotational difference between each IMU and the reference IMU. This was done by placing the glove on a flat object and moving it around slowly. Equation 2.7 was then used to calculate the relative rotation from the MPU6050 plane to the reference.

After running this algorithm, the relative quaternion between the two IMUs would converge quite quickly. All relative rotations were then saved so the algorithm did not have to be executed every time the program starts. A problem that comes with this is that different persons will receive different values when running this algorithm. Therefore it is important to know that the values that are measured are

only true for one person and if others are to use the glove they should rerun the algorithm for it to be accurate. After the rotational quaternion was known it was used to rotate the quaternion of the IMUs to fit the plane in which it should be located if it was placed ideally.

3.3.3 Calibration

It was realized that each IMU comes with constant errors which were trivial to handle. Therefore it was implemented a function that demanded the user to hold the hand still and perpendicular to gravity. The raw data should then be 0 in all measurements except the accelerometer in the z-axis which should represent 1g. The constant offsets for each IMU were then calculated by taking the difference between the read values and the expected ones. After running the code for around 100 iterations the average offset converged and this value was then saved to be used.

To get the percentage measurement error of the gyroscope that came with the IMU, the IMU was aligned with the gravitational orientation and then rotated a full lap with the hand by also using a table as support. This was done for all three turns that were possible. The measured rotation was then converted to Euler angles and then divided by 360°. This was done a few times to make sure that it was precise. These errors were only ± 1% but taking these into account slightly improved the accuracy. This was only done for the reference IMU since the raw data of the MPU6050 was not read and therefore it was not found how to apply percentage offsets while also using the filter implemented in the DMP. Constant offsets were nevertheless possible to apply. The method to measure these error rates could be improved since using the hands would not result in perfect rotations around the gravity vector. However, these error rates were not very significant and therefore it was not something that was prioritized.

After calculating the offsets e for each IMU, Mathematically what was done with the raw data for all measurements of each IMU after the can be viewed in equation 3.2.

$$a_{x,true} = a_{x,read} \cdot e_{relative} + e_{constant} \tag{3.2}$$

3.4 Simulation software

Simulating the glove in a 3D environment is important for understanding the data that is retrieved from the IMUs. The game engine Unity was used for this purpose since it allows the creation of 3D objects and is considered easy to use for beginners. All code was written in C# using the Microsoft Visual Studio extension for Unity. With this, a graphical model was created to visualize how well the system tracks hand motion. The Arduino MC sends data in the form of quaternions to a script manager in Unity. This data is then distributed to scripts responsible for the movement of joints. This is a method for recreating the movements of the IMUs in the 3D environment. For the hand model, an Oculus hand model [71] was used because it had a well-designed thumb that accurately followed the movement done with the glove. This model can be observed in Figure 3.4 below.

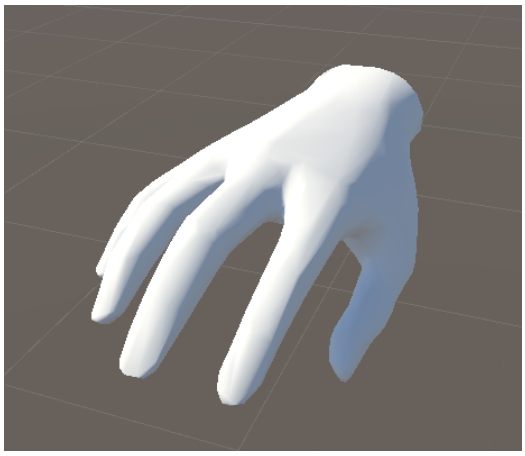


Figure 3.4: Backside of the designed hand in Unity

The finger joints of the hand model use C# scripts to rotate according to the quaternion data. Each finger is attached to the "Palm" object that can be made to follow the IMU on the MC. This enables rotations of the entire hand to be captured if appropriate input is retrieved from the Arduino.

3.5 Validation system

In order to confirm that the glove produces accurate data, a validation system was needed. A camera-based validation system was developed since it is dynamic and has the potential to validate complex movements. Some form of physical angle measurement system like a digital protractor lacks this potential. A method for validating the data is by using a different system with better accuracy and comparing the data given by the same hand movement.

The system considered for validation was a camera tracking system based on Mediapipe in Python [72]. The program is capable of detecting hand landmarks, such as finger joints, via regression [21]. This can in turn be used to calculate the angles of the finger joints.

Mediapipe has an average precision of 95,7% in palm detection [21]. Correctly identifying the palm is essential for finding landmarks of the hand. The mean regression error (normalized by palm size) when identifying landmarks is 13,4%. This means that the program slightly displaces the landmark which in turn will affect the angles calculated for the finger joints.

It is difficult to measure the finger angles with only one camera since it is hard to get an accurate estimation of the depth. This will cause problems since a good estimation of 3D coordinates is essential to measure the angles in different poses. Another problem with one camera is that there will be blind spots that can not be seen when the fingers are bending and blocking the sight of the camera. MediaPipe can through usage of multiple cameras detect landmarks, with the help of a method called photogrammetry [73]. This gives more than one perspective, in other words, there will be fewer blind spots and easier for the program to estimate 3D coordinates.

3.5.1 Photogrammetry

Photogrammetry is defined by the American Society for photogrammetry and Remote Sensing (ASPRS) as the following:
"The art, science, and technology of obtaining reliable information about physical objects and the environment, through processes of recording, measuring, and interpreting imagery and digital representations of energy patterns derived from non-contact sensor systems" [73].

It works because each camera gets its own 2D coordinates which means that 3D coordinates can be estimated with something called "Triangulation" if the cameras are positioned apart from each other as Figure 3.5 illustrates.

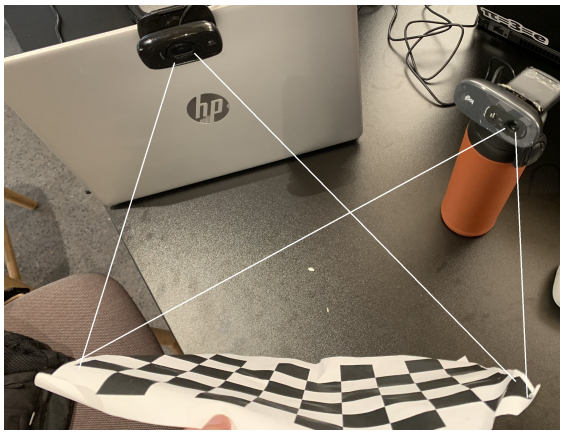


Figure 3.5: Triangulation with two cameras

The process involves identifying common points in multiple overlapping images from each camera and considering various camera parameters such as principal point, orientation, focal length, and lens distortion. By triangulating these common points, it is possible to calculate their 3D coordinates [74].

Once a common point is detected in at least two images taken from different known locations, imaginary lines can be drawn from the two camera positions toward that point. By mathematically determining the intersection of these lines, the 3D coordinates of the targeted point are obtained. With enough such points, a model of the scene can be constructed.

3.5.2 Detection of camera parameters

The first step in the process of photogrammetry is to estimate the instinct parameters of each camera.[75] This was done by taking multiple pictures of a checkerboard in different positions and with the help of OpenCV's library, the function "findChessboardCorners()" was used to detected the 2D coordinates of the checkerboard corners for each camera [76]. Different position between the pictures is important to increase the accuracy of the system and make a better triangulation, see Figure 3.6.

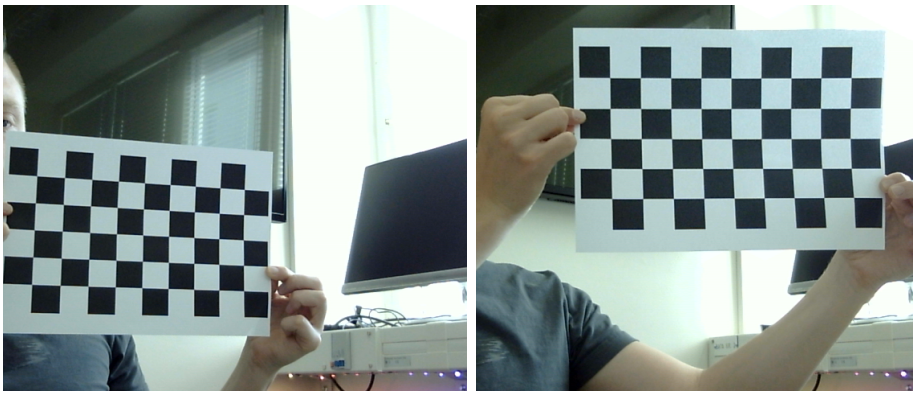


Figure 3.6: Pictures of checkerboard in different positions

Then OpenCV's "calibrateCamera()" function was used [77]. This function takes the 2D coordinates found as input parameters and returns the intrinsic parameters: Camera matrix and the distortion coefficients, and the extrinsic parameters which describes the position and orientation in 3D space with respect to a global coordinate system, these are not used since the extrinsic parameters are estimated with regard to the other camera in the section "Stereo calibration".

The camera matrix contains parameters such as the focal length and principal point of the camera and is unique for each camera. The camera matrix can be estimated since the size of the squares of the checkerboard is known. The principal point is the center of the image in pixel coordinates and was used as a reference point for measuring distances and angles [78]. The focal length of a lens is a measure of the distance between the lens and the image plane [79].

The camera matrix describes the mapping of a camera from 3D points in the world to 2D points in an image [77]. The distortion coefficients of the camera is used to correct for lens distortion.

3.5.3 Stereo Calibration

The same process as described in 3.5.2 was done with multiple cameras at the same time, thus taking pictures on the checkerboard pattern with both cameras simultaneously [75]. Instead of using the "calibratecamera()" function, this time Open CV's "Stereocalibrate()" function was used [80]. This takes in the camera matrices and the distortion coefficients derived from the "calibratecamera()" function as input parameters along with the corner coordinates of the checkerboard detected. Apart from other parameters, this function returns the rotation matrix and the translation vector.

The last step in this calibration process was to create a representation matrix. This matrix was created by concatenating the rotation matrix and the translation vector. The representation matrix was then used to translate a given point from one coordinate system to another. So here it is used to get the relative pose/perspective of the cameras. So, using each cameras representation matrix and each camera detected 2D coordinates of the finger joints, 3D coordinates can be estimated with the help of direct linear transformation (DLT), which is an advanced mathematical technique used for 3D reconstruction [81].

3.5.4 Landmark detection

"Mediapipe hands" is a part of the Mediapipe library and it is used to detect finger joints in a video or image [72]. Here it was used in real-time to update the 2D coordinates of the finger joints for each camera. The finger joints coordinates were stored as 21 points with x- and y coordinates in a list. In Figure 3.7 it is shown how each point was represented in the list.

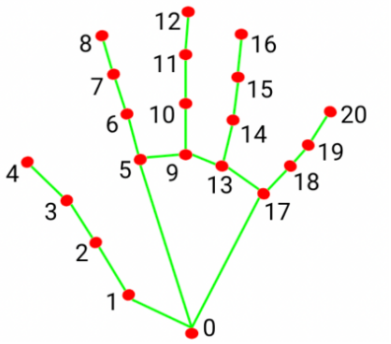


Figure 3.7: Explanation in which order the finger joints are stored [72].

With the help of DLT, the 3D coordinates of the points were attained. The angles were then calculated. This was done by making the 3D coordinates into different vectors and using formula 3.3 multiple times.

$$\cos\theta = \frac{\mathbf{v_1} \cdot \mathbf{v_2}}{\|\mathbf{v_1}\| \, \|\mathbf{v_2}\|} \tag{3.3}$$

The cables on the glove disturbed the landmark detection, therefore another glove was put on the sensor glove to cover up the cables, displayed in Figure 3.8.

