UNICAM
Università di Camerino
1336

iSAS
nternational
nterdisciplinary
ntersectoral

School of Advanced Studies

Università degli Studi di Camerino
School of Advanced Studies

DOCTORAL COURSE IN

MATHEMATICS

XXXIII cycle

Development of a DataGlove for Gesture Recognition based on Inertial Measurement Units

PhD Student:
Francesco Pezzuoli

Supervisor:
Prof. Eng. Maria Letizia Corradini

“*Engineering is not about perfect solutions;*
it’s about doing the best you can with limited resources.”

Randy Pausch

Abstract

This thesis aims to deep analyse the application of Inertial Measurement Units (IMUs) for gesture recognition purposes, aiming to develop a Data-Glove for people with communication diseases as non verbal autistic child and people with aphasia. The specific field of application is gesture recognition. This research was conducted in collaboration with LiMiX with the final objective of create a device which can translate gestures in voice. The main idea was to build a wearable device able to give to the final user capabilities to rapidly customize his how dictionary of gestures and translate his signs in voice in real time. Different techniques for gesture translations has been adopted.

Contents

1 Introduction **3**

1.1 Applications . . . 3

1.2 Overview . . . 8

2 State Of Art **9**

2.1 A bit of History . . . 10

2.2 Input Method - Vision Based Systems . . . 13

2.3 Input Method - Data Gloves . . . 17

2.4 Gesture Learning Methods . . . 22

2.4.1 Results . . . 25

3 Talking Hands Platform **29**

3.1 Requirements . . . 29

3.2 Talking Hands Architecture . . . 31

3.3 Flex Sensors . . . 33

3.3.1 Resistive flex sensors . . . 33

3.3.2 Capacitive flex sensors . . . 35

3.3.3 Sensors with Stainless Steel Soul . . . 38

3.4 Inertial Measurement Units . . . 40

3.4.1 9 or 6 Degree of FreedomIMUs . . . 40

3.4.2 BNO055 . . . 42

3.5 Processing Units . . . 42

3.6 Talking Hands Prototypes . . . 44

3.6.1 Talking Hands Ergonomics . . . 49

3.7 Smartphone Application . . . 52

3.8 Interaction . . . 54

3.8.1 Reset procedure . . . 55

3.9 Discussion . . . 57

4 Gesture Recognition Process **59**

4.1 Static Gestures Recognition . . . 60

4.1.1 Translation through Scenarios . . . 62

4.1.2 Distance Function . . . 63

4.1.3 Gesture Recognition Algorithm . . . 64

4.1.4 Tests . . . 66

4.2 Considerations on proposed Static Gesture method . . . 67

4.3 Dynamic Gestures Recognition . . . 68
4.4 Experimental . . . 73
4.5 Results and Discussion . . . 74
4.5.1 Experiment 1 . . . 74
4.5.2 Experiment 2 . . . 75
4.5.3 User Independence . . . 76
4.5.4 Comparison with other works . . . 77

5 New Platform **81**
5.1 Future Hardware . . . 81
5.2 Future Design . . . 85

6 Conclusion and future work **91**

A Accelerometer as finger tracker? **93**

B Gesture detection system within relative reference systems **97**

C List of Publication **103**

Bibliography **105**

List of Figures

1.1 Talking Hands Render 4
1.2 Example of speech generating device for patients affected by cerebral palsy 6

2.1 Put-That-There 11
2.2 SignAll System - marked gloves 14
2.3 SignAll usage in a Welcome Desk 15
2.4 Kinect Sensors 16
2.5 Gesture recognition using single and double hand: (a) A single hand gesture (“welcome”) that was hard to estimate in Kinect, however, captured by Leap motion sensor (b) A single hand gesture (“office”) is captured better by Kinect (c) A double hand gesture (“thanks”) that was not correctly decoded by Kinect (d) A double hand gesture (“where”) that was not correctly decoded by Leap motion. Image from [45] . . . 17
2.6 Z-glove Architecture 19
2.7 Z-Glove System 19
2.8 Power Glove for NES (Nintendo Entertaining System) 19
2.9 Errors Estimating Velocity and Position from Acceleration 21
2.10 American Sign Language Alphabet 22

3.1 Simple Architecture of the System 31
3.2 Type of Communication (yellow) between MCU and peripherals (grey) 32
3.3 Exploded Diagram of Talking Hands’s bracelet 34
3.4 Scheme of a resistive flex sensor. (a) Top view: electrical contacts in grey, conductive film in black. (b) Lateral view: conductive film, in black, on top of a substrate, in a lighter color. (c) Bending the substrate causes a mechanical stress of the conductive pattern that leads to a change in its electrical resistance from [68] 35
3.5 How a Flex Sensor Works 36
3.6 Silicon Bend Sensor from Stretchsense and Circuit 36
3.7 Silicon Bend Sensor from Stretchsense Charateristics 37
3.8 Stretchsense’s MoCap Glove 38
3.9 How Splay Sensor works 38
3.10 Yarn-Based Stretchable Sensor Array - Schematic illustration 39
3.11 (Left) Linear fitting of stress-voltage - (Right) Generated voltage signals of the YASSA sensor 40

3.12 Comparison among SOC (System-On-Chip), and SIP (System-In-Package) from [74] 43
3.13 Palm 44
3.14 Back of the Hand 44
3.15 Cad of Talking Hands. a) System of Rings b) Hand Part c) Arm Part 46
3.16 Real First Prototype of Talking Hands 46
3.17 Lateral view of the second prototype 47
3.18 Top view of the second prototype 47
3.19 Current Prototype with Naked Hand Part 48
3.20 First Sketch - Organic Geometry 50
3.21 Second Sketch - Auxetic Geometry 51
3.22 First Render, using flex sensors 51
3.23 Home screen Architecture 52
3.24 Dictionary screen Architecture 52
3.25 Interaction Activity Diagram - Talk and Record Gesture 55
3.26 Reset Button Position 56
3.27 Explanation of Reset Procedure 57

4.1 Gesture recognition using Talking Hands System 61
4.2 Talking Hands Set 62
4.3 Schematic representation of three simple scenarios 63
4.4 High Level Flow Chart of Talking Hands 65
4.5 Successful Translation Histogram 67
4.6 Data without Segmentation 69
4.7 Segmentation of a data sequence with six gestures 69
4.8 Velocity computed from the data sequence of quaternion using the distance function $_{e}xtended_{e}qref : neq : quatDist$. 70
4.9 Velocity computed from the data sequence of quaternion using the Euclidean distance function. The red line indicates the threshold used for the segmentation. 70

5.1 Broken F-PCB's tracks after few hours incorrect usage, flex sensor's pin has been stressed during usage, white surface means broken bus 82
5.2 Full IMU Architecture 84
5.3 New Hardware Architecture 85
5.4 New microcontroller - Peripherals Communication 85
5.5 Second Render, using IMUs 86
5.6 Last Render, using IMUs 87
5.7 Bracelet Dimensions 88
5.8 Bracelets Comparison (render) 88
5.9 Talking Hands sizes 89

A.1 Hardware Architecture of accelerometers as finger tracker 93

A.2 Hand position in the experiment: (Left) Palm facing left, fingers up (Center) Palm facing left, front fingers, (Right) Palm down 94
A.3 Data sequence of the three conditions 94
A.4 Finger's flexion in the critical case 95

B.1 AGIS render 98
B.2 Example of user's and car's reference system 99

Acknowledgments This work could not be even possible without the collaboration with Dario Corona, Juri Bruciati, Athos Capriotti and Michele Di Carlo.
A special acknowledgment is addressed to the supervisor of this thesis Prof. Maria Letizia Corradini.

Chapter 1

Introduction

This thesis presents in details **Talking Hands**, an innovative wearable gesture recognition device that aims to support people with verbal communication impairment. Talking Hands acquires data through a data-glove and performs gesture recognition-algorithms on a mobile application which is able to process both static and dynamic gestures in real-time. Talking Hands' hardware, presented in Figure 1.1, consists of a sensorized hand module and an arm module, which acquires and pre-processes gesture's data. These data are sent over bluetooth to a smartphone application that associates the gesture to a word or an expression, which is then pronounced through smartphone's speaker.

This thesis is about the entire process and steps performed during Talking Hands' development over the last 3 years.
Main output from this thesis will be an industrial prototype that could efficiently help speech-impaired people on a daily basis, and can help people with different kind of verbal disease, rehabilitating vocalization. Talking Hands can also be useful in user's daily routine, for example when he/she is at home with his/her family and needs something: water, food, go to the toilet, etc. Talking Hands in those situations will be used as Augmentative and Alternative Communication system. Project's requirements are visible in Section 3.1.

Meaningful gestures, can be divided in *static* and *dynamic*. The former ones are configurations of hands and arms that do not involve any motion. Dynamic gestures, like "Good morning" and "House" in American SL (Sign Language), express their meaning also through the movement and they are usually more difficult to detect and translate. In this thesis, we will also compare the results of different machine learning algorithms for dynamic signs translation.

1.1 Applications

Talking Hands will help people with both temporary or permanent speech disorders, therefore they can finally increase their experience in social interaction and have better living conditions.

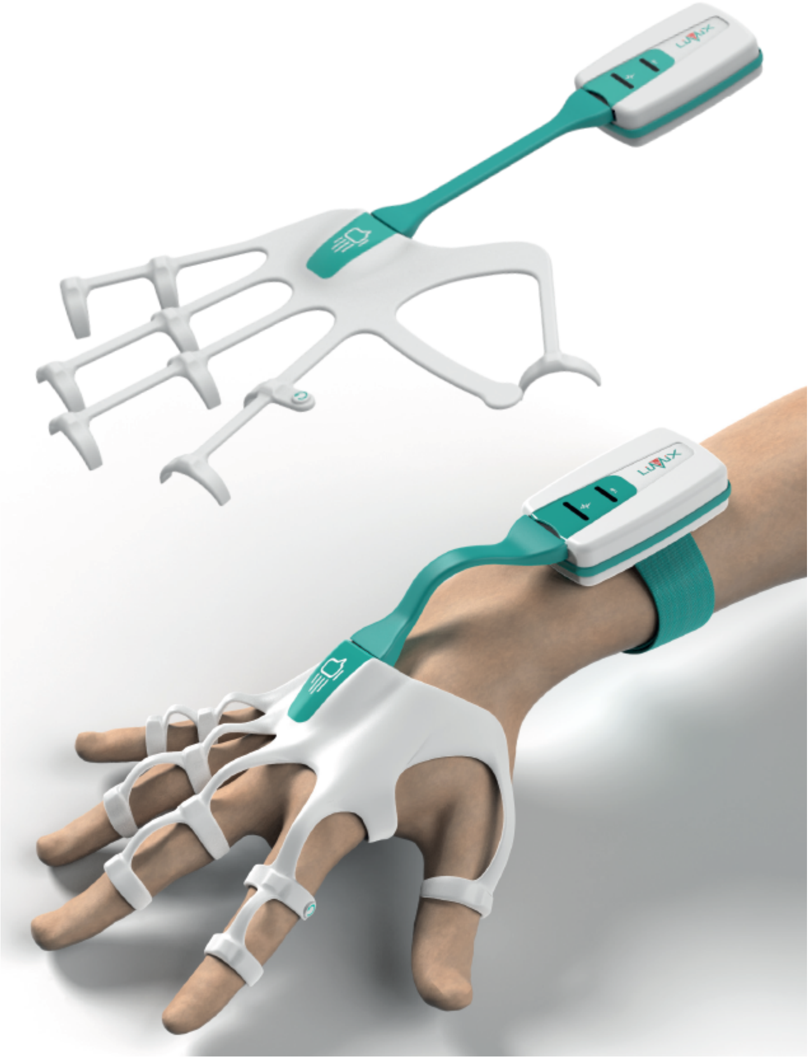


FIGURE 1.1: Talking Hands Render

Communication is perhaps one of the most important privileges enjoyed by the majority of people. It plays an indispensable role in everyday life helping to facilitate the process of sharing information, knowledge and development of personal relationships. The importance of communication cannot be overstated, being essential to access education, healthcare, employment, entertainment and social interactions (see, for example, [1–5]).