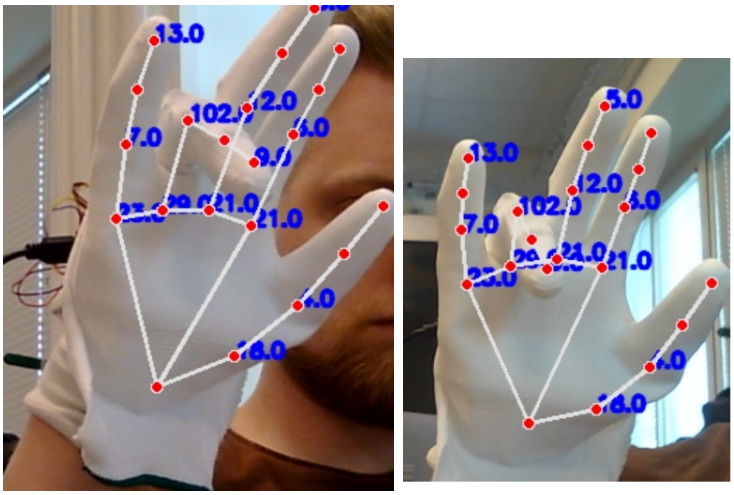


Figure 3.8: Landmark detection with glove on.

3.6 Testing

The testing of the prototype was divided into two parts. The user initially did movements correlating to the letters of the Swedish sign language alphabet, excluding the letters y, å, ä, and ö since they include motions of the wrist not tracked by the glove. The movements were performed according to the alphabet in Figure 3.9.

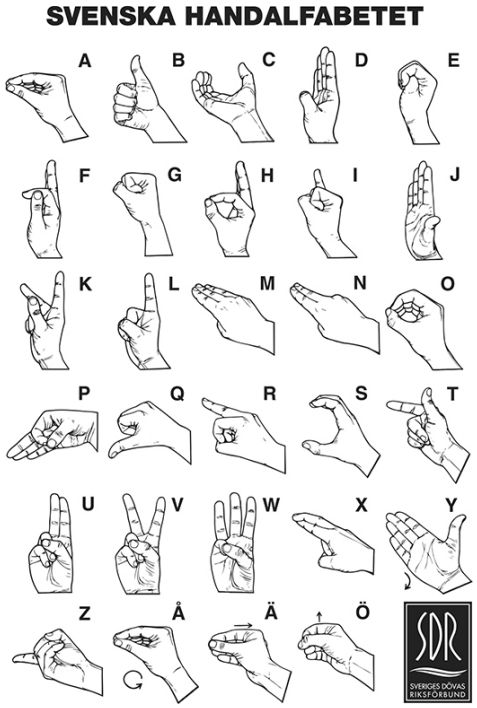


Figure 3.9: The Swedish sign language [82].

The Unity visualization system was used to display the tracked poses. This particular test was performed to compare the prototype to the project from last year, whose validation also incorporated sign language. It is a way to evaluate the performance of the prototype in its entirety.

The second test consisted of measurements made to evaluate the accuracy of the system. The angles for the finger joints were measured and compared to the equivalent values obtained from our camera-validation system. The test subject bent the joints of the fingers in different poses, and the differences in the camera tracking measurements and IMU tracking measurements were noted. An outer glove was used for part of the test to counteract the problem of the cables obstructing the view of the camera. All of the joints were measured twice, once with the outer glove and one without, with the exception of the thumb which was measured twice both with and without the glove. Some measurements of the abduction angles were also taken with the outer glove. For the 90-degree angle, a cardboard piece was used to confirm that both the glove and validation system gave accurate readings. The angles for all fingers except for the thumb and the abduction angles were measured with this cardboard piece.

4
Results

The combination of software and hardware described in the previous chapter is presented here as the definitive result of the project. Assembly of the prototype, its performance and test results will also be covered. The problem specification is used as a guideline for how well the prototype meets expectations.

4.1 Assembly

The final sensor glove prototype used a glove made of nylon fabric. Most components were directly sewn onto the glove using a needle and thread. In total, 10 IMUs, one MUX and one Arduino MC were attached to the glove. The MPUs were sewn on the part of the glove that would represent the proximal and middle phalanges of each finger, which can be viewed in Figure 4.1.

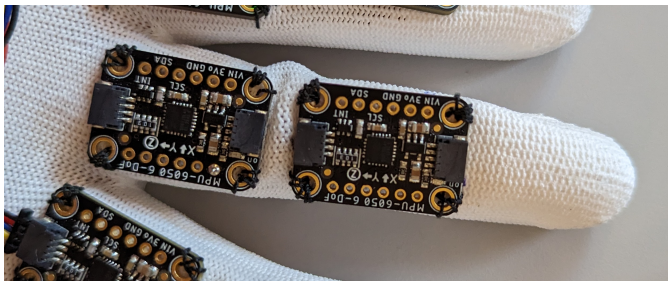


Figure 4.1: Two MPU6050 placed on the middle finger

An exception was made for the thumb since it lacks the middle phalanx, here the IMUs were placed on the proximal- and distal phalanx instead, which is presented in Figure 4.2.

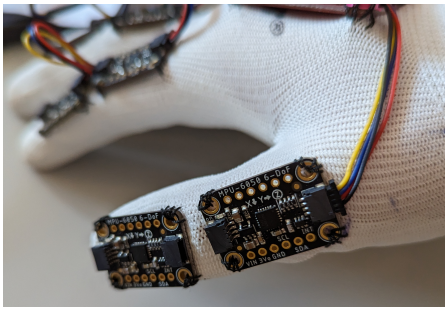
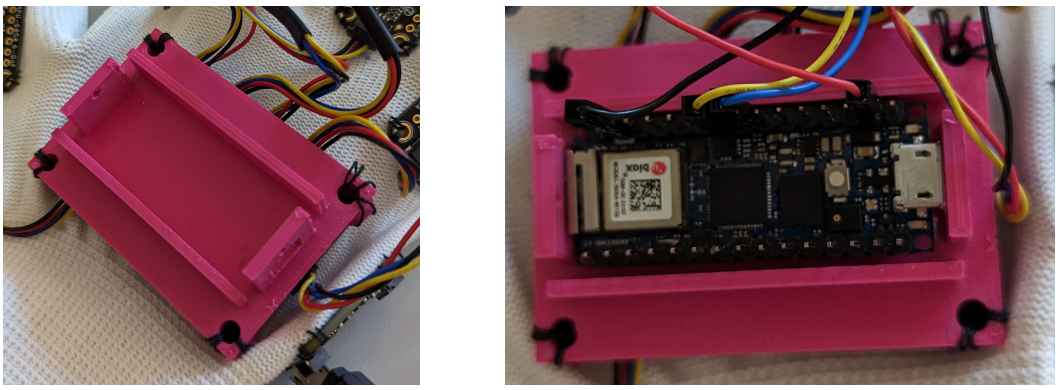


Figure 4.2: View of the IMU placements on the thumb

All of these sensors are connected with Qwiic wires to the MUX that in turn is attached to the back (dorsal part) of the hand. An Arduino MC is placed on a 3D printed part specially designed to be mounted on top of the MUX. This is done in order to optimize usage of available space on the back of the hand and also to make the glove less stiff for the user. Lastly, a micro-USB cable was used to connect the Arduino MC to the computer, and to also keep the MC in place. This can be seen in Figure 4.3.



(a) 3D-printed pocket **(b)** Pocket with MC

Figure 4.3: Plastic pocket above MUX with MC inside

The system as a whole can be viewed in Figure 4.4.

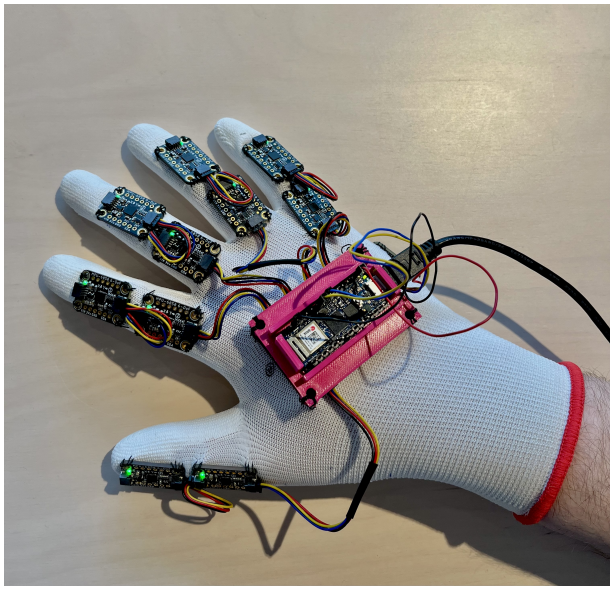


Figure 4.4: The finish sensor glove with all of its components

The choice of sewing the components to the glove does require a little bit of extra force during flexion since the fabric needs to stretch. However, the glove weighs 63g which is less than stated in the requirement list, and all finger movements are possible to execute using the glove.

4.1.1 Visual results

The simulation software described in section 3.4.1 was used in the first test as described in section 3.7. The results can be found in appendix B, in the form of pictures showing the glove and Unity simulation. Out of the 25 letters done with the glove, 13 were done to satisfaction, 10 were considered similar but imperfect and two were completely incorrect. How well the system performed was judged by looking at the hand model from Unity and comparing how well it is replicating the test person's finger positions.

4.2 Accuracy and speed

In the tables below, the measured values from the second test are listed. Accuracy percentages were calculated for the two scenarios, one without (table 4.1) and one with the outer glove (table 4.2). The first nine angles of each table were measured whilst using the 90-degree cardboard piece for all fingers except for the thumb. The remaining angles in the first table are measurements for the thumb. The remaining angles in the second table are angles for the thumb and three abduction angles.

Arduino (without outer glove) [degrees]	Camera validation (without outer glove) [degrees]	Degree difference [degrees]
87.93	Cameras could not detect	-
82.19	94	11.81
79.97	84	4.03
83.64	85	1.36
77.33	Cameras could not detect	-
88.7	100	11.3
85.74	110	24.26
89.97	100	10.03
84.35	95	10.65
60.16	68	7.84
65	40	25
	Average accuracy: 83.60%	Average degree difference: 11.81 degrees

Table 4.1: The result of the accuracy test for the glove, performed without an outer glove.

Arduino (with outer glove) [degrees]	Camera validation (with outer glove) [degrees]	Degree difference [degrees]
81.97	75	6.97
76.65	105	28.35
84.32	103	18.68
73.32	100	26.68
92.72	110	17,28
97.72	110	12.28
96.55	90	6.55
98.98	70	28.98
120	110	10
63.4	103	39.6
41.47	45	3.53
32.03	30	2.03
38.57	31	7.57
30.76	26	4.76
	Average accuracy: 81.32%	Average degree difference: 15.23 degrees

Table 4.2: The result of the accuracy test for the glove, performed with an outer glove.

The differences between the measurements from the Arduino system and the camera validation system are demonstrated in Figure 4.5 and 4.6.