Talking Hands aims to give voice to people with speech impairments, supporting recovery or acquisition of basic verbal skills. Speech and language are central to the human experience: they are vital, indispensable means for people to convey and receive knowledge, thoughts, feelings as well as to participate in social interactions and relationships.
However, a huge number of people worldwide suffer from speech disorders and impairments: communicative disabilities in which natural speaking faces difficulty. This is

more common during childhood but they can occur at any age (stuttering, articulation, voice or language disorders) due to conditions like – but not limited to – Autism Spectrum Disorder, stroke, deafness, mouth or throat cancer, Down Syndrome, neurodegenerative diseases, etc., which result in a substantial inability to naturally or fluently communicate.

The prevalence estimated on the extent of people experiencing such disorders varies with the age and diagnostic criteria applied; however, it is estimated that:

- About 3% of the global population (i.e., 234 million people, Baukelmann, 2005) suffers from one of the 27 classified speech disorders;
- The prevalence of speech and language disorders among school-aged children ranges between 2.3-24.6% [6] in developed countries, median prevalence=5.95%;

In addition, specific studies report that:

- Disorders resulting from stroke (e.g., aphasia) occur in about 83% [7] of patients, with stroke representing the third leading cause of disability worldwide (WHO);
- Approx. 60% of 4 to 6-year old children with Autism Spectrum Disorder (ASD) shows moderate to severe language impairments [8]; the same is observed in about 97.8% of adolescents with the Down Syndrome [9]. In addition, the ability to speak deteriorates in patients suffering from ALS (Amyotrophic Lateral Sclerosis) in almost 60% of the cases [10].

Communication disability is associated with long-term disadvantage, placing individuals and their families at greater risk of adverse outcomes. Even though legislative measures to avoid discrimination exist in EU (e.g., Treaty on the Functioning of the EU, Art. 10 and 19), and support equal opportunities, fair working conditions and social inclusion for people with disabilities, several critical issues remain:

- Discrimination in the labour market: Verbal communication is a key skill for employers: 85% of them report that mere stuttering heavily decreases a person's employability [10]. As a matter of fact, young people with a history of speech disorder face huge barriers in their professional career, are unemployed for longer times respect to peers (47% unemployment ratio), and rate themselves as significantly less likely to get a job[11].
- Poor access to quality healthcare services [12]:People with communication impairment often require more frequent interactions with health services, but find them difficult to access and are more vulnerable during encounters. A growing body of literature reports that these people face poorer health and medical care outcomes compared to peers without disabilities (medical errors, delays in treatments, marginalization, inadequate care) [13],[14].

- Lower quality of education:Communication disabilities pose severe risk of educational failure (Education and employment outcomes of young adults with history of developmental language disorder, Gina Conti-Ramsden et al.,2018.). Access to quality schooling is a serious concern for parents of children with communication impairments (only 2% of them receive education with sign language).
- Social exclusion: People with communication diseases face unacceptance, stigmatisation and isolation; their families – mothers above all – are also much more frequently subject to emotional disorders (anxiety, depression etc.) [15] , besides the fact that disability is an established cause of social exclusion [16].

To remove these barriers, different assistive technologies [17] have been developed over the years. Among them, of key importance are the so-called Augmentative and Alternative Communication (AAC) technologies [18], ranging from the simplest picture board to PC, tablets, or programs synthesizing speech from text, (see Figure 1.2). Even though AAC tools can provide vital support, they are still assessed to be limited in performance, with only basic vocabulary and translation of words/concepts into speech through cumbersome and slow processes, costly and unpractical, making them substantially unappealing to users, not fitting all their needs, and therefore scarcely adopted. The dramatic impacts of communication disruption clearly show how urgent is the need for the development of functional and accessible technologies giving voice to people with speech disorders, to reduce for them of being left further behind in social participation and inclusion. Reader is addressed to [19] for in depth-analysis of this topic.

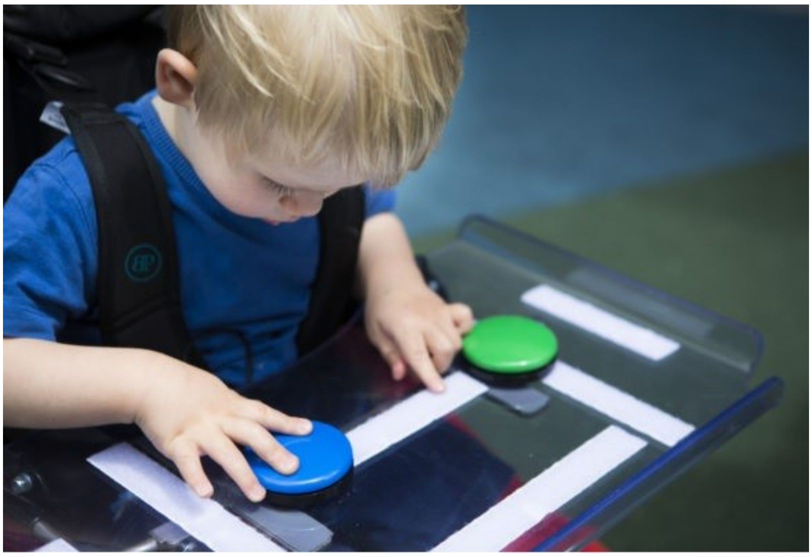


Figure 1.2: Example of speech generating device for patients affected by cerebral palsy

To analyse gesture recognition problem, let us examine definition of sign language. It is a similar problem and it is possible to find a wide literature about engineering approaches to the problem. Sing Language: a system of communication using visual gestures and signs, as used by deaf people. Sign language involves not only one hand, but also shoulders movement, facial expression and many other kinds of non verbal communication activities performed by the human body. This work is limited to gesture recognition using one hand, tracking movements/orientation of fingers, hand and forearm in the space.

Gesture Recognition's applications may vary. Input system and gesture recognition algorithms may be different, but once data are acquired and processed, it is time to decide what our system should do. Collected data can be used directly from the Data-Glove's microncontroller or can be send to another device. In Talking Hands, for example, data are sent via Bluetooth communication to a smartphone were the gesture recognition algorithm will run and select which stored gesture matches with the performed one, for detailed information reader is addressed to Section 4.1.

Other devices used by researchers are [20]: Computer screens, Liquid crystal displays and Speakers. LCDs and speakers sometimes are mounted directly of the Data-Glove.

During research activity we did also different tests and developed different prototypes to use the same/similar gesture recognition algorithm aiming to different use of the technology. For more information reader is addressed to Annex B.

1.2 Overview

This thesis explains in detail the overall process performed to achieve the current Talking Hands technology state, starting from the first prototype of Talking Hands shown in Figure 1.1.

In Chapter 3 we describe the hardware of Talking hands, sensors used to acquire information about the user's movements and the others electronic components. In this chapter we also analyse possible sensors inside a data glove that can measure the movements of fingers, hand and forearm. Clarifying why we used some sensors instead of others. In last sections of this chapter we describe designs and mechanical structures developed to overcome the issues of a wearable system for the hand, including how interaction between user and device works.

In Chapter 4 we illustrate an high level flow chart of the main parts of the translation algorithm, which is the core of the system, and see how dynamic and static gesture recognition algorithms may be integrated in the final application.

In the same chapter we describe the main strategy behind the static gesture recognition algorithm developed during this project, how dictionaries work and how we technically recognize gestures in real time. Relevant tests will be illustrated in the same section.

In the same chapter we show trials of different gesture learning methods, mainly machine learning methods for dynamic gesture recognition. We explain strategy developed for data segmentation, feature extraction and classification with relevant tests.

In Chapter 5 is described the new architecture of the system, replacing flex sensorswith IMUs. To do that, we need also to change the main microncontroller. We will discuss also about future mechanical/design and ergonomics.

Chapter 2

State Of Art

Before continuing, there is the necessity to underline a concept: in this thesis sometimes we refer to Sign Language Recognition as research field. We have to use Sign Language Recognition as keyword because something similar and comparable to Talking Hands only refers to Sign Language Recognition, but in reality what we do with Talking Hands is gesture recognition. For example in this chapter we will analyse also some research about Sign Language Recognition.

Data acquisition of the user movement has to be simple enough to realize a portable system. Second, recognition and translation of a meaningful gesture must be conducted in real time. The translation within a very large set of signs, such as an entire sign language, needs an heavy computation that can only be achieved in real time with powerful hardware and software systems. Last, a systems for sign language translation has to reconstruct the grammar structure of the phrases, because the sign languages are very different from their respective spoken languages. Different papers and studies of last decades face with these challenging tasks. For example, [21] and [22] are two surveys about the major challenges and tools for gesture recognition, in particular for video-based systems, while [23] presents the issues for an automatic sign language analysis. In [24] a framework for recognizing American Sign Language (ASL) from 3D data is presented, with a extended analysis of the sign language modelling.

In spite of these studies, nowadays there is not a commercial gesture recognition system that could improve vocal-impaired interaction.

The majority of works concerning gesture and SL recognition use cameras or other external devices for data acquisition (e.g. [25],[26],[27],[28], [29],[30],[31]).This approach can obtain all the data required for a perfect SL translation, such as position, configuration and movement of the hands, facial expressions, position and movement of the body.However, it has some disadvantages when applied in the realization of a SL translation system that can be used daily. In fact, cameras and position trackers do not allow system portability, since the users must rearrange their location whenever they move. Moreover, most of the papers rely on a simple background or other hypothesis about the environment, this being a strong restriction in a real-life application. The user is referred to [32] for an extensive survey about the gesture recognition works from 2009 to 2017 focusing on sign language. To overcome the cited drawbacks,

data-gloves have been adopted by a number of researchers. A major advantage is that gloves can acquire data directly (degree of bend, wrist orientation, hand motion, etc.), thus eliminating the need to process raw data into meaningful values. Furthermore, this approach is not subject to environmental influences such as the location of the individual or the background conditions and lighting effects, hence the acquired data are more accurate (the reader is referred to [20] for a detailed survey on system-based sensory gloves for sign language recognition from 2007 to 2017).
In literature there are systems that use cameras for data acquisition (e.g. [33],[34],[26] and [29]). The major drawback of these solutions is the system portability: the user is forced to be stand in front of the camera and to rearrange the camera location each time he moves from a place to another. For this reason, other systems are designed for gesture data acquisition, such as data gloves.