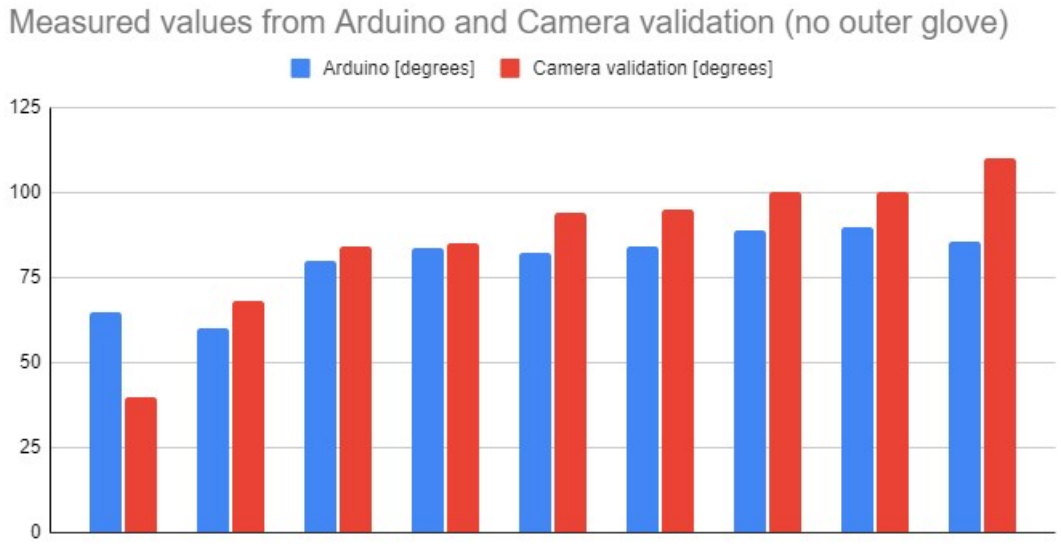


Figure 4.5: Measurements from Arduino and camera validation system, no outer glove

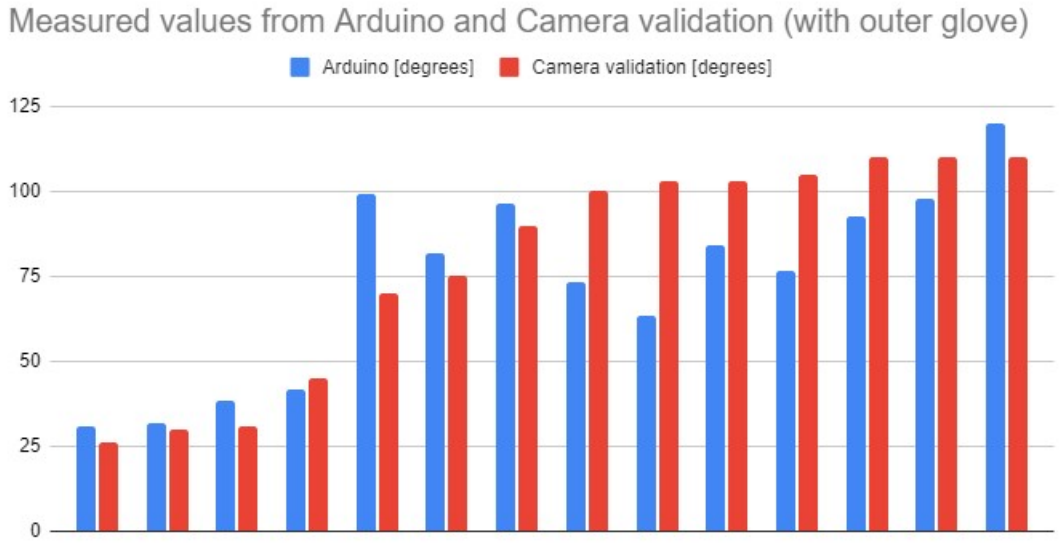


Figure 4.6: Measurements from Arduino and camera validation system, with outer glove

To measure the runtime, the program was executed for 500 iterations and then an average was calculated. Parts of the code that would slow down the process were commented out during these measurements, such as print commands and other sections that are used when testing. Printing the quaternion values is necessary for enabling Unity to retrieve the values and has a big effect on the pace.

Measured unit	Time [ms]
MPU DMP runtime	6.53
LSM6DOX	2.48
System runtime	59.5
System runtime with printing	110.0

Table 4.3: Runtimes, both IMUs and also total system

Worth noting in table 4.3 is that the iteration time is above 100 ms and therefore also above the requirement. Also worth noting is that the MPU DMP is far slower than expected and is slowing down the system significantly. There is however room for improvements that will be discussed in the following chapter.

4.2.1 Validation system

The accuracy of the validation system was difficult to calculate since it is inconsistent. It depends on the cameras, their position, the angle in which the landmarks are detected, the checkerboard and more. The system was tested to see in which angles best results were achieved. These tests were done with cut cardboard figures, where the angles were known, which can be seen in Figure 4.7. The angles tested were 90 degrees and 135 degrees, and the results can be seen in table 4.4.

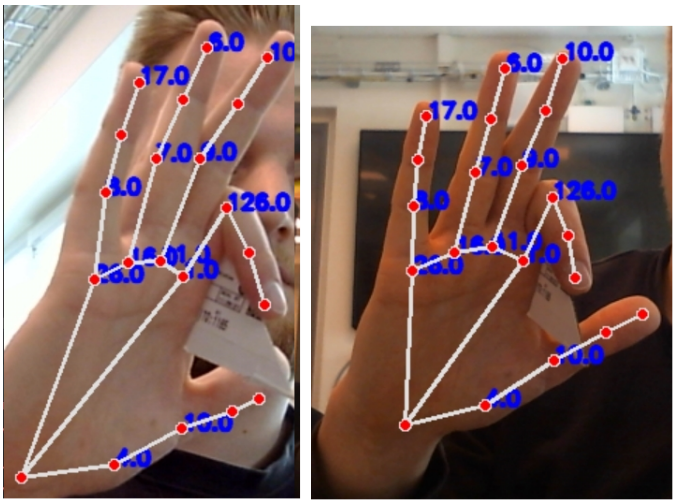


Figure 4.7: Cardboard with known angle to find out most accurate position.

Cardboard Angle [degrees]	Measured angle range [degrees]
130	100-150
90	70-110

Table 4.4: Accuracy test on the validation system

5
Discussion

Many different approaches for creating the prototype have been tried. Some were too time consuming while others were not within the scope of the project. In this chapter, the final prototype and decisive decisions will be reviewed. In addition, the remaining issues, future improvements and ethics of the product will be covered.

5.1 Accuracy

In this section the accuracy of the model will be discussed. The discussion will cover what conclusions can be drawn and what improvements can be done to solve the difficulties that are observed with regard to accuracy.

5.1.1 Testing

The test utilizing Unity to perform Swedish sign language did not achieve the desired results. Optimally, all letters within the range of this project should be very similar in simulation and reality. In practice, only half of the sign letters done achieved this similarity to a satisfactory level. Satisfactory in this case means that the sign could be interpreted correctly.

The project last year managed to do all signs on a satisfactory level which means that their use of sensors might be more suitable. Another conclusion could be that IMUs are quite difficult to work with and that there was missing prerequisite knowledge in this group. Quaternions, sensor fusion filters and photogrammetry have been difficult topics due to lack of knowledge. Much had to be learned during the project and it has taken a lot of time. There are several improvements that could be done to this prototype that potentially has the ability to be a greater solution than last year. Therefore it is not yet possible to conclude that last year's sensors are better suited for this purpose since there is still a lot of room for improvement with this prototype.

There are several reasons why the result was not the same as anticipated. One such reason is that even though the thumb had very good and accurate movement, we couldn't move the entire thumb. Opposition of the thumb was possible to a certain degree, but the thumb could not reach all the way to the ring- and little finger. Another IMU would need to be used to capture the full movement of the CM joint, see figure 2.2 for visualization. More about this issue can be read in section 5.2.

Another issue encountered was that the index finger would occasionally drift when

moving the other fingers. The exact reason for this is unknown, but it was speculated that it could be because one of the IMUs needed to be replaced.

It is worth noting that several sign letters are the same, but with a different position of the wrist. These letters were included in the test, but it also meant that we had leniency when interpreting them. For example, if a 'P' was interpreted as an 'H', then that would still be acceptable. The other letters that are similar in the first test is 'Q' and 'B', 'M' and 'D', and 'L' and 'Z'. Lastly regarding the first test, the sign for 'M' was slightly incorrect. The small finger is supposed to be pointing at a 90-degree angle, but we performed the sign with a straight little finger. However, we think that the error is trivial for the result of the test.

The achieved accuracy for the glove was a bit over 80% when tested with the camera validation system. The angle difference between the glove and the validation system was on average about 12 degrees for the test without the outer glove and 15 degrees with the outer glove. This result is unsatisfactory since a requirement of the system was to have a mean accuracy of less than five degrees. For the angles measured with the 90-degree cardboard piece, the Arduino (IMU) angles were closer to 90 degrees, indicating that it might have been more accurate than the validation system. The average deviation from 90 degrees for the angles measured with the cardboard piece is around 8 degrees for the Arduino values and 11 for the camera values, calculated from the values in table 4.1 and 4.2.

5.1.2 Hand modeling

A big challenge has been to model the hand to be able to constrain unnatural movements. An unnatural movement is for example a finger rotating around its own axis. This has been a difficulty since quaternions are not easy to comprehend. Effort was put into figuring this out but in the end it was not successful. A possibility would be to do these constraints using Euler angles, but they are not robust since they risk of going into Gimbal lock and they are also not continuous since they can jump from -180° to 180°. Another reason for wanting to keep things in quaternions is that Unity has great compatibility with them. A way to improve this project would be to implement constraints that take away movement that is not anatomically possible. These unnatural movements do occur sometimes when visualizing the hand and constraining it would be a priority if this project would continue further. This could also be a possible way to increase accuracy.

The placements of the IMUs affect the sensor readings as well. Since the placements never perfectly align with what is intended, they will have an effect on the measured values as described in section 3.3.2. A slightly misplaced or crooked IMU will produce less accurate results. This problem might become more prevalent when the fingers are bent. This was partly accounted for by correcting for the rotational errors during calibration, but this did not seem to have a significant effect on the overall accuracy of the system. The reasoning for this might be that the IMUs become significantly more misaligned when the fingers are bent and the correction was performed with straight fingers.

5.1.3 IMU selection

During this project, IMUs with 6 degrees of freedom have been used. These ones came with some challenges regarding the starting position. If a 9 DoF IMU was used instead, it would have been possible to find out the starting position of the yaw angle. This could also lead to some angles being more accurate since without a magnetometer there is no reference when rotating an IMU around the direction of gravity. A 6 DoF IMU relies purely on the gyroscope in this case which is not optimal since this will cause drift issues. A problem with a 9 DoF IMU could be that the magnetometers could disturb each other since electrical components produce a magnetic field. There are other solutions that have overcome this problem so as a consequence it is not impossible to do it [36]. A 9 DoF IMU also tends to be a more expensive option. Therefore, disregarding the question of cost, a future improvement could be changing the type of IMU from a 6 DoF to a 9 DoF. Another improvement could be to go for more high-end sensors. Although, this should not be the first thing to try and change since there have been papers receiving great angle measurements using MPU6050 [83]. Still, this is potentially a change that could improve accuracy.

It was not optimal to use two different types of IMUs. The reference IMU is of type LSM6DSOX and the libraries differed and this complicated things. Two examples of differences are that the readable raw data came in different units and IMU pins were used for different causes. This made the software development go more slowly and this would have been easy to solve by simply having only one type of IMU. This should not affect the accuracy but would have made the project easier.

5.2 Opposition of the thumb

The prototype only used two IMUs on the thumb. This is mainly because it is only possible to connect two IMUs in series at a time, without using a external microcontroller. It could have been connected to the MUX but this was considered too late during the project. The lack of a third IMU between the MP and CM joint causes some restrictions. It was not possible to measure the complete opposition of the thumb without this last IMU. A future improvement would therefore be to implement one more Arduino Nano to the system, which is something that was tried by using a master-slave communication between the two Arduino boards. This was not successful due to lack of time, which makes it a good possible improvement for the future. When the thumb has been tracked alone it followed the movements that could be tracked well. This also makes it possible to think that a sensor glove made with only IMUs, is capable of detecting the complex movement of the thumb.

5.3 System speed

To improve the prototype, experimenting with other filters could be a reasonable thing to study further. Especially since the system's frequency is quite low, which can be seen in table 4.3. Furthermore, the runtime of one iteration with the MPU6050 DMP is 2.6 times greater compared to the Madgwick filter implemented

in the LSM6DOX. This was realized too late during the project since most experimentation was done with only a few IMUs and when only a few IMUs were used it was still possible to maintain a high frequency. This is the reason why it was not examined exactly how fast the DMP filter was, which is unfortunate since it in the final prototype lowers the speed of the system.

One consideration made was that maybe the filter was not the problem and instead the data transfer was the cause for the MPU6050 being slower. To test this a program was created where the raw data from the MPU6050 was read and filtered with the same Madgwick filter that had been utilized on the LSM6DOX IMU. It turns out that the time between iterations was 2.72 ms which is 2.4 times faster than the DMP filter. With this in mind, it can be concluded that the filter was the main reason for halting the system and not the data transfer time. If this project is built upon in the future switching away from the DMP filter to another faster sensor fusion filter should be a high priority. The first filter to be tried is the Madgwick filter because it has already been implemented and would probably not be too difficult to implement on the fingers as well. To further improve this system it would be reasonable to try with other sensor fusion filters, especially the Mahony filter since it requires less computational power compared to the Madgwick filter.

A lower computational load is important because it allows the sensors to take in more samples. This in turn causes the system to be more smooth and notice movements that could have happened in between movements if the frequency was lower. Therefore a good way to improve performance would be to make the system faster and switching filters could be a good way to achieve this. At least switching away from the DMP filter could lower this time significantly since the filtering is taking much time in the iterations.

Another time-consuming thing was printing all quaternion values which can be seen in table 4.3. There might be faster ways of doing this. Currently the library "Serial Port" is utilized in Unity and it might be good to investigate if there is faster ways of transferring data between the MC and Unity. Another simple way to increase the system speed would be to buy a more powerful MC. It is also possible to explore different communication protocols than I2C to see if they are faster.

In the final prototype, the update frequency is suboptimal which causes problems in achieving a good visual speed and also causes the system to lose precision during movements. Improving speed should be a high priority if this project would continue further. Changing the DMP filter to a Madgwick or Mahony is probably the first change that should be implemented but there are more things that can be tested and implemented.

The Kalman filter or the extended variant would be interesting to investigate since they have been proven to outperform Madgwick in terms of precision. However, the system currently observes problems with update frequency and thus it would probably require hardware upgrades for a KF or EKF to be a good fit.

5.4 Flexibility of the glove

The glove in its current state suffers from the same issue as last year's Bachelor's thesis that also explored hand motion sensors. The glove only fits certain hand

sizes. This means that the glove would need to be produced in several sizes in order to be available to everyone, which increases production costs. The increased costs can in turn make the product a less viable choice for occupational therapists. An improvement could therefore be to make the placement of IMUs more modular. For example, using a sort of exoskeleton on the hand instead of the glove can make it possible to move or detach parts to make it better fit the user's hand. Though this would most likely increase the cost, it would also make it so that one size fits all. Consecutively, this makes the product more convenient to use for all kinds of users and might make it cheaper in total since it would eliminate the need for several sizes.

In its current state, the prototype is somewhat rigid due to most components being directly sewn onto the glove. This leads to slight limitations in possible hand movement and makes the glove less comfortable for the wearer. For example, during the process of developing and testing the glove, one major challenge encountered concerned the cables. When bending the fingers, the cables were stretched and consequently broke. This probably occurred because of the close spacing between the IMUs on the finger, which caused the Qwiic connectors to break. An improvement would be to create a more flexible and modifiable system. This has been done before by using flexible composite boards (FFCB) [36]. This makes it possible to solder the IMU directly to the board, which makes it take up less space, and will therefore make the glove more flexible.

An upside of this prototype is that the glove with all the components does not weigh much. More generally it is possible to state that IMU-based gloves do not have heavy components on them. Therefore it is possible to construct a glove that is light and flexible with IMUs which is a big upside. In order to minimize obstruction the changes discussed in the paragraph above should be implemented.

5.5 Validation system

The validation system was very hard to validate which was a problem since the purpose of the validation system was of course to be very accurate. The accuracy mentioned in section 3.5 is the accuracy of mediapipe detecting landmarks with only one camera. When multiple cameras are used there are a lot of different error sources so it is difficult to estimate the error.

We could though see that the system was a little bit inaccurate since the same hand pose did give a slightly different output of some angles in different positions. As seen in the results, the same angle could very with about $\pm$ 20 degrees. If the system was accurate the angles should not have changed.

Two things that probably could have improved the accuracy of the system would first be to have a third camera. The more cameras that help with the locating of the finger joints the better the accuracy. The second thing would be to have a better stand for the cameras, this in combination with a smaller checkerboard would probably make the system better at detecting the depth and the checkerboard could have a bigger variety of position in the calibration process.

Even if the system would be made more accurate there was still some issues found about the system. The finger joints detected with one camera could be in perspective

of the palm while for the other camera it could be the joints in perspective of the back of the hand. This causes inaccuracy when calculating the angles. Also, the angle calculation for the validation system was not exactly the same as the angle calculations of the sensor glove. The sensor glove checked how much a sensor tilted while the validation system created vectors based on landmark detection. The targeted angle is quite similar but not exactly the same.

Lastly, the validation system is best at detecting hands and not gloves. When using the glove the system had a harder time locating the finger joints and the position of the joints could differ a little bit. One problem was the cables on the back of the hand of the sensor glove. These cables was in the way of the finger joint detection by the cameras. Some angles could not be detected because the system was confused by the cables. Therefore a second glove was put on the sensor glove to cover up the cables, using this method all angles could be estimated by the system but the method is believed to be slightly more inaccurate since the cables inside the glove caused the glove to be thicker and misshaped and finger joints might not have been as accurately placed by the system. Both methods were used in the accuracy test.

A system where markers would be used to track the angles would perhaps be better, because markers could be places and more exactly track the angle we wanted. Another solution would be to find and use an already good system for validating the sensor glove, since the validation system is only for validating and rarely used.

5.6 Accessible software

During this project much of the software is not intuitive and easy to use. The reason behind this is that a lot of testing has been done and the primary focus has been to make sure that the glove works and to improve its performance. Especially since it has not been satisfactory. The software is part of the glove and a possible improvement is to make the software more user-friendly.

An issue when it comes to rehabilitation is that it can be repetitive and mundane. Serious games, also known as applied games, are games that are designed to counteract this by making the rehabilitation process more fun and engaging. This is achieved by using the system given to the patient and making the patient's movements correspond to different actions in a computer game. The hand motion sensor prototype can be used to, in addition to collecting data, play games that encourage the patient to continue with their exercises. The game would need to have a large set of options that allow the therapist to adapt the game after each patient's need but it could prove to be a complementary tool to the prototype.

5.7 Ethics

It is essential to consider the ethical aspect during the development and production of medical equipment since the healthcare system deals with delicate issues concerning human well-being. The sensor glove will gather information that is to be utilized for rehabilitation purposes, this may raise ethical concerns about privacy, safety, and

accessibility.

Given that the glove is used to gather information about a person's health, privacy may become an issue. In the hands of employers, this information could be used to discriminate against employees or applicants [84].The data could also be used by health insurance companies to predict how expensive their customers will be, and in some cases, they might wrongfully raise insurance rates for people they deem expensive [85]. To prevent this, and prevent other harmful uses of the data, secure data storage is of great importance.

Medical equipment must be safe and of high quality [86]. Since the glove will be used by injured patients, it is critical that it does not strain the hand enough to cause further damage. The glove should collect accurate data in order for proper procedures to be followed, incorrect procedures may result in additional harm.

If the glove is not made adjustable enough, its usage might be limited to patients with hands that fit the hand that the glove is modeled for. This issue was discussed by the group from last year whose work this project is based upon. In a commercialized version of the final product, it should be able to fit the vast majority of hands. Another option is to produce the gloves in various sizes, this would drive up the costs for the people intending to use the product on multiple patients.

Another issue that was discussed by last year's group is the issue that cost has on product accessibility. If the price of the product is too high, healthcare providers with access to lower amounts of money become less likely to buy and utilize the product. This may widen the gap in health care quality between rich and poor regions.

6
Conclusion

This project concludes that IMUs are a viable option for detecting finger movements. However, the overall accuracy of this prototype was not improved compared to last year's project when looking at the results of performing sign language. One reason behind this could be the use of 6 DoF IMUs. Using 9 DoF IMUs with the addition of a magnetometer could potentially improve the accuracy. This is because of the possibility to find the starting position of the yaw angle and there is a reference when rotating around the direction of gravity. Last years project had an issue with the opposition of the thumb, which still remains with our system, but should be easy to fix with the addition of an extra IMU placed on the outside of the CM joint.

Regarding the performance of the glove, the Madgwick filter was found adequate for the purpose and with the limited processing power available provided reasonable filtering speed. The Mahony filter should however not be overlooked and could be a possible improvement. One strong suggestion is to not use the filter implemented in the DMP which exists on the MPU6050 since it delayed the system significantly.

A validation system to test the accuracy was implemented but should be reviewed and either improved upon or changed altogether in the future. MediaPipe had some issues with tracking the joints of the digits and calculate angles when the glove was worn which affected the accuracy score. One improvement could be to use markers placed on the exact points where tracking should occur. A validation system that is able to validate the angles of the digits in real time should however be a priority.

The placement of the sensors was satisfactory and the user was able to move the hand executing all normal angles normally done with the digits. However, there were some constraints in movement mostly due to the size of the sensors and the placement, which both should be reviewed in the future.

Bibliography

[1] C. B. Wilson, "Sensors in medicine," *BMJ : British Medical Journal*, vol. 319, no. 7220, p. 1288, Nov. 1999, ISSN: 0959-8138. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1129066/ (visited on 01/30/2023).

[2] *Sensors for Health Monitoring*, en. [Online]. Available: https://encyclopedia.pub/entry/1482 (visited on 01/30/2023).

[3] F. Porciuncula *et al.*, "Wearable Movement Sensors for Rehabilitation: A Focused Review of Technological and Clinical Advances," *PM & R : the journal of injury, function, and rehabilitation*, vol. 10, no. 9 Suppl 2, S220–S232, Sep. 2018, ISSN: 1934-1482. DOI: 10.1016/j.pmrj.2018.06.013. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6700726/ (visited on 01/30/2023).

[4] *Rehabilitation*, en. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/rehabilitation (visited on 01/30/2023).

[5] G. De Pasquale and L. Mastrototaro, "Glove-based systems for medical applications: Review of recent advancements," English, *Journal of Textile Engineering & Fashion Technology*, vol. Volume 4, no. Issue 3, Jun. 2018, ISSN: 2574-8114. DOI: 10.15406/jteft.2018.04.00153. [Online]. Available: https://medcraveonline.com/JTEFT/JTEFT-04-00153.pdf (visited on 01/30/2023).

[6] *Handkirurgi | Institutionen för kliniska vetenskaper, Göteborgs universitet*, sv. [Online]. Available: https://www.gu.se/kliniska-vetenskaper/om-oss/amnesomraden/handkirurgi (visited on 01/30/2023).