Classic literature about Sign Language Recognition aims to solve deaf community's problems building a technical solution that will translate one sign language, usually the American Sign Language as in [35].
SignAll is the top technology for Sign Language Recognition based on a video system, SignAll has developed a technology leveraging AI and computer vision that is able to recognize and translate a sign language. As of now, SignAll is the only company in the world which has a commercially available product utilizing sign language to spoken language technology. We refer to SignAll because is the only technology available on the market for Sign Language Recognition, therefore it performs also gesture recognition. We refer to SignAll also because there is not efficient and comparable projects on the market which performs Sign Language Recognition or gesture recognition. This will be discussed more in detail in Section 2.2.

2.1 A bit of History

Since late 70's scientists tried to figure out how to remove physical devices as keyboard, pointing devices as mouses. In fact in [36] wrote in 1997 authors reported that Chris Schmandt and his colleagues at MIT Architecture Machine Group demonstrated the "Put That There" system [37] a system able to recognize pre-defined sentences and word, a maximum of 120 words exactly, treating words as a set of word reference patterns. An example on command is "Create a blue square there". After the vocal command, the user uses a pointing device to point on the screen "where" the system needs to create the object. The space position and orientation technology was made by Polhemus Navigation Science Inc. The system was called ROPAMS (Remote Object Position Attitude Measurement System) and it is based on measurements made of nutating magnetic field. All is based on magnetic field changes about three mutually orthogonal coils to correspond to x,y and z spatial axes. One is mounted close to the chair where the user sits and another is mounted on user's wrist like a watch.

A frame of the original video shown in Figure 2.1.

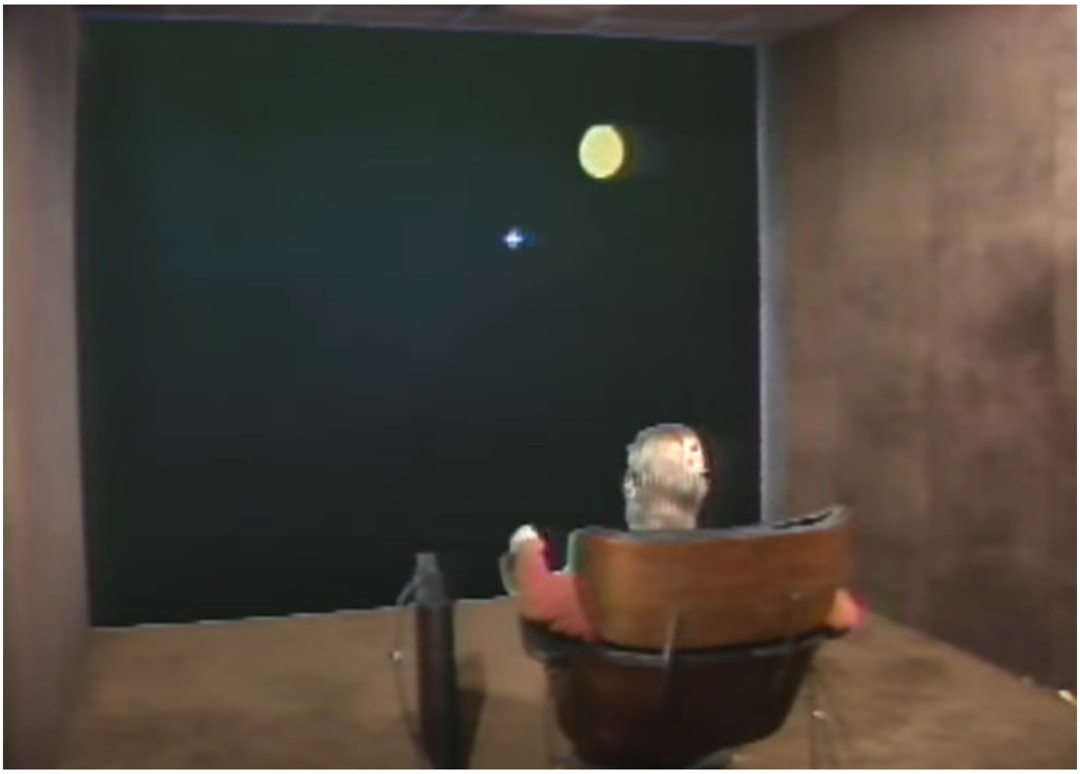


FIGURE 2.1: Put-That-There

It is possible to assume that the first device that tries to recognize gestures, in this case working as pointer (as a mouse) was developed (published) in 1979. [36] continues asserting that in the last two decades (1980-2000) promises about remove classical pointing devices and keyboard has largely fallen. Systems that use natural gesture are mostly confined to research laboratories. The scenario described by authors does not evolved a lot since late ’90s.

Till now data gloves or gesture recognition technologies did not replaced mouses and keyboards. Integrating that technology in operative systems is the minor problem; the biggest problem is that a really cheap, reliable, ergonomic and friendly to use technology is still not available on the market.

Nowadays to develop a device similar to ROPAMS, mapping an exact position on a screen will be really cheap and easy to use. The problem is that as humans, we suffer of the "gorilla-arm syndrome": if we pose for a short period of time with arm fully extended arm, hanging our arm in the air using ROPAMS system, we will start to feel fatigue and discomfort really soon, also after 10 minutes usage.
This is the main motivation behind the failure of this kind of pointing devices.
One thing is to map a pointing gesture on a surface as a screen and another one is to recognize hand gestures. Talking about gesture recognition algorithm , the former is really easy to solve, the second is harder.
In "Gesture and Sign language in Human-Computer Interaction" [36], Alistar D.N. Edwards in "Progress in Sign Language Recognition" divided gestural interaction into

three categories:

- Natural Gestures: spontaneous gesture made by people in human to human interaction supporting other forms of communication as the verbal form.
- Synthetic Gestures: used in human-computer interaction and designed specifically for one application.
- Virtual Reality interaction: where gesture mimics a real word action. Natural and Synthetic Gestures may be included in this set.

Harling et al. in their work: "Hand Tension as a Gesture Segmentation Cue" [38] explain that sign language recognition can be divided in four mainly classes, considering gestures simply as any possible movement that the human hand can make. The first group consists of static hand shapes, where only positions of the fingers are relevant; the second group consists of dynamic hand shapes, where the gesture is made also changing and moving hand during time. Considering also hand motion and orientation the main two groups can be subdivided to:

- SPSL: Static Posture, Static hand location
- DPSL: Dynamic posture, static hand location
- SPDL: Static posture, dynamic hand location
- DPDL: Dynamic posture, dynamic hand location

In [38] author considered hand configuration and motion to divide gesture into this four group, we can refer to them lately.

One thing is to recognize hand gesture and fingers postures and another one is to recognize sign languages in all their complexity. In [38] authors mistook gestures for signs of a sign language, as many researchers did.

Today a more accurate definition of sign language is reported also in a lot of studies, especially [20] summarized in this list:

1. Hand position and orientation:

   Palm

   Proximal Phalanges

   Intermediate Phalanges

   Distal Phalanges

2. Forearm position and orientation
3. Wrist position and orientation
4. Elbow position and orientation
5. Shoulder position and orientation

6. Arm position and orientation
7. Facial Expressions

   Happy Face

   Angry Face

   Lips Movement

   Shake the Head
8. Movements talking about all the previous points in the list, for example about forearm rotation performing a gesture:

   Backward or forward

   Clockwise or counterclockwise

   Left or right

As previously explained, it is important to underline that this research does not aim to translate SLs or a particular SL as American Sign Language.

2.2 Input Method - Vision Based Systems

The majority of works concerning gesture and SL recognition use cameras or other external devices for data acquisition (e.g. [25, 26, 29, 30, 39–42]).This approach can obtain all the data required for a perfect Sign Language translation, such as position, configuration and movement of the hands, facial expressions, position and movement of the body.

Vision Based Systems usually use one or multiple cameras to acquire gestures. Anyway there are different techniques to capture gesture's information using a vision system. Using for example normal cameras extracting 2D images and a 2D matrix of pixels, others use techniques based on non standard cameras as thermal cameras or IR cameras. Some studies are using more invasive techniques requiring gloves with markers or a particular colour distribution as in [35].

Some studies as [43] developed easy and efficient way to recognize gestures using a single camera. As requirement the user needs to frame camera onto one hand, after that the user can perform a sign, the system will recognize mainly:

1. Skin coloured object,
2. If there is an hand in camera's frame,
3. Fingers configuration.

Extracting this information [43] recognizes 4 different gestures (thumb up, thumb down, point and stop gestures) with final aim of answering to questions coming from a robot using a probabilistic framework. Some other works do hand shape recognition, recognizing just Sign Language alphabet. For example it [44] recognizes using a particular feature extraction American Sign Language's numbers from 0 to 9.

In a more complex system like SignAll [35], the solution goes beyond hand shape recognition or finger configuration recognition, as simultaneously tracks the:

1. position of the hands (relative to each other)

2. hand form/hand-shape (position and direction of fingers)

3. mimicry, facial features

4. position of hands and fingers (relative t to other body parts)

5. movements etc.

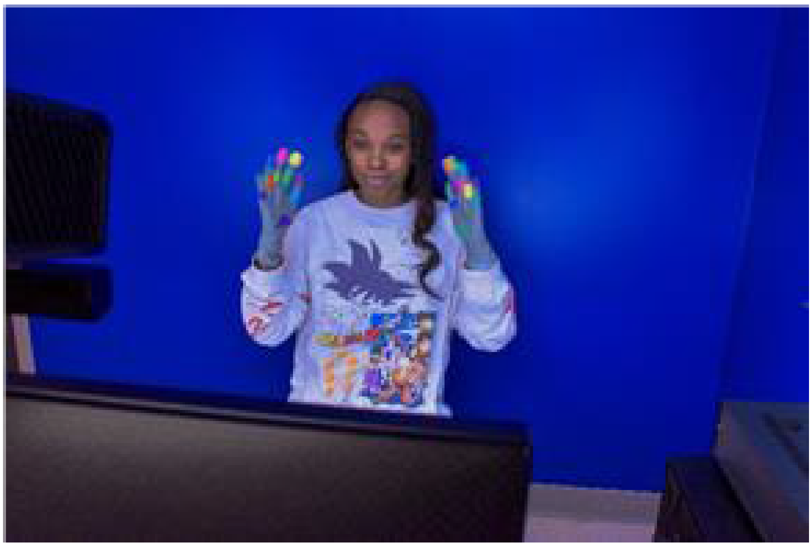


Figure 2.2: SignAll System - marked gloves

SignAll is different from classic research because aims to develop an useful product not only to translate the entire data-set of a Sign Language. SignAll has been cited because today is the only available product in the world doing American Sign Language recognition, based on cameras and marked gloves. For this motivation we think it needs an extended explanation to better understand Vision Based Systems for gesture recognition.