[7] *History of Hand Surgery*, en. [Online]. Available: https://medicine.yale.edu/surgery/plastics/programs/hand/history/ (visited on 01/30/2023).

[8] R. Kabir, M. S. H. Sunny, H. U. Ahmed, and M. H. Rahman, "Hand Rehabilitation Devices: A Comprehensive Systematic Review," *Micromachines*, vol. 13, no. 7, p. 1033, Jun. 2022, ISSN: 2072-666X. DOI: 10.3390/mi13071033. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9325203/ (visited on 01/30/2023).

[9] A. AXÉN KRÜGER *et al.*, "Wearable sensor based system for measuring hand motions," English, DEPARTMENT OF ELECTRICAL ENGINEERING, Bachelor's thesis, 2022, p. 74. [Online]. Available: https://drive.google.com/file/d/1hWnpf0wDGVZUAzTsQjk-fDeMwfi0vg8T/view (visited on 02/02/2023).

[10] P. J. Bazira, "Surgical anatomy of the hand," *Surgery (Oxford)*, vol. 40, no. 3, pp. 155–162, 2022, ISSN: 0263-9319. DOI: https://doi.org/10.1016/j.mpsur.2022.01.001. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0263931922000011.

[11] A. Zapatero-Gutiérrez, E. Castillo Castaneda, and m. a. Laribi, "On the optimal synthesis of a finger rehabilitation slider-crank-based device with a prescribed real trajectory: Motion specifications and design process," *Applied Sciences*, vol. 11, p. 708, Jan. 2021. DOI: 10.3390/app11020708.

[12] B. Hirt, H. Seyhan, and M. Wagner, *Hand and Wrist Anatomy and Biomechanics : A Comprehensive Guide*, 1st ed. Thieme Medical Publishers, Incorporated, 2016, ISBN: 9783132053519.

[13] *Movements of the fingers*, en. [Online]. Available: https://www.researchgate.net/figure/Various-movements-of-the-fingers-A-The-radial-abduction-adduction-of-the-thumb-and_fig5_348437896 (visited on 01/30/2023).

[14] B.-S. Lin, I.-J. Lee, P.-Y. Chiang, S.-Y. Huang, and C.-W. Peng, "A modular data glove system for finger and hand motion capture based on inertial sensors," *Journal of Medical and Biological Engineering*, vol. 39, no. 4, pp. 532–540, Aug. 1, 2019, ISSN: 2199-4757. DOI: 10.1007/s40846-018-0434-6. [Online]. Available: https://doi.org/10.1007/s40846-018-0434-6 (visited on 01/30/2023).

[15] S. Shin, H. U. Yoon, and B. Yoo, "Hand gesture recognition using EGaIn-silicone soft sensors," *Sensors*, vol. 21, no. 9, p. 3204, Jan. 2021, ISSN: 1424-8220. DOI: 10.3390/s21093204. [Online]. Available: https://www.mdpi.com/1424-8220/21/9/3204 (visited on 01/30/2023).

[16] M. Oudah, A. Al-Naji, and J. Chahl, "Hand gesture recognition based on computer vision: A review of techniques," *Journal of Imaging*, vol. 6, no. 8, p. 73, Aug. 2020, ISSN: 2313-433X. DOI: 10.3390/jimaging6080073. [Online]. Available: https://www.mdpi.com/2313-433X/6/8/73 (visited on 02/03/2023).

[17] A. Axén Krüger *et al.*, "Sensor system for modeling of hand movements," Planning report Bachelor thesis, Chalmers University of Technology, Gothenburg, Jan. 26, 2022, 16 pp.

[18] M. Zia ur Rehman *et al.*, "Multiday EMG-based classification of hand motions with deep learning techniques," *Sensors*, vol. 18, no. 8, p. 2497, Aug. 2018, ISSN: 1424-8220. DOI: 10.3390/s18082497. [Online]. Available: https://www.mdpi.com/1424-8220/18/8/2497 (visited on 02/03/2023).

[19] J. J. LaViola Jr, "A survey of hand posture and gesture recognition techniques and technology," 1999.

[20] G. Buckingham, "Hand tracking for immersive virtual reality: Opportunities and challenges," *Frontiers in Virtual Reality*, vol. 2, 2021, ISSN: 2673-4192. [Online]. Available: https://www.frontiersin.org/articles/10.3389/frvir.2021.728461 (visited on 02/03/2023).

[21] *On-Device, Real-Time Hand Tracking with MediaPipe*, en, Aug. 2019. [Online]. Available: https://ai.googleblog.com/2019/08/on-device-real-time-hand-tracking-with.html (visited on 02/26/2023).

[22] Z. Zhang, K. Yang, J. Qian, and L. Zhang, "Real-time surface EMG pattern recognition for hand gestures based on an artificial neural network," *Sensors (Basel, Switzerland)*, vol. 19, no. 14, p. 3170, Jul. 18, 2019, ISSN: 1424-8220.

DOI: 10.3390/s19143170. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6679304/ (visited on 02/03/2023).

[23] A. G. Jaramillo and M. E. Benalcázar, "Real-time hand gesture recognition with EMG using machine learning," in *2017 IEEE Second Ecuador Technical Chapters Meeting (ETCM)*, Oct. 2017, pp. 1–5. DOI: 10.1109/ETCM.2017.8247487.

[24] J. McIntosh, A. Marzo, M. Fraser, and C. Phillips, "EchoFlex: Hand gesture recognition using ultrasound imaging," in *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, ser. CHI '17, New York, NY, USA: Association for Computing Machinery, May 2, 2017, pp. 1923–1934, ISBN: 9781450346559. DOI: 10.1145/3025453.3025807. [Online]. Available: https://doi.org/10.1145/3025453.3025807 (visited on 02/03/2023).

[25] X. Yang, J. Yan, Y. Fang, D. Zhou, and H. Liu, "Simultaneous prediction of wrist/hand motion via wearable ultrasound sensing," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 28, no. 4, pp. 970–977, Apr. 2020, ISSN: 1558-0210. DOI: 10.1109/TNSRE.2020.2977908.

[26] J. Y. Guo, Y. Zheng, Q. H. Huang, X. Chen, and J. F. He Chen, "Comparison of sonomyography and electromyography of forearm muscles in the guided wrist extension: 5th international workshop on wearable and implantable body sensor networks, BSN 2008, in conjunction with the 5th international summer school and symposium on medical devices and biosensors, ISSS-MDBS 2008," *Proc. 5th Int. Workshop on Wearable and Implantable Body Sensor Networks, BSN 2008, in conjunction with the 5th Int. Summer School and Symp. on Medical Devices and Biosensors, ISSS-MDBS 2008*, pp. 235–238, Sep. 23, 2008, ISSN: 9781424422531. DOI: 10.1109/ISSMDBS.2008.4575062. [Online]. Available: http://www.scopus.com/inward/record.url?scp=51949108783&partnerID=8YFLogxK (visited on 02/03/2023).

[27] H.-H. Chang *et al.*, "A clinical observational study of effectiveness of a solid coupling medium in extracorporeal shock wave lithotripsy," *BMC Urology*, vol. 22, p. 56, Apr. 12, 2022, ISSN: 1471-2490. DOI: 10.1186/s12894-022-01001-y. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9006431/ (visited on 02/03/2023).

[28] J. N. Wilkinson and L. M. Saxhaug, "Handheld ultrasound in training – the future is getting smaller!" *Journal of the Intensive Care Society*, vol. 22, no. 3, pp. 220–229, Aug. 2021, ISSN: 1751-1437. DOI: 10.1177/1751143720914216. [Online]. Available: http://journals.sagepub.com/doi/10.1177/1751143720914216 (visited on 02/03/2023).

[29] A. Bhuyan *et al.*, "Integrated circuits for volumetric ultrasound imaging with 2-d CMUT arrays," *IEEE transactions on biomedical circuits and systems*, vol. 7, no. 6, pp. 796–804, Dec. 2013, ISSN: 1940-9990. DOI: 10.1109/TBCAS.2014.2298197.

[30] T. Kuroiwa *et al.*, "Device development for detecting thumb opposition impairment using carbon nanotube-based strain sensors," *Sensors (Basel, Switzerland)*, vol. 20, no. 14, p. 3998, Jul. 18, 2020, ISSN: 1424-8220. DOI: 10.3390/s20143998. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7412202/ (visited on 01/30/2023).

[31] "IMU-sensorer – därför behövs de i automations- och automotive applikationer | Cumatix." (), [Online]. Available: https://cumatix.se/imu-sensorer-darfor-behovs-de-i-automations-och-automotive-applikationer/ (visited on 01/30/2023).

[32] "Accelerometer vs gyroscope sensor, and IMU, how to pick one?" Latest Open Tech From Seeed. (Dec. 24, 2019), [Online]. Available: https://www.seeedstudio.com/blog/2019/12/24/what-is-accelerometer-gyroscope-and-how-to-pick-one/ (visited on 01/30/2023).

[33] "IMU - inertial measurement unit," SBG Systems. (), [Online]. Available: https://www.sbg-systems.com/inertial-measurement-unit-imu-sensor/ (visited on 01/30/2023).

[34] E. Nelson Henderson, *AN INERTIAL MEASUREMENT SYSTEM FOR HAND AND FINGER TRACKING*, Dec. 2011. [Online]. Available: https://scholarworks.boisestate.edu/cgi/viewcontent.cgi?referer=&httpsredir=1&article=1233&context=td.

[35] B. O. PhD. "What is IMU?" Medium. (Nov. 19, 2022), [Online]. Available: https://towardsdatascience.com/what-is-imu-9565e55b44c (visited on 04/28/2023).

[36] B.-S. Lin, I.-J. Lee, S.-Y. Yang, Y.-C. Lo, J. Lee, and J.-L. Chen, "Design of an Inertial-Sensor-Based Data Glove for Hand Function Evaluation," *Sensors (Basel, Switzerland)*, vol. 18, no. 5, p. 1545, May 2018, ISSN: 1424-8220. DOI: 10.3390/s18051545. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5982580/ (visited on 04/11/2023).

[37] "What is an IMU sensor and how to use with arduino?" Latest Open Tech From Seeed. (Jan. 17, 2020), [Online]. Available: https://www.seeedstudio.com/blog/2020/01/17/what-is-imu-sensor-overview-with-arduino-usage-guide/ (visited on 01/30/2023).

[38] J. Gullberg Carlsson and J. Åkerblom Svensson, *Analysis of comparative filter algorithm effect on an IMU*, May 2021.

[39] F. Fei *et al.*, "Development of a wearable glove system with multiple sensors for hand kinematics assessment," *Micromachines*, vol. 12, no. 4, p. 362, Mar. 27, 2021, ISSN: 2072-666X. DOI: 10.3390/mi12040362. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8066750/ (visited on 02/22/2023).

[40] Strickland. "What is a gimbal – and what does it have to do with NASA?" HowStuffWorks. (May 20, 2008), [Online]. Available: https://science.howstuffworks.com/gimbal.htm (visited on 02/22/2023).

[41] G. Sanderson, *Quaternions and 3d rotation, explained interactively*, Oct. 26, 2018. [Online]. Available: https://www.youtube.com/watch?v=zjMuIxRvygQ (visited on 02/23/2023).

[42] M. Hu, Q. Zhang, J. Yang, and X. Li, "Unit quaternion description of spatial rotations in 3d electron cryo-microscopy," *Journal of Structural Biology*, vol. 212, no. 3, p. 107 601, Dec. 1, 2020, ISSN: 1047-8477. DOI: 10.1016/j.jsb.2020.107601. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S104784772030174X (visited on 02/24/2023).