From SignAll website is possible to read requirements for a correct usage of the system:

- Both users (hearing and Deaf) must have an intermediate knowledge of written English The Deaf user should use clear ASL, avoiding regional signs

- Users should avoid clothing:

  Extremely colourful

  Excessive clothing (coats, layers)

  Oversized jewellery (bracelets)

  Hats that shade or occlude the face

- The system is sensitive to light, including direct sunlight; the backdrop should be free of people and objects.
- Hardware needed to utilize the system is the following:

  Cameras / camera mounts (desktop version)

  Customized lighting (desktop version)

  PC

  Monitor

  Touchscreen / Tablet

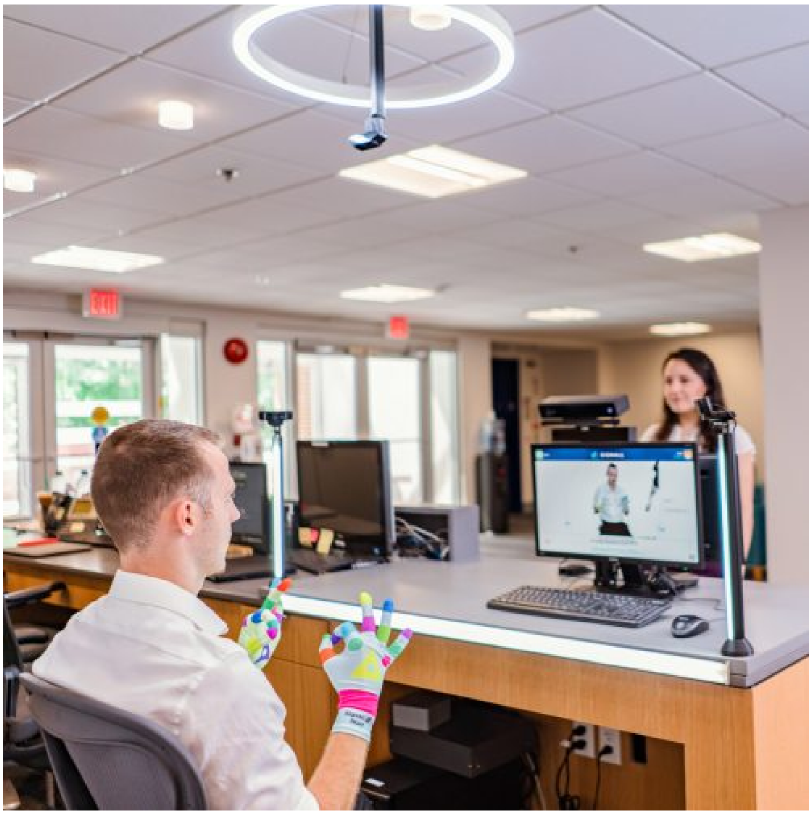


FIGURE 2.3: SignAll usage in a Welcome Desk

However, reading SignAll requirement's list is easy to understand that it is not a portable system and not easy to use in different kinds of environment. Environment or clothing (e.g. light, background, objects etc.) dramatically influence the right operation of SignAll.

In fact, cameras and position trackers do not allow system portability, since the users must rearrange their location whenever they move. Moreover, most of the papers rely on a simple background or other hypothesis about the environment, this being a strong restriction in a real-life application.

Another case study regarding Vision based systems has been done in [45], a novel multimodal framework for isolated Sign Language Recognition has been developed using Microsoft Kinect and Leap Motion as input devices. Leap motion is kept below the hands and Kinect is placed in front of the signer, capturing horizontal and vertical movements of fingers while the user performs gestures.

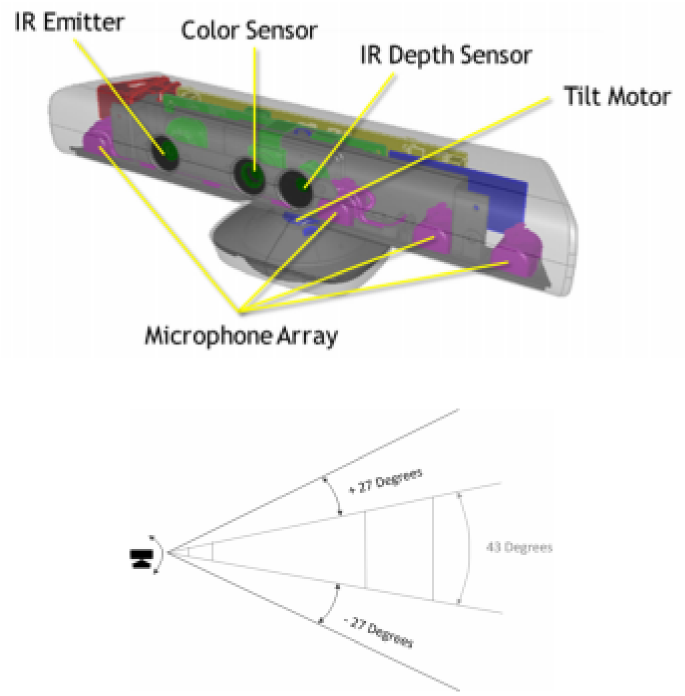


Figure 2.4: Kinect Sensors

Kinect uses one RGB camera, as "colour sensor", an IR receiver which understand deepness of an object thanks to IR emitter placed on the other side of the colour sensor as illustrated in Figure 2.4, then there is a tilt motor, thanks to this motor Kinect system can follow recognized object having a total view of 97 (43+27+27) degrees on the vertical plane. There are also 4 microphones (24 bit 16kHz ADC), anyway audio in Sign Language Recognition applications is not used.

Leapmotion uses two 640x240-pixel near-infrared cameras spaced 40 millimetres each other with infrared-transparent window, operates at 120Hz capable of image capture within 1/2000th of a second and e is able to discern 27 distinct hand elements (bones and joints).

In [45], authors proposed a system which recognizes 50 different gestures based on a framework of 7500 gestures (words) of Indian Sign Language (ISL) using Hidden Markov Model (HMM) and Bidirectional Long Short-Term Memory Neural Network (BLSTM-NN) based sequential classifiers.

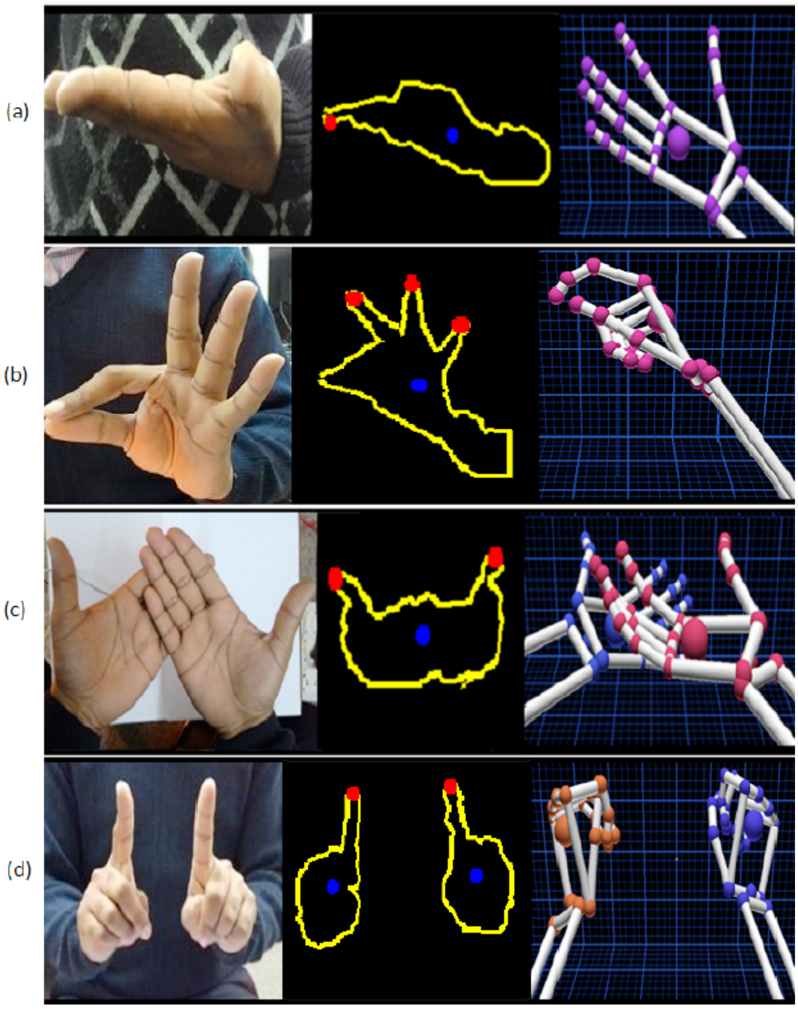


FIGURE 2.5: Gesture recognition using single and double hand: (a) A single hand gesture (“welcome”) that was hard to estimate in Kinect, however, captured by Leap motion sensor (b) A single hand gesture (“office”) is captured better by Kinect (c) A double hand gesture (“thanks”) that was not correctly decoded by Kinect (d) A double hand gesture (“where”) that was not correctly decoded by Leap motion. Image from [45]

2.3 Input Method - Data Gloves

A Data-Gloveis a sensorized wearable device usually shaped as a classic glove. In 1982 Thomas G. ZimmermanOffsite Link of Redwood City, California filed a patent (US Patent 4542291). In 1987 the first article about a Data-Glove was published by Thomas G. Zimmerman et al. in [46].
Authors presented two versions of Data-Gloves:

- Version 1: the cheaper one. It includes 14 flex sensors disposed to read abduction between each finger and flexion on metacarpophalangeal joints and proximalinterphalangeal joints. See Figure 2.6
- Version 2 (Z-Glove): the expensive one. It includes 14 flex sensors as version 1, plus 3SPACE system from Polhemus Navigation Sciences Division. The

> 3SPACE uses low frequency magnetic fields to measure six degrees of freedom. The small 3SPACE sensor is mounted on the dorsal side of the hand between the glove's two layers. The 3SPACE is connected to one of the serial ports on the Apple Macintosh computer, see Figure 2.7.

The 3SPACE requires no filtering of data. Pholemus Inc. designed the ROPAMS system for [37]. Probably 3Space is based on the same technology and it is the evolution of ROMPAS.

The research done in [46] was passed to Nintendo Co. Ltd. The first wired glove available to home users in 1989 was the Nintendo Power Glove (see Figure 2.8) built to work with the NES (Nintendo Entertaining System). The Power Glove was designed by Mattel, based on Zimmerman's Data-Glove. Nintendo's Power Glove did not use the 3SPACE navigation system but used a series of ultrasonic sensors. There are two ultrasonic transmitters in the glove and three ultrasonic receivers around the TV monitor. The ultrasonic speakers transmit few pulses of 40 kHz sound and the system measures the time it takes for the sound to reach the microphones. A triangulation is performed to determine the X, Y, Z location of each of the two speakers, which specifies the yaw and roll of the hand. Pitch cannot be measured.
Power glove was discontinued after one year, probably because it did not work as promised and because it was not so intuitive to use. Power Glove was sold for 75$, equivalent to $155.30 nowadays.

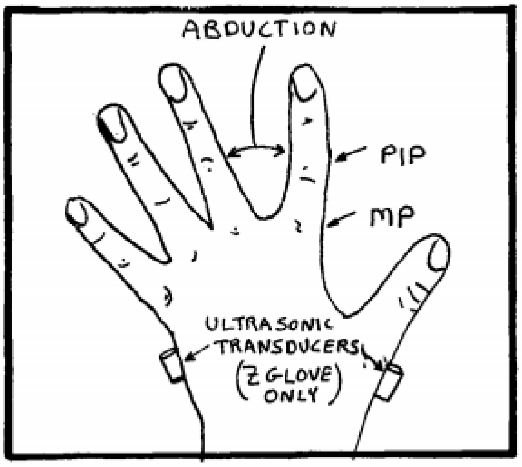


Figure 2.6: Z-glove Architecture

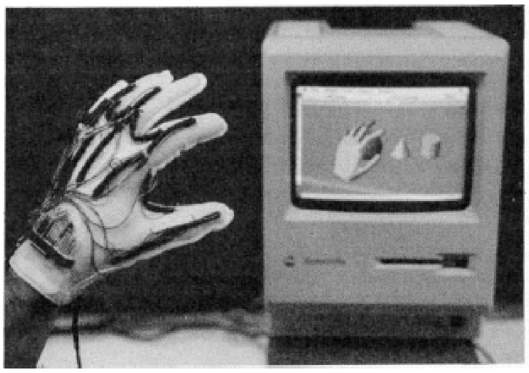


Figure 2.7: Z-Glove System

Figure 2.8: Power Glove for NES (Nintendo Entertaining System)

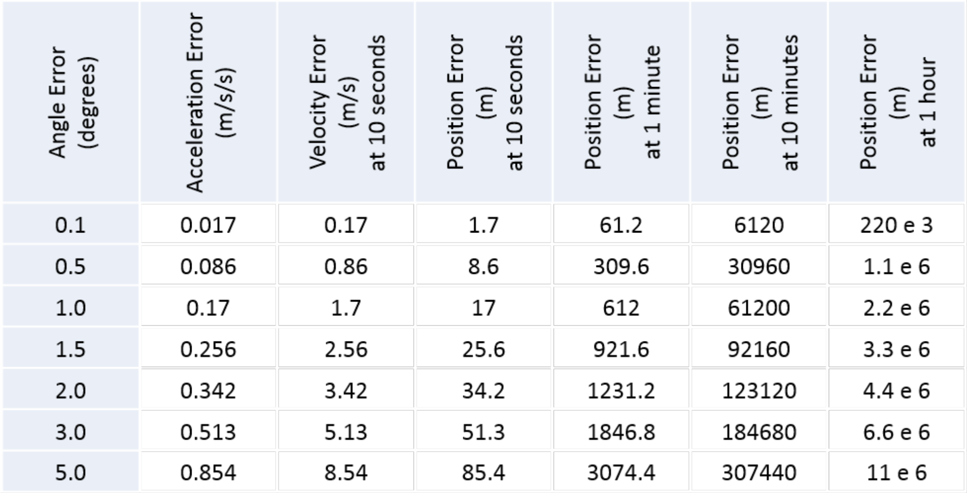
Nowadays after Data-Gloves evolution, is possible to say that Data-Gloves use different kinds of sensors to understand motion of hands (including fingers), sometimes also arms and shoulders.
Usually data gloves are connected to other devices, i.e. Personal Computers or Mobile Devices, sending them data about:

- Position;
- Velocity;
- Orientation;
- Acceleration;

of hand, arm, shoulder and fingers or some of them.

The other component connected to the glove is usually a personal computer, a microprocessor or a smartphone that can elaborate the data and interpret them in different manners, depending on the specific application.
As an example, let us consider a data glove used for virtual reality like the Power Glove. This glove needs to give to the console/PC data about hand's and arm's orientation, including fingers' position. From a theoretical point of perspective, it is possible to achieve this goal using an accelerometer for each finger, one for the hand and one for the arm. The position can be obtained by integrating twice the acceleration measures provided by accelerometers.

However, in practice, it is not possible to attain a reliable estimation of position using this procedure, unless very expensive and sophisticated accelerometers are available. In fact integration errors on cheap accelerometers (less then 10USD) are really huge, an error within the order of 3Km/min. If the reader needs more information about this problem, reader is addressed to [47]. The problem of integrating acceleration to have velocity and position using commercial accelerometers is explained in detail in [47].
We can also directly refer to the Figure 2.9 for reader convenience. Figure 2.9 gives us a better understanding of expected errors. It suggests that is not recommendable to use only accelerometers for this kind of tasks.



Angle Error (degrees)	Acceleration Error (m/s/s)	Velocity Error (m/s) at 10 seconds	Position Error (m) at 10 seconds	Position Error (m) at 1 minute	Position Error (m) at 10 minutes	Position Error (m) at 1 hour
0.1	0.017	0.17	1.7	61.2	6120	220 e 3
0.5	0.086	0.86	8.6	309.6	30960	1.1 e 6
1.0	0.17	1.7	17	612	61200	2.2 e 6
1.5	0.256	2.56	25.6	921.6	92160	3.3 e 6
2.0	0.342	3.42	34.2	1231.2	123120	4.4 e 6
3.0	0.513	5.13	51.3	1846.8	184680	6.6 e 6
5.0	0.854	8.54	85.4	3074.4	307440	11 e 6

Figure 2.9: Errors Estimating Velocity and Position from Acceleration

Data glove may also be realized with sensor parts based on different principles:

- Based on resistive or capacitive technology, those sensors read flexion of a two bodies between a joint. Those kind of sensors are the widely used, two separate sections will be used to describe them 3.3.1, 3.3.2;
- Optic fiber: data are extracted observing light's defelxion. Usually those sensors are composed by a light transmitting material as transparent PET, light emitter and light receiver. Light emitter are usually LEDs, normal LEDs or infrared LEDs with its own receiver (light sensor or IR LED receiver); those sensors are really cheap and can be usually handcrafted. Resolution and robustness may be compared to resistive flex sensors.
- Magnetic Hall effect sensors: they usually placed directly on the top of each finger with a strong magnet on the back of the hand. When a sensor placed on the top of a finger moves, hall effect sensor recognizes changes about magnetic field orientation and those information will be useful to recognize finger's flexion. This system guarantees high level of accuracy and has low manufacturing cost. It is not suitable to use this finger tracking system in combination with an IMU because the strong magnet on the palm will badly influence the 3 axis magnetometer mounted in the 9DOF IMU. An example of this kind of sensors used for finger tracking is available [48].
- Pressure sensors, using data collected from pressure sensors placed on the glove. Those kind of sensor which measure force applied to its own surface can be used to understand if the user is touching a part of its Data-Glove improving recognition performances, for example if we are doing the "O" gesture (as in Figure 2.10) thumb's fingertip with index's fingertip and a pressure sensor is placed on one or both fingertip we can be sure that the user is performing the "O" gesture instead of the "C" gesture (see Figure 2.10) because he/she is touching his/her

fingertips.

- Accelerometers, in [20] accelerometers are reported as finger tracker. During project development we tried to use accelerometers as finger tracker, and we reported that it is not possible to use an accelerometer as finger tracker. The reader is addressed to Appendix A for a detailed explanation.

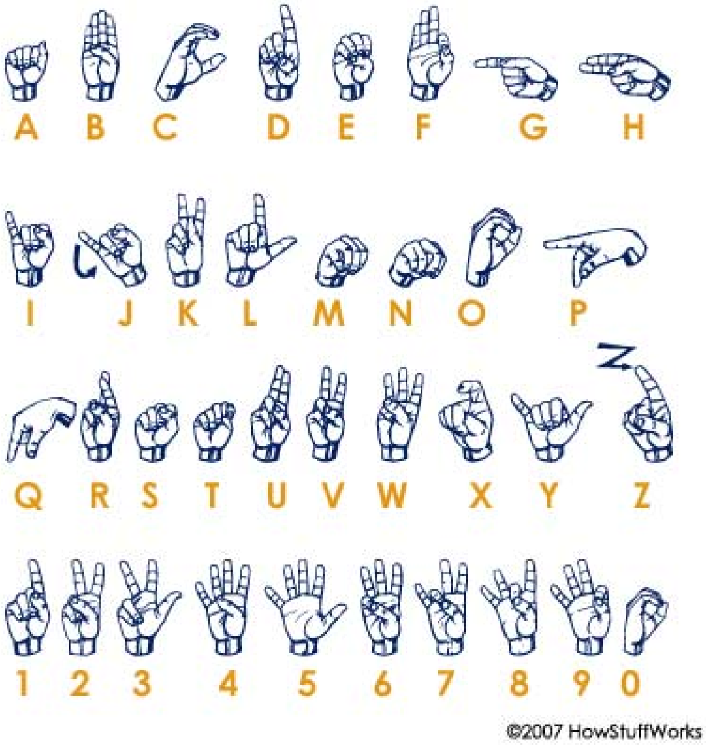


FIGURE 2.10: American Sign Language Alphabet

2.4 Gesture Learning Methods

The Software and firmware development for gesture recognition is related to methods used to classify gestures.

In gesture recognition field, information coming from Data-Gloves are input for our classifiers, data are time-varying signals. It is possible to define two different kinds of gesture recognition: one is the dynamic gesture recognition, usually referring to Sign Language Recognition, then we have the simplest one, called static gesture recognition. In this thesis we deal with both problems, respectively in Section 4.3 and Section 4.1. We can imagine a static gesture as a picture of user's hand, arm and finger. Acquiring that picture we can analyse which gesture the user is performing. A dynamic gesture instead is a short video of the user's hand, arm and finger; we have to analyse the gesture and how the user is moving articulations. For example if I am rotating my index finger clockwise or counter-clockwise it will mean two different things, they are

two different gestures. This is true if I am doing dynamic gesture recognition, will not be true if I am doing static gesture recognition. If I am doing static gesture recognition the gesture will be the same because motion is not taken into account. A sub problem in both static and dynamic gesture recognition is to understand when the user is really performing a gesture. This problem is known as segmentation, more about segmentation will be explained in Section 4.3.
There are different approaches to solve the problem. It is possible to develop a gesture recognition algorithm directly on a Data-Glove to recognize a small set of gestures, storing gestures' information directly into the Data-Glove or the gesture recognition device, as in [49, 50]. It is also possible to use a Data-Glove only as input device, using a more powerful machine to run the gesture recognition algorithm.