[43] R. Bhadani, *Quaternion and euler angles*, Jun. 9, 2019.

[44] J. Voight, *Quaternion Algebras*, English. Springer Nature, 2021, ISBN: 9783030566944. DOI: 10.1007/978-3-030-56694-4. [Online]. Available: https://library.oapen.org/handle/20.500.12657/50018 (visited on 05/15/2023).

[45] W. Elmenreich, "An introduction to sensor fusion," Apr. 2023.

[46] N. Büscher, D. Gis, V. Kühn, and C. Haubelt, "On the Functional and Extra-Functional Properties of IMU Fusion Algorithms for Body-Worn Smart Sensors," *Sensors (Basel, Switzerland)*, vol. 21, no. 8, p. 2747, Apr. 2021, ISSN: 1424-8220. DOI: 10.3390/s21082747. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8069451/ (visited on 04/11/2023).

[47] P. Gui, L. Tang, and S. Mukhopadhyay, "MEMS based IMU for tilting measurement: Comparison of complementary and kalman filter based data fusion," in *2015 IEEE 10th Conference on Industrial Electronics and Applications (ICIEA)*, Jun. 2015, pp. 2004–2009. DOI: 10.1109/ICIEA.2015.7334442.

[48] *Kalman Filter - an overview | ScienceDirect Topics*. [Online]. Available: https://www.sciencedirect.com/topics/earth-and-planetary-sciences/kalman-filter (visited on 05/15/2023).

[49] Y. Kim and H. Bang, *Introduction to Kalman Filter and Its Applications*, en. IntechOpen, Nov. 2018, ISBN: 9781838805371. DOI: 10.5772/intechopen.80600. [Online]. Available: https://www.intechopen.com/chapters/63164 (visited on 05/15/2023).

[50] C. Yang, J. Zheng, X. Ren, W. Yang, H. Shi, and L. Shi, "Multi-Sensor Kalman Filtering With Intermittent Measurements," *IEEE Transactions on Automatic Control*, vol. 63, no. 3, pp. 797–804, Mar. 2018, ISSN: 1558-2523. DOI: 10.1109/TAC.2017.2734643.

[51] "A Comparative Analysis of Orientation Estimation Filters using MEMS based IMU," in *2nd International Conference on Research in Science, Engineering and Technology (ICRSET'2014), March 21-22, 2014 Dubai (UAE)*, International Institute of Engineers, Mar. 2014, ISBN: 9789382242819. DOI: 10.15242/IIE.E0314552. [Online]. Available: http://iieng.org/siteadmin/upload/7697E0314552.pdf (visited on 05/15/2023).

[52] P. Kumar L.S and S. Mansukhani, "Prediction using Kalman filter," 3400 Dundee Rd, Suite 160, Northbrook, Tech. Rep., Jun. 2011. [Online]. Available: https://www.mu-sigma.com/wp-content/uploads/2019/08/Prediction-using-Kalman-Filter.pdf (visited on 05/15/2023).

[53] M. Linderoth, K. Soltesz, A. Robertsson, and R. Johansson, "Initialization of the Kalman filter without assumptions on the initial state," in *2011 IEEE International Conference on Robotics and Automation*, ISSN: 1050-4729, May 2011, pp. 4992–4997. DOI: 10.1109/ICRA.2011.5979684.

[54] N. Thacker, T. Lacey, and N. Thacker, "Tutorial: The kalman filter," May 2023.

[55] S. A. Ludwig and K. D. Burnham, "Comparison of Euler Estimate using Extended Kalman Filter, Madgwick and Mahony on Quadcopter Flight Data," in *2018 International Conference on Unmanned Aircraft Systems (ICUAS)*, ISSN: 2575-7296, Jun. 2018, pp. 1236–1241. DOI: 10.1109/ICUAS.2018.8453465.

[56] S. Wilson *et al.*, "Formulation of a new gradient descent MARG orientation algorithm: Case study on robot teleoperation," en, *Mechanical Systems and Signal Processing*, vol. 130, pp. 183–200, Sep. 2019, ISSN: 0888-3270. DOI: 10.1016/j.ymssp.2019.04.064. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0888327019303012 (visited on 04/11/2023).

[57] S. O. Madgwick, "An efficient orientation filter for inertial and inertial/magnetic sensor arrays," Apr. 2010. [Online]. Available: https://courses.cs.washington.edu/courses/cse466/14au/labs/l4/madgwick_internal_report.pdf.

[58] C. Yi *et al.*, "Estimating Three-Dimensional Body Orientation Based on an Improved Complementary Filter for Human Motion Tracking," en, *Sensors*, vol. 18, no. 11, p. 3765, Nov. 2018, ISSN: 1424-8220. DOI: 10.3390/s18113765. [Online]. Available: https://www.mdpi.com/1424-8220/18/11/3765 (visited on 04/11/2023).

[59] a. H. J. Fakhri Alam Zhou ZhaiHe, "A Comparative Analysis of Orientation Estimation Filters using MEMS based IMU," in *ICRSET'2014*, Dubai: International Institute of Engineers, Mar. 2014.

[60] I. Prayudi, E.-H. Seo, D. Kim, and B.-J. You, "Implementation of an Inertial Measurement Unit based motion capture system," in *2011 8th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, Nov. 2011, pp. 425–429. DOI: 10.1109/URAI.2011.6145856.

[61] I. Prayudi and D. Kim, "Design and implementation of IMU-based human arm motion capture system," in *2012 IEEE International Conference on Mechatronics and Automation*, ISSN: 2152-744X, Aug. 2012, pp. 670–675. DOI: 10.1109/ICMA.2012.6283221.

[62] B. Siepert, *MPU6050 datasheet*, publisher: Adafruit learning system, Dec. 2019. [Online]. Available: https://www.elfa.se/Web/Downloads/_m/an/3886_eng_man.pdf (visited on 05/05/2023).

[63] I. Susnea and M. Mitescu, *Microcontrollers in Practice. [electronic resource]* (Springer Series in Advanced Microelectronics: 18), 1st ed. 2005. Springer Berlin Heidelberg, 2005, ISBN: 9783540283089. [Online]. Available: https://search.ebscohost.com/login.aspx?direct=true%7B%5C&%7Ddb=cat07472a%7B%5C&%7DAN=clec.SPRINGERLINK9783540283089%7B%5C&%7Dsite=eds-live%7B%5C&%7Dscope=site%7B%5C&%7Dauthtype=guest%7B%5C&%7Dcustid=s3911979%7B%5C&%7Dgroupid=main%7B%5C&%7Dprofile=eds%20https://cthliblicenses.azurewebsites.net/License?id=354%20http://proxy.lib.chalmers.se/login?url=https://doi.org/10.1007/3-540-28308-0.

[64] C. K. Mummadi *et al.*, "Real-Time and Embedded Detection of Hand Gestures with an IMU-Based Glove," en, *Informatics*, vol. 5, no. 2, p. 28, Jun. 2018, ISSN: 2227-9709. DOI: 10.3390/informatics5020028. [Online]. Available: https://www.mdpi.com/2227-9709/5/2/28 (visited on 04/11/2023).

[65] J. N. A. L. Leijnse and C. W. Spoor, "Reverse engineering finger extensor apparatus morphology from measured coupled interphalangeal joint angle trajectories — a generic 2D kinematic model," en, *Journal of Biomechanics*,

vol. 45, no. 3, pp. 569–578, Feb. 2012, ISSN: 0021-9290. DOI: 10.1016/j.jbiomech.2011.11.002. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0021929011006841 (visited on 04/18/2023).

[66] S. O. H. Madgwick, A. J. L. Harrison, and R. Vaidyanathan, "Estimation of IMU and MARG orientation using a gradient descent algorithm," in *2011 IEEE International Conference on Rehabilitation Robotics*, ISSN: 1945-7901, Jun. 2011, pp. 1–7. DOI: 10.1109/ICORR.2011.5975346.

[67] B. Siepert. "MPU6050 6-DoF accelerometer and gyro." (), [Online]. Available: https://www.elfa.se/Web/Downloads/_m/an/3886_eng_man.pdf (visited on 03/30/2023).

[68] L. Ada, "I2C addresses!," p. 18, 2021. [Online]. Available: https://learn.adafruit.com/i2c-addresses.

[69] E. W. Weisstein, *Quaternion*, en, Text. [Online]. Available: https://mathworld.wolfram.com/ (visited on 05/12/2023).

[70] A. Asadzadeh, T. Samad-Soltani, P. Rezaei-Hachesu, and Z. Salahzadeh, "Low-Cost Interactive Device for Virtual Reality," in *2020 6th International Conference on Web Research (ICWR)*, Apr. 2020, pp. 38–42. DOI: 10.1109/ICWR49608.2020.9122307.

[71] *Oculus hand models*, publisher: Meta Platforms Technologies, LLC and its affiliates, Jul. 2017. [Online]. Available: https://developer.oculus.com/downloads/package/oculus-hand-models/.

[72] *Hand landmarks detection guide | MediaPipe*, en. [Online]. Available: https://developers.google.com/mediapipe/solutions/vision/hand_landmarker (visited on 05/02/2023).

[73] *Photogrammetry - an overview | ScienceDirect Topics*. [Online]. Available: https://www.sciencedirect.com/topics/agricultural-and-biological-sciences/photogrammetry (visited on 05/02/2023).

[74] *What is photogrammetry | Professional 3D scanning solutions | Artec 3D*, en-us. [Online]. Available: https://www.artec3d.com/learning-center/what-is-photogrammetry (visited on 05/15/2023).

[75] T. Batpurev, *Real time 3D hand pose estimation using MediaPipe*, 2021. [Online]. Available: https://temugeb.github.io/opencv/python/2021/02/02/stereo-camera-calibration-and-triangulation.html.

[76] *OpenCV: Camera Calibration and 3D Reconstruction*. [Online]. Available: https://docs.opencv.org/4.x/d9/d0c/group__calib3d.html (visited on 05/02/2023).

[77] *OpenCV: Camera Calibration*. [Online]. Available: https://docs.opencv.org/4.x/dc/dbb/tutorial_py_calibration.html (visited on 05/11/2023).

[78] *Things to know about the main points of the lens*, sv-SE. [Online]. Available: https://hackmd.io/@fkYvVlntToCm6v873TKilg/BJ-Svgx8Y (visited on 05/11/2023).

[79] *Focal Length | Understanding Camera Zoom & Lens Focal Length | Nikon | Nikon*, en. [Online]. Available: https://www.nikonusa.com/en/learn-and-explore/a/tips-and-techniques/understanding-focal-length.html (visited on 05/11/2023).

[80] *OpenCV: Camera Calibration and 3D Reconstruction*. [Online]. Available: https://docs.opencv.org/4.x/d9/d0c/group__calib3d.html#ga9d2539c1ebcda6%2047487a616bdf0fc716 (visited on 05/11/2023).

[81] T. Batpurev, *Direct linear transformation*, 2021. [Online]. Available: https://temugeb.github.io/computer_vision/2021/02/06/direct-linear-transorms.html.

[82] "Svenska handalfabetet," Sveriges Dövas Riksförbund. (Apr. 8, 2021), [Online]. Available: https://sdr.org/media/svenska-handalfabetet/ (visited on 05/15/2023).

[83] C. Yang *et al.*, "Physical extraction and feature fusion for multi-mode signals in a measurement system for patients in rehabilitation exoskeleton," *Sensors*, vol. 18, p. 2588, Aug. 2018. DOI: 10.3390/s18082588.

[84] *Employment Discrimination Based on Medical Conditions and Disabilities*, en. [Online]. Available: https://www.disabilitysecrets.com/resources/disability/disability-discrimination/discrimination-workp (visited on 01/30/2023).