Gesture Recognition problem can be viewed from a pattern recognition standpoint: *"The field of pattern recognition is concerned with the automatic discovery of regularities in data through the use of computer algorithms and with the use of these regularities to take actions such as classifying the data into different categories"* [51]. Theoretically a gesture has the same pattern every time, for this reason Pattern Recognition fits perfectly as automated strategy for gesture recognition.
Set of methods for pattern recognition includes a variety of algorithms. Depending on final application we can choose for example parametric classification methods as: Linear Discriminant Analysis, Quadratic Discriminant Analysis. Or Non parametric classification methods as: Decision Trees, Artificial Neural Networks (ANN) and Support Vector Machine (SVM). For non-parametric classification we mean that no distributional assumption regarding shape of feature distributions per class is known; viceversa for parametric approaches distributional shape of feature distributions per class is known, as the Gaussian shape for example.

The artificial neural network (ANN) is the most popular method used for machine learning in the gesture recognition field as reported in [20]. ANN are based on units (nodes) connected by directed links. A link from an unit a to unit b is used to propagate the activation between two nodes. Each link has a numeric weight associated to it and provides the output of one neuron (node) as an input to another neuron. ANN are usually represented as weighted graphs. Neurons are described also as set of layers. Different layers perform different transformations on input signal. Signals travel from the first layer, called input layer, to the output layer (last layer). Neural networks needs to be trained by processing examples, each example provide to an ANN a known input, this input is pre-associated to the right-known output. This procedure forms a weighted connection between the two. Those information are then stored within the ANN data structure. When trained enough, ANN should be ready to perform classification even on not pre-learned examples and without being programmed with specific rules. For more information about artificial neural networks the reader is addressed to [pp.727-737] [52].

Hidden Markov Models (HMMs) have been also a popular technique in gesture recognition field; HMMs are noticeable also in many different applications as computer vision and speech recognition. HMMs are based on Markov Chain Theory. An example of gesture recognition using HMM is [53]. Galka et al used Parallel Hidden Markov Model (PaHMM). In this study, researchers say that this model can be successfully used for Automatic Speech Recognition and Automatic Sign language Recognition. Each channel of the PaHMM corresponds to a group of features which describes different articulations: phalanges, hand, arm. In this work the recognition for a single channel is performed by a token passing algorithm and an analysis of the N-best list which contains values (scores) obtained by each gesture model. For more details about PaHMM and how they are used for gesture recognition the reader is addressed to [53].

Support Vector Machines (SVM) is one of the most popular approach for supervised learning; they are learning algorithms that analyse data used for classification and regression analysis. After having trained a SVM, it will store most of the training examples, associating each example to a class (or set), and when receives a new example it will associate it to one of the classes. For more detail about SVMs reader can reefer to [pp.744-748] [52]. Reading descriptions of ANN and SVM they looks pretty similar, usually is hard to understand which one is the best. The difference is mainly about how non-linear data is classified. Basically, SVM utilizes non-linear mapping to make the data linear separable. However, ANN employs multi-layer connection and various activation functions to deal with non-linear problems. An interesting study about this topic has been edited by Jinchang Ren [54] on classification of MCCs in mammogram imaging.

A strategy we want to underline, has been used by a gesture recognition commercial product called Myo armband. Myo device turns electrical activity in the muscles of a user's forearm into gestures to control computers and other devices [50], it uses a 9-dof IMU and electromyography (EMG) sensors as input data.
Myo's gesture recognition method is interesting because is an hybrid solution in terms of dividing computational load for gesture recognition between the device (a bracelet) and a PC. Myo recognizes gestures using an ANN. Which kind of artificial neural network has been used is not public declared, but the procedure to perform gesture recognition is really clear. First of all, the user has to train the artificial neural network connecting the Myo Armbend to his own PC, than he has to perform 5 different gestures when Myo's software asks. User's input will be used to train the ANN. Once the artificial neural network has been correctly trained, the software on the PC will re-program the bracelet with the trained ANN.
When this procedure is correctly completed, gesture recognition is directly performed on the bracelet using trained artificial neural network.

2.4.1 Results

Most of research deals with static gestures, achieving high levels of accuracy, even 99% [55–58]; in [55] an applicative hand glove architecture is described. It has five flex sensors, three contact sensors and one three-axis accelerometer that serves as an input channel. It can translate American sign language in form of alphabet and it can transmit data to a mobile phone; and the overall translation efficiency is about 83%. Researches in this study do not explain which gesture recognition algorithm has been developed. Similar projects are described in [56–58].
In [56] 9 flex sensors, 11 contact sensors (to detect if user is touching some part of his hand) and one accelerometer have been used as sensing parts with samples at 500Hz. This study reports an accuracy on gesture recognition process of about 92%. After feature extraction, Principle Component Analysis (PCA) was used for the classification in this study. For more information about PCA, reader is addressed to [56].
In [57] a Data-Glovecomposed of 10 flex sensors (two on each finger), an IMU on the hand and touch sensors between each finger have been used. They tried to recognize SALS (South African Sign Language) collecting data of different participants. Using the described Data-GloveḂased on this hardware, they tried different methods: Elliott neural network, log-sigmoid neural network and SVM. They respectively obtained 96%, 94% and 99% of recognition accuracy. In [58] two flex sensors, one on index finger and one on medium have been used as sensors, together with an ADXL345 accelerometer. At the end of their experiment they came out with 93.75% gesture recognition accuracy, to understand which gesture the user is performing they used Euclidean distance, the most similar stored gesture respect the one performed by the user will be selected.
The hardware in [55–58] is almost the same, and they can vary for some details of implementation and for mathematical techniques for signs detection and translation.

Others researchers have faced the problem of dynamic gestures recognition (e.g. [59–61]) with a classification accuracy between 78% and 94%. In [59], a wireless hand gesture recognition glove is proposed for real-time translation of Taiwanese sign language. This work does not use any machine learning technique, but they use a brute force comparison, recording the former posture, how long does a posture persists, the orientation of palm and the motion trajectory. The accuracy achieved by the authors was 94% recognizing 5 dynamic gestures. Even in [60] a glove-based system to recognize a sign language is proposed. In this case, the data-glove comprised of 10 tilt sensors to capture finger flexion, an accelerometer for hand motion and an Arduino platform with a Bluetooth module to connect it to a mobile device and send the recognized gesture. Since the recognition is performed directly by the microcontroller, which has limited memory, only 9 gestures are recognized, with an accuracy of 78% for dynamic gestures. No information about the used algorithm/method has been shown in the paper. In [61], a data-glove system is proposed for the automated recognition of Greek sign language isolated signs. In this case, the glove uses a five channel surface electromyogram and a 3D accelerometer to acquire data. The data was analysed using Intrinsic Mode Entropy [62], achieving an accuracy of 93% on a dataset of 60 isolated signs. However, the classification technique was briefly described.

In our opinion, the most remarkable papers about dynamic gesture recognition through data gloves are [63] and [64]. In [63] the commercial sensory glove Cyberglove^{TM} is exploited to acquire data about hand configuration, while the Flock of Birds® motion tracker provides data about hand position and orientation. Different features for each gesture (distance and time of the gesture; bounding boxes; position, orientation, hand-shape and velocity histograms) are extracted and all these features allow to describe a gesture with a 151-dimensional vector. The tests of [63] show a 92% (277/300) of translation accuracy between a set of 50 isolated dynamic gestures, using a multi-layer Artificial Neural Network.

In [64], a publicly available dataset of 95 sign categories is used. Data consist of 11 channels of information for each hand: spatial coordinates of the hand, the roll, the pitch and yaw rotation angles of the wrist and a bend coefficient for each finger. The real novelty of that paper was the application of a Bayesian-like paradigm, known as hierarchical temporal memory [65], for sign language recognition. Their method learns the spatio-temporal data structures and transitions that occur in the data without depending on manually predefined features to be searched for and works well in real time. They reach 91% recognition accuracy for a dataset of 95 categories of Australian sign language.

The differences between our approach and the one presented in [64] will be further discussed in Chapter 4. Table 2.1 shows further details regarding the databases used in most of the previous studies, such as the type of gestures used with the frequency of conducting these gestures. In addition, the table clarifies who created the signs alongside the number of performers. The total number of samples used in the experiments are known. Respect to studies from Table 2.1 our aim is to recognize any gesture and not only a fixed Sign Language, mainly because users will not be restricted to deaf people, Talking Hands aims to help people with different communication diseases. Users must be able to choose which gesture they can associate to a word, because in 99% of the cases they don't know SLs. Some study uses a pair of gloves, this is equal to have a larger set of gesture and more accuracy because user's has doubled possible configurations. Talking Hands system uses a single glove to contain hardware costs, being also a less invasive wearable system respect to one which uses a pair of gloves. Talking Hands will be fore sure less accurate using a fixed size dictionary e.g. 200 words dictionary compared to a system which uses a pair of gloves. Respect to some study Talking Hands system has not been tested on a large amount of samples and from different users. If Talking Hands system will be a product available on the market an extremely large study may be conducted on every user/group of users.

Device/Components	Language	Gesture	Samples per Gesture	Gesture Performer	Sample Size
five flex sensors	American Sign Language	four gestures			
five flex sensors, accelerometer, and tactile (contact) sensor	American Sign Language	set of 8 gestures A-H	10 times		80 samples
fiveflex sensors and ADXL335 accelerometer	American Sign Language	26 gestures alphabet and 10 more gestures to numbers			256 samples
8 touch sensors	American Sign Language	numbers 0 to 9 and the 26 English alphabets, A to Z	30 times		1080 samples
five flex sensors and a 3D accelerometer	American Sign Language	American National Corpus is used A-Z and "space" plus "full stop"	5 times	6 females and 4 males age between 20–26	1400 samples
six inertial measurement units (IMUs) accelerometer	American Sign Language	American Sign Language (ASL) letters without letters J and Z	one time	data was collected from 9 participants	216 samples
5DT Glove	American Sign Language	26 letters of the alphabet	3 times	three subjects familiar with the sign language	234 samples
five flex sensors, MEMS accelerometer (ADXL345), and contact sensor	American Sign Language	A-Z letters	10 times		
CybergloveTM	American Sign Language	50 ASL word	12 times	multiple person trained	120 samples
five fabric contact sensors, five flex sensors, and 3D accelerometer	American Sign Language	A to Z and "THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG" statement	5 times	seven subjects, including six hearing and speech-impaired high school students and teachers	
Cyberglove	American Sign Language	74 distinct sentences from 107-sign vocabulary	2–6 times	eight signers	2393 sentences and 10,852 sign instances
two CyberGloves	Arabic Sign Language	100 two-handed signs	20 times	adult volunteer from the deaf community	2000 samples
DG5-VHand data gloves	Arabic Sign Language	40 sentences using an 80-word lexicon	10 times	24-year-old right-handed female	800 samples
flex and contact sensors	Australian Sign Language	120 static gestures	100 times		3600 samples.
flex sensors with 9-axis IMU sensor	Chinese Sign Language	Chinese phonetic alphabet including a, b, c, zh, and ch	30 times	two different individuals	150 samples
three-axis accelerometer (ACC) and multichannel electromyography (EMG)	Chinese Sign Language	72 signs	12 times	Two subjects: male (age 27) and female (age 25)	

Device/Components	Language	Gesture	Samples per Gesture	Gesture Performer	Sample Size
9-axis accelerometer	English Alphabet	26 English alphabet	one time	one person	26 samples
Hall Effect sensor and accelerometer (ADXL-535).	English Numbers	English Numbers 0–9	20 times		200 samples
3-axis accelerometers (ACC) and electromyogram (EMG)	German Sign Language	seven words	10 times	eight subjects (6 males and 2 females, aged 27 to 41)	560 samples
EMG and 3-D Accelerometer	Greek Sign Language	60-word lexicon	10 times	three native signers	1800 samples
Three-flex sensors and three axes accelerometer	Indian Sign Language	four words namely HELLO, YES, SORRY, and PLEASE			
flex sensors and accelerometer	Indian Sign Language	eight commonly used words			
five flexure sensors and three accelerometers	Malay Sign Language	25 Bahasa Isyarat Malaysia (BIM) sign words are used	20 times	only one signer is used for creating signer	500 samples
10 tilt sensors and 3-axis accelerometer	Malaysian Sign Language	A, B, and C. 1, 2, and 3 ‘Saya’, ‘Makan’, and ‘Apa’.	10 times	three individuals	270 samples
five flex sensors and 3-axis accelerometer	Pakistani Sign Language	10 static gestures		(15 females and 15 males) who varied from13 to 45 years old	
5DT Data Glove	Spanish Alphabet	six movements	10 times		60 cases and 37 attributes
10 flex sensors attached to each finger and three-axis accelerometer	Taiwanese Sign Language	five words, namely, Lonely, Promote, Assist, Love, and Protect	each with 50 tests	five subjects	1250 tests
10 flex sensors and one accelerometer ADXL345	Vietnamese Sign Language	29 letters	50 tested for each letter		1450 samples
five ADXL202 accelerometers	Vietnamese Sign Language	23 Vietnamese-based letters with two postures for “space” and “punctuation	40 times	five different persons	200 samples

TABLE 2.1: Table of Data-Gloves for Sign Language Recognition, from [20]

Chapter 3

Talking Hands Platform

Data-Gloves' hardware in the last decades started to be built around specific sensors and architectures. For example a Data-Glove which uses flex sensors to detect fingers' flexion is a cliche just as a Data-Glove which uses an IMU to detect hand orientation can be considered a cliche. Since sensor's industry started to produce really cheap and reliable commercial IMUs, those sensors are becoming more and more popular. From Section 3.3.1 to Section 3.4 we will explore the most used sensors to detect fingers' flexion and orientation of a rigid body. This will help the reader to understand why some hardware and architectural solution have been used in the Talking Hands system. It is important to underline that during thesis workflow various components were added, for example in former prototypes an haptic motor was not integrated in the hardware architecture. From the current prototype a haptic motor with relevant driver have been included inside Talking Hands' hardware architecture and will be present also in future hardware iterations.
We can divide Talking Hands hardware mainly in two versions.

- Architecture used most of time during thesis lifetime, which is described in Section 3.2. It is recognizable because it integrates as sensors two IMUs and ten flex sensors.

- The one described in Chapter 5 where hardware architecture has been totally revised and it will integrates just IMUs as sensors.

First of all lets have a general view of Talking Hands' requirements.

3.1 Requirements

This section explains which are the requirements for our system. Talking Hands needs to keep track of mainly two things:

1. Orientation of user's hand and forearm;

2. Flexion of user's fingers.

User's shoulder is not relevant to recognize a gesture. System will be too much invasive. Starting from those basic facts we can now define more specific requirements.

Requirement specification for an embedded object/system is not so trivial, especially when several scientific/engineering sectors are involved, e.g. computer science, mathematics and electronic engineering. In this section we will show the overall requirements independently merging all macro areas involved, e.g: design, firmware, hardware etc.

Requirement Name	**Requirement Description**
Talking	The device must have speakers to talk on behalf of the user
Real Time	The device must translate in real time gestures into sounds corresponding to a word, letter or a phrase
Fingers' Flexion	The system must recognize and compute the flexion of each finger (proximal and intermediate phalanges).
Hand Orientation	The system must recognize and compute the orientation of the hand in a 3D space
Arm Orientation	The system must recognize and compute the orientation of the arm in a 3D space
Wearable	The device needs to be wearable. All the HW components need to be mounted on a device that can be worn by the user.
User's Tact	The device must not influence the user's tact. It is necessary to create a device that will stand on top of the hand, leaving the bottom free as the user must be able to directly touch things.
User's Movement Freedom	The device should not influence user's movements. The user should be totally free to move his/her hand and arm.
Modularity	The device must be composed of two parts, connected by a coupling/uncoupling system. Hand Module and Arm Module.
Record Gestures	The user should be able to register a gesture.
Association to Gestures	The user must be able to associate to a gesture a letter, a word, an entire phrase or a sound
Haptic and Visual Feedback	The user must receive a haptic and/or visual feedback, because some user could be heard impaired.

3.2 Talking Hands Architecture

Talking Hands system does not use a commercial data-glove. A Data-Glove has been internally developed. Our Data-Glove is realized using a simple architecture to obtain a low cost device which can be accessible to everybody, aiming for less than 1.000$ per piece.

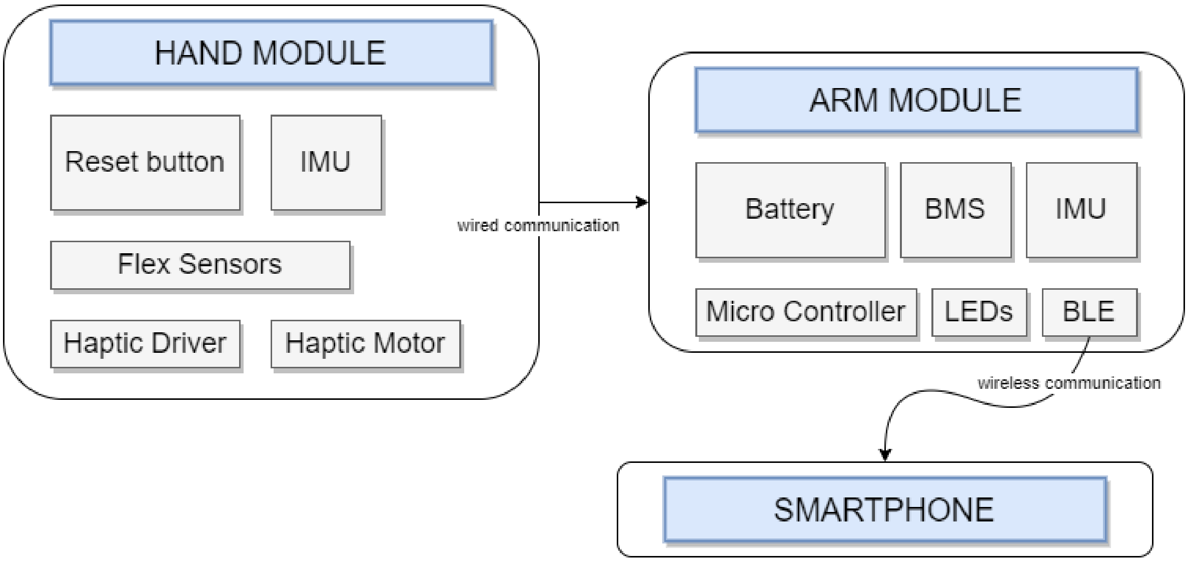


FIGURE 3.1: Simple Architecture of the System

The hardware can be divided into three main modules:

- Hand Module: a small custom Printed Circuit Board (PCB) has been developed. It integrates one IMU (Inertial Measurement Unit) to detect hand orientation; one button to initialize the system and one haptic motor with relevant driver to inform user about system's status. The rigid PCB described before has been hybridized with a flexible PCB. Thanks to the hybridization process, it is possible to integrate also 10 different flex sensors tracking proximal and intermediate phalanges (1 flex sensor for each phalange) in the hand module. The reader is addressed to Figure 3.19 for a better understanding.

- Arm Module: one custom board which includes a 32-bit microprocessor; one IMU to detect forearm orientation; two RGB (Red-Green-Blue) LEDs to check the system status; a Battery Management System, including a mini-usb connection to recharge the battery; a Bluetooth Low Energy (BLE) module which contains also a 32-bit Cortex M0 micro controller to manage BLE protocol and connections between BLE devices. Moreover Arm Module can be used standalone for gesture recognition purposes involving just arm orientation as done in experiment reported in B.

- User's smartphone.

Micro Controller Unit (MCU) communicates with different peripherals: IMUs, Bluetooth Low Energy module, LEDs, Haptic Driver, flex sensors and the reset button. In Figure 3.2 is possible to see a summary of how peripherals communicate with the microcontroller. LEDs, flex sensors and reset button are treated simply as generic I/O, communicating with relevant microcontroller's digital or analog ports, IMUs and Haptic Driver are connected to the same I^2C (Inter-Integrated Circuit) bus, each peripheral has a different address on the I^2C bus, the BLE module communicates with the MCU using a 1152008N1 UART (Universal Asynchronous Receiver-Transmitter). BLE communicates with smartphones opening a BLE connection as wireless UART. BLEs are possible to be configured in different ways thanks to GATT Profiles, GATT stands for Generic ATTribute Profile, it governs data organization and data exchanges between connected devices. One device (the peripheral) acts as a GATT Server, which stores data in Attribute records, and the second device in the connection (the central) acts as a GATT Client, requesting data from the server whenever necessary. For more information about BLE and GATT the reader is addressed to [66].

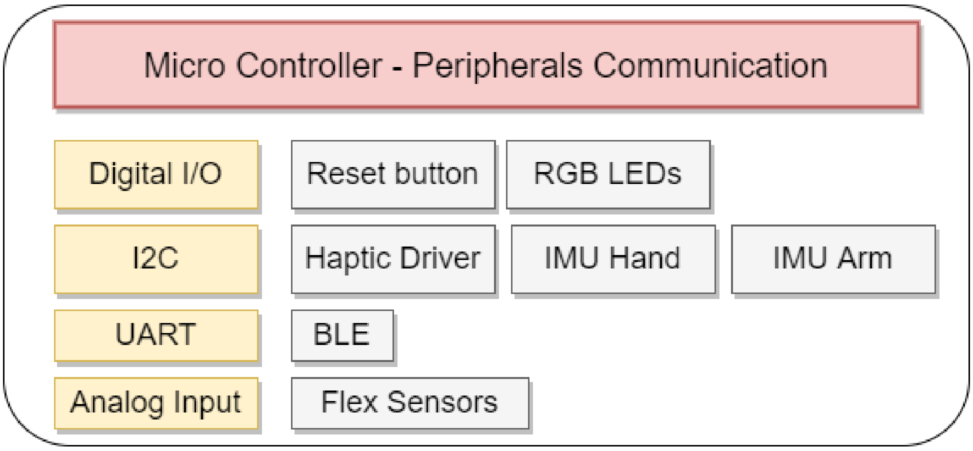


FIGURE 3.2: Type of Communication (yellow) between MCU and peripherals (grey)

The initialization of the system through a button is required to set initial orientation of the user at the origin, more precisely to set the initial data orientation of the IMUsto point in the 3D space $x, y, z = 0, 0, 0$. For a better understanding of this process the reader is addressed to Section 3.8.1. This procedure from now on will be called: Reset Procedure. Once performed the reset procedure, the user can perform gestures and the system will correctly translate the signs. In Figure 3.1 more details about architectural components are presented:

- A Cortex M3 Atmel SAM3X8E is used as micro controller;
- 10 resistive flex sensors has been used;
- Two BNO055 IMUs are used to collect orientation data;
- A Bluetooth Low energy module is used to communicate with smartphones using our communication protocol.