[85] M. Allen, "Health Insurers Are Vacuuming Up Details About You — And It Could Raise Your Rates," en, *NPR*, Jul. 2018. [Online]. Available: https://www.npr.org/sections/health-shots/2018/07/17/629441555/health-insurers-are-vacuuming-up-details-about-you-and-it-could-raise-your-rates (visited on 01/30/2023).

[86] *Management and Use*, en. [Online]. Available: https://www.who.int/teams/health-product-policy-and-standards/assistive-and-medical-technology/medical-devices/management-use (visited on 01/30/2023).

A

Appendix 1

The list below list contains specific requirements that were in mind when the product was created and that should be fulfilled. In addition, there are wishes that are marked with priority 1 (highest) to 10 (lowest). The list also includes accompanying validation methods for some of the specifications.

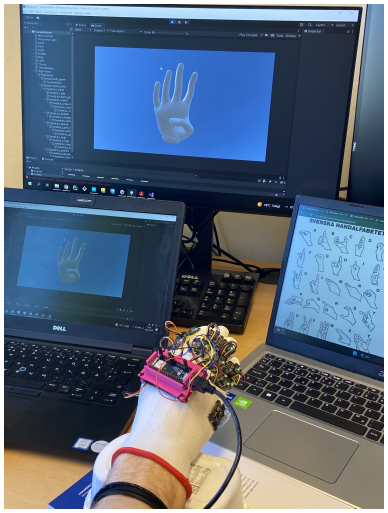
Requirement specification		**Project Name: Robotic Rehabilitation: Hand Motion Sensor**			
Group: EENX16-23-14		Date: 13/2-2023			
	Name	Description	Requirement/Wish	Priority (1-10)	Validation
Data communication					
	Response time is less than 100 ms	Delay from user to digital output	Requirement		Arduino IDE
Electronics					
	Cable management	Have cables not obscure the hand	Requirement		
	Cable management wish	Minimize the number of loose cables and use circuit boards instead	Wish	2	
Material					
	Weight on hand < 100g	The total weight on the users hand should be minimized	Requirement		Scale
Safety					
	Emergency stop	It must be a way to cut the power if an emergency occurs	Requirement		
Software					
	Accurate signal	Noise and artefacts needs to be kept to a minimum with i.e filters.	Requirement		Simulation
Accuracy					
	Mean accuracy < 5°	Angle between real and calculated angles, for 10 samples.	Requirement		Simulation
	Mean accuracy should be minimized	Accuracy is measured in angle between real and calculated angle and should therefore be minimized to improve the hand tracking	Wish	1	
Usage					
	Portable	The system should be easy for a patient to bring home so that it can be used regularly without hospital visits.	Requirement		
	Manipulation of every finger	Each finger should be individually measured and the system should not restrict the movement of fingers	Requirement		
Budget					
	The total cost less than 5000 SEK		Requirement		Supervisor/ Budget

Figure A.1: Requirement Specification for Hand Motion Sensor Glove

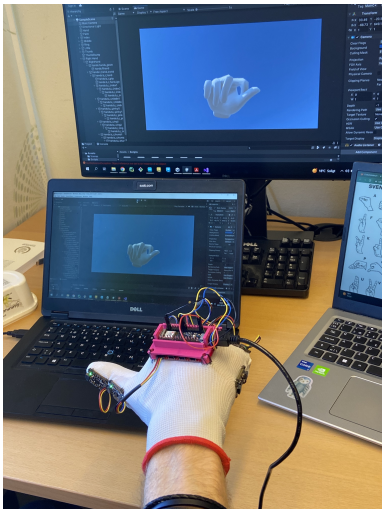
B

Appendix 2

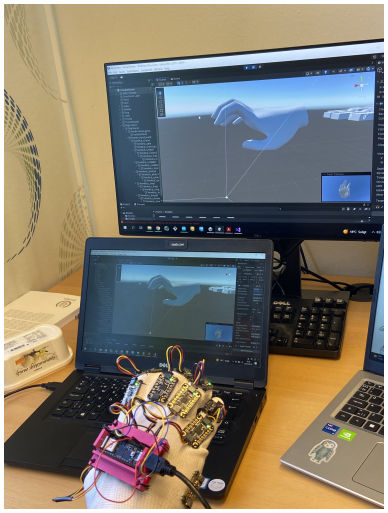
The pictures of the visualisation of the hands gesture in Unity is presented below in figure B.1.