- A 3.7V 1100mAh Li-Ion Battery is used, which ensures an effective duration of 5.94 hours.

A solution with 10 flex sensors has been developed to acquire data of finger's flexion. Two sensors are used on each finger, having information about intermediate and proximal phalanges joints. The distal phalanges joints are ignored because they are not relevant for gesture recognition. This solution can be improved using a resistive flex sensor which have 2 sensible parts. This will ensure less probability of breaking sensor's solder pins after intensive usage. Those pins soldered onto Flexible PCB, after an intensive usage may generate problems decreasing system's reliability, those problems will be taken into account in Chapter 5.

In some studies (e.g.[67]) the position of the fingers are detected using EMG sensors. Even if this can be an elegant solution with only one sensor, it can not recognize the positions of single fingers, so it is not enough if we want to discriminate a huge number of different gestures, for example more than 30 gestures.

Two 9 DOF IMUs from BOSCH are used to obtain information about the forearm and hand orientation. These IMUs require an initial calibration, which it lasts about 30 seconds, but they achieve a high-level of reliability even for prolonged use, thanks to the built-in functions of sensor fusion. This process will be removed thanks to new sensor fusion algorithm used in BNO080. It implements an auto-calibration mode and in Talking Hands future platform there will be no need of calibrating IMU's sensors.

The bracelet (arm part) is the "brain" of the system, inside there is the micro controller, which manages glove's sensors, the Bluetooth module managing the connection to the personal device and the battery that powers the whole system. The exploded diagram alongside shows all the components present in the bracelet. A switch has been used to turn on/off the whole system.

In following sections 3.3.1, 3.3.2, 3.3.3 and 3.4 we are going to describe different possible sensors that could be mounted on Talking Handswith explaining why a solution (sensor) has be choose instead of others.

3.3 Flex Sensors

3.3.1 Resistive flex sensors

Resistive flex are widely use to measure bending angle of a finger in applications which involves Data-Gloves; data about flex sensors are not difficult to be interpreted because they work just as a resistive potentiometer. Flex sensors are comfortable to wear because they are really light, thin and with a low power consumption.

Flex sensors are not used only as finger tracker, they can be used to track other part of the body, for example knees, they can used also in industrial application, or as simple switches.

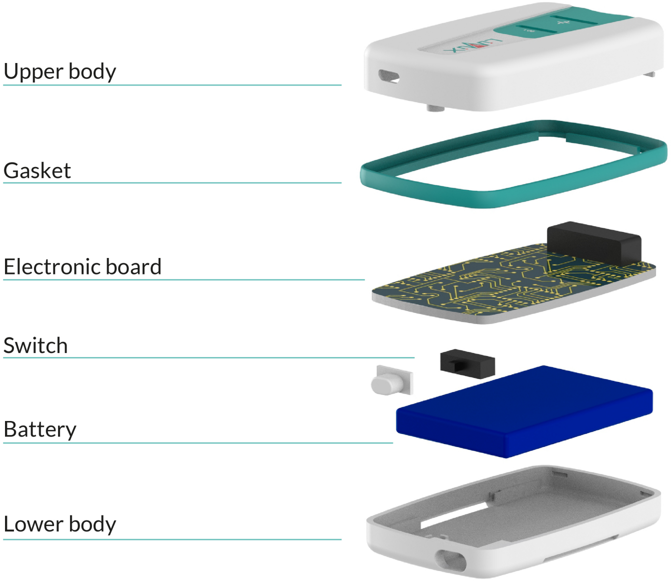


FIGURE 3.3: Exploded Diagram of Talking Hands's bracelet

In this thesis resistive flex sensors are mainly considered for the measurement of fingers' flexion.

Among flex sensors, a special role is played by the passive resistive ones. They are made of electrically conductive patterns, engineered on top of or within a flexible substrate (schematized in Figure 3.4) that is able to perceive bending without concerning electromagnetic interference or sensor occlusion [68].

Most of the time flex sensors are used to measure the flexion of a material or an object. Imagine to fix a rubber band on a finger, from the nail to the back of the hand: when the finger is bent, the rubber elastic extends. This allow to measure the finger's degree of extension. In our application they are used to measure finger's flexion, more specifically flexion between: phalanges and metacarpals bones; between proximal and intermediate phalanges.

For example in Talking Hands first prototype, illustrated in Figure 3.15 ten flex sensors from Spectrasymbol have been used, each one long 55.37 mm. Data of each flex sensor is an input for the main microcontroller, it has been treated as a variable resistance, using a simple voltage driver configuration.

After intensive testing, is possible to say that resistive flex sensors are good to understand fingers' flexion, but in a Data-Glove application repeatability on measure is no longer guaranteed because the user will not always wear in the same way the Data-Glove, flex sensors may be translated upward or downward the finger surface. This translation influences a correct reading of finger's flexion.

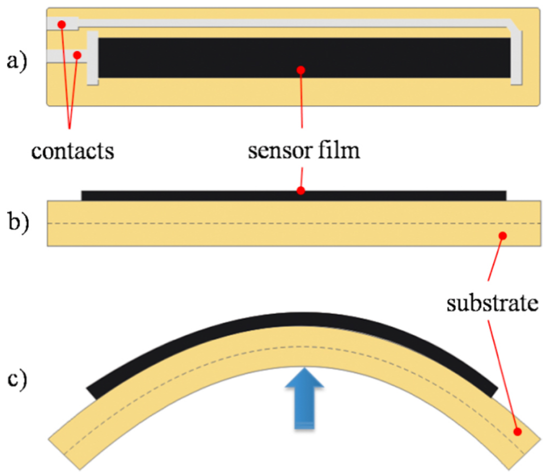


FIGURE 3.4: Scheme of a resistive flex sensor. (a) Top view: electrical contacts in grey, conductive film in black. (b) Lateral view: conductive film, in black, on top of a substrate, in a lighter color. (c) Bending the substrate causes a mechanical stress of the conductive pattern that leads to a change in its electrical resistance from [68]

3.3.2 Capacitive flex sensors

Silicone-capacitive flex sensorsare highly accurate and reliable.

They are usually applied for sensing technology in medical, robotics and soft industrial industries. The sensor responds to changes in geometry, and communicates to a 5-channel circuit to display stretch information.

As illustrated in Figure 3.7, these sensors work with differences on capacitance. The sensor recognizes a maximal extension of 79.9% of its nominal length ($78.00mm$). A wider extension will be ignored. This fact is significant for our purposes. Capacitive flex sensors works on sensing the elongation and not flexion. This is a really important fact to consider. For example they will suffer much less from the problem described previously in Section 3.3.1.

One of the first points to consider in this study has been the calculation of the maximum extension for each finger in the interested areas. A test of the sensor mounted on the fingers has been performed and the results are reported on the following table:

Finger	Max Extension[mm]
Thumb	25.16
Index	30.24
Middle	25.52
Ring	29.51
Little	32.11

This is enough to infer that the extension of these sensors is 50% larger than the one needed for our purposes.

Capacitive flex sensor are really precise and easy to use, more than the resistive one. Capacitive flex sensors have not been used in Talking Hands because of their extremely

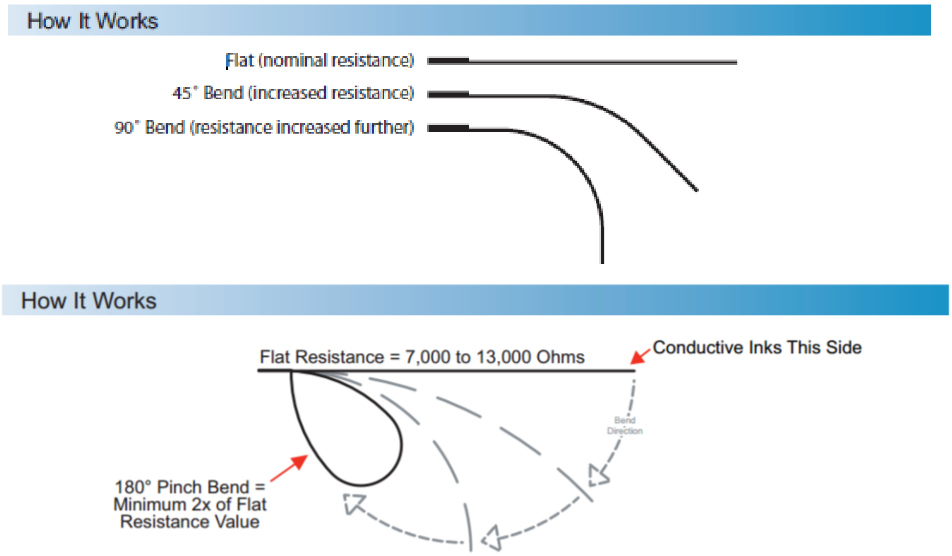

How It Works
Flat (nominal resistance)
45° Bend (increased resistance)
90° Bend (resistance increased further)
How It Works
Flat Resistance = 7,000 to 13,000 Ohms
Conductive Inks This Side
Bend Direction
180° Pinch Bend = Minimum 2x of Flat Resistance Value

Figure 3.5: How a Flex Sensor Works

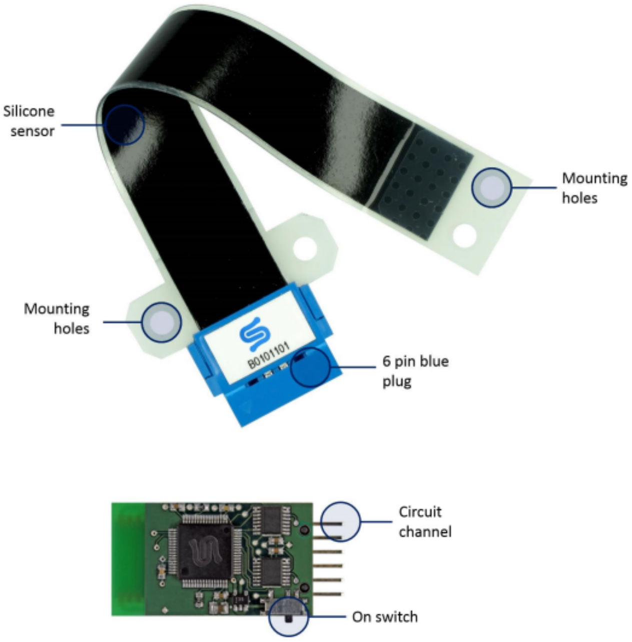