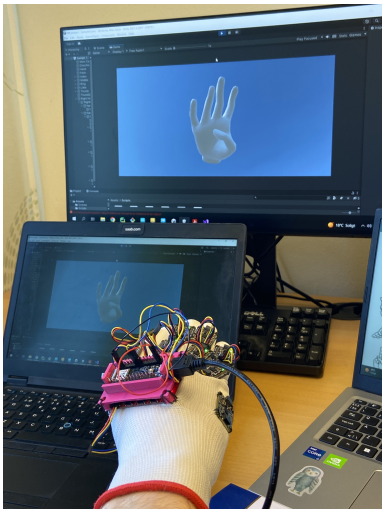
(a) A



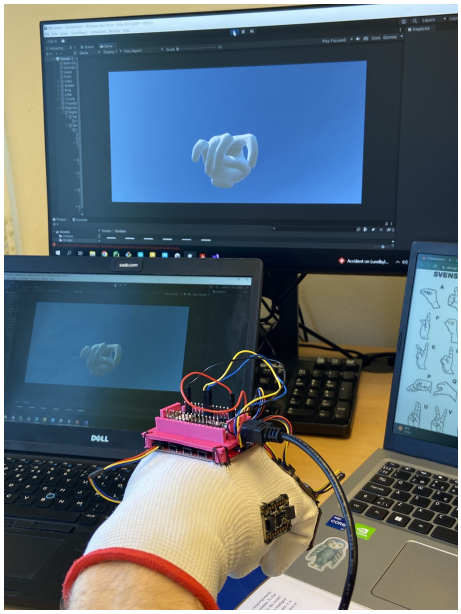
(b) B



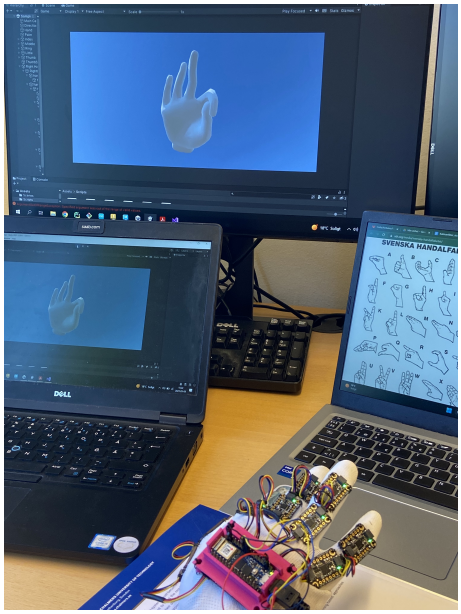
(c) C



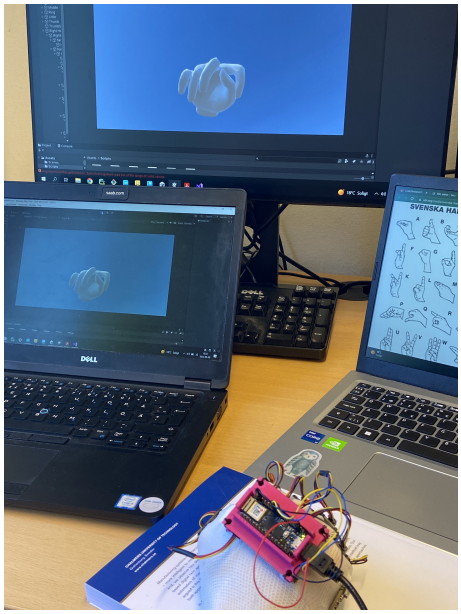
(d) D



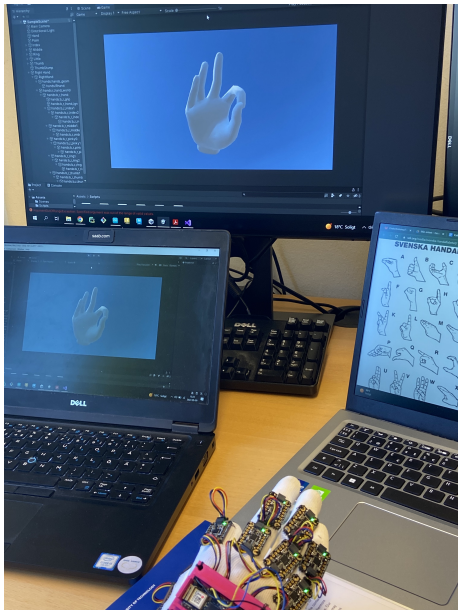
(e) E



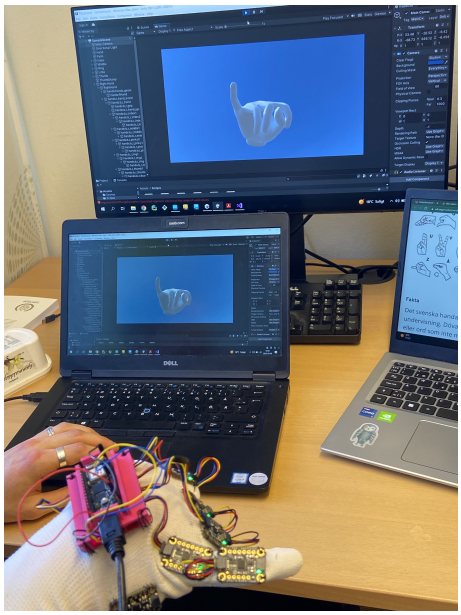
(f) F



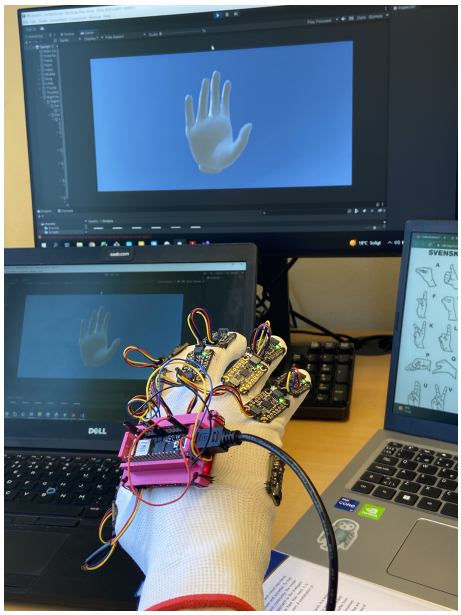
(g) G



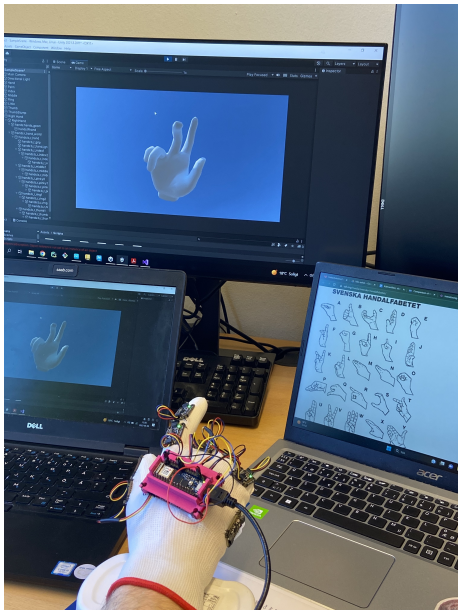
(h) H



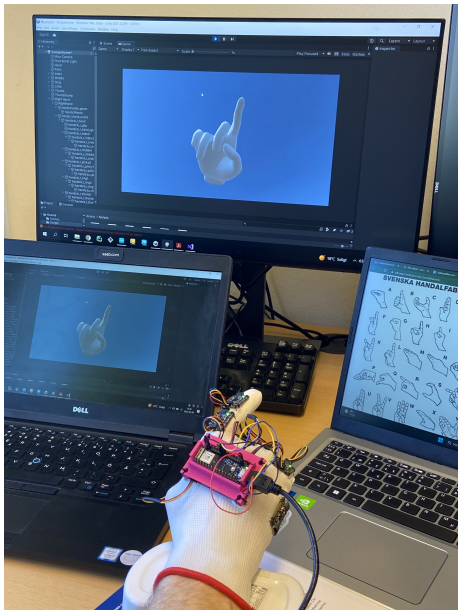
(i) I



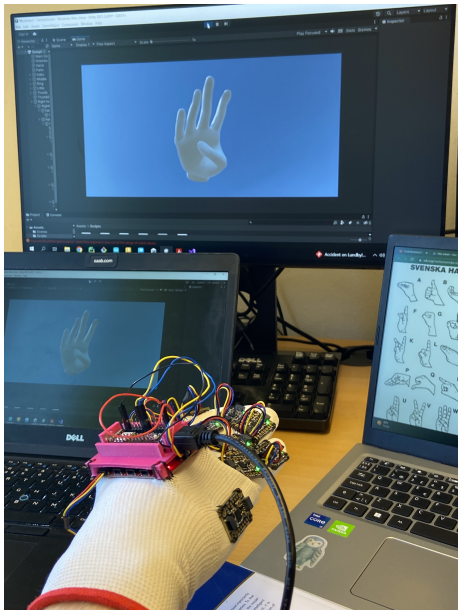
(j) J



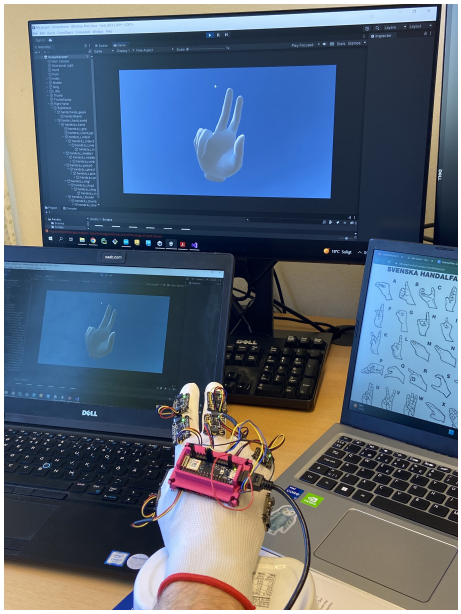
(k) K



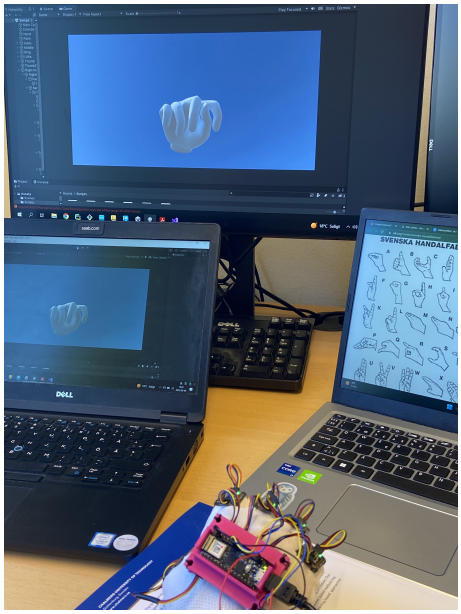
(l) L



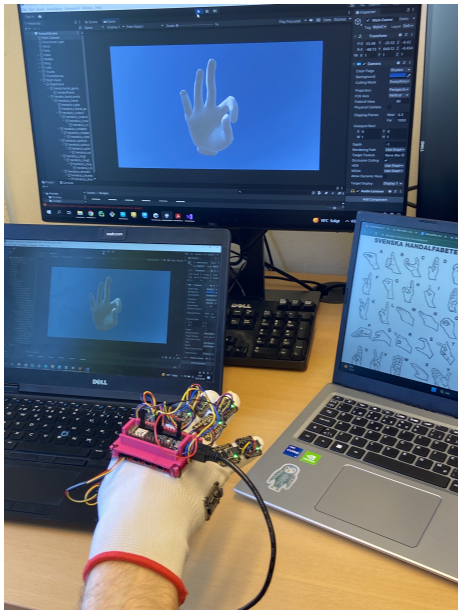
(m) M



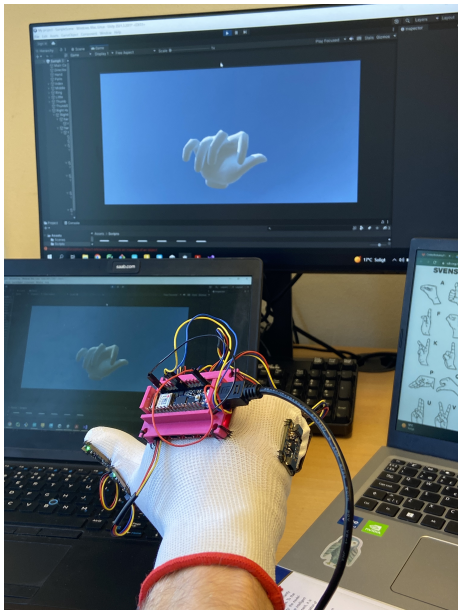
(n) N



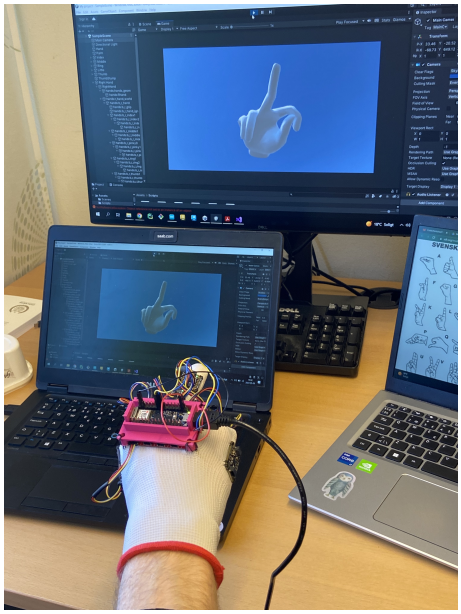
(o) O



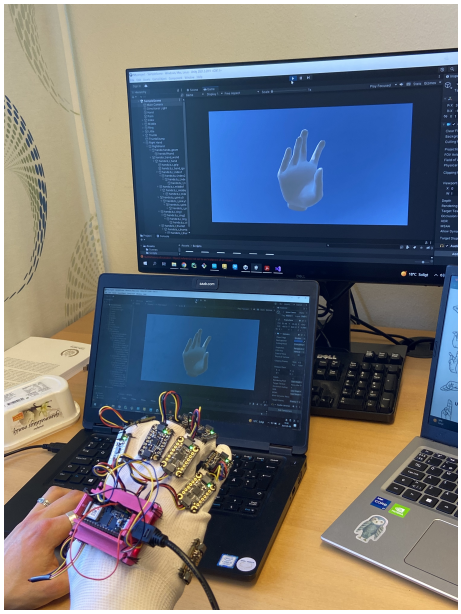
(p) P



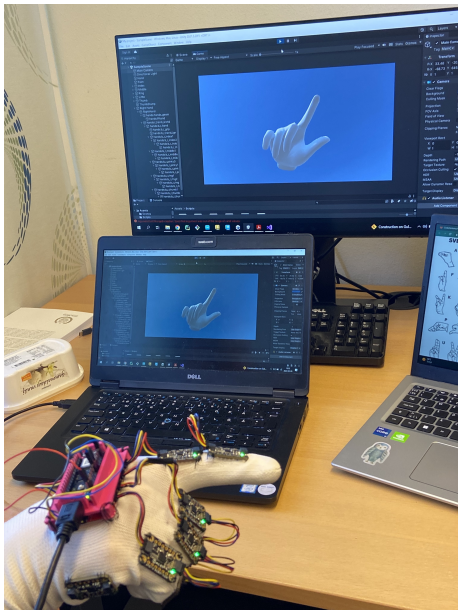
(q) Q



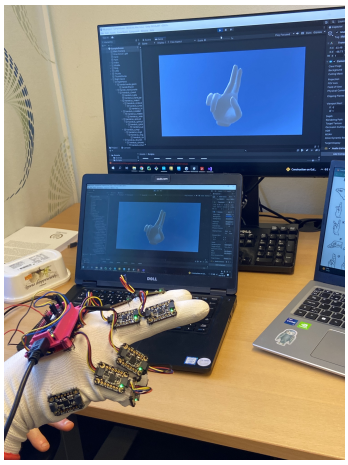
(r) R



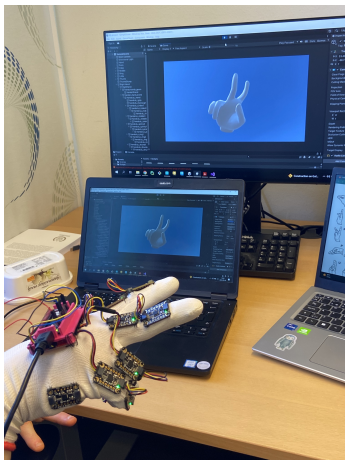
(s) S



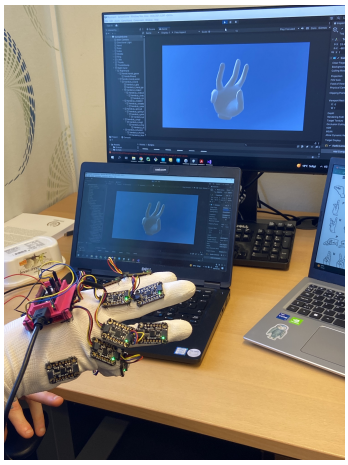
(t) T



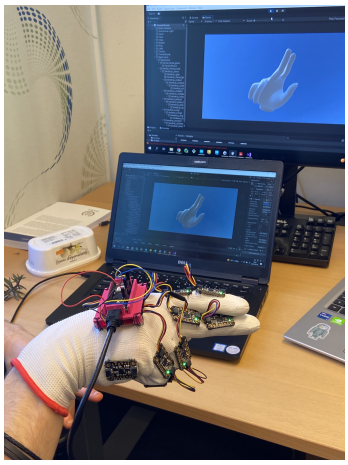
(u) U



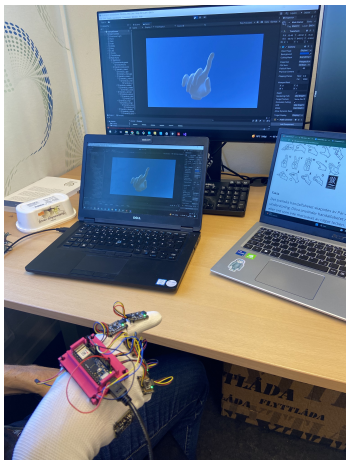
(v) V



(w) W



(x) X



(y) Z

Figure B.1: Preformed letters from the Swedish sign language, except y, å, ä and ö. Note that M was preformed incorrectly, it should be done using three fingers instead of four.

Systems and Control
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se

CHALMERS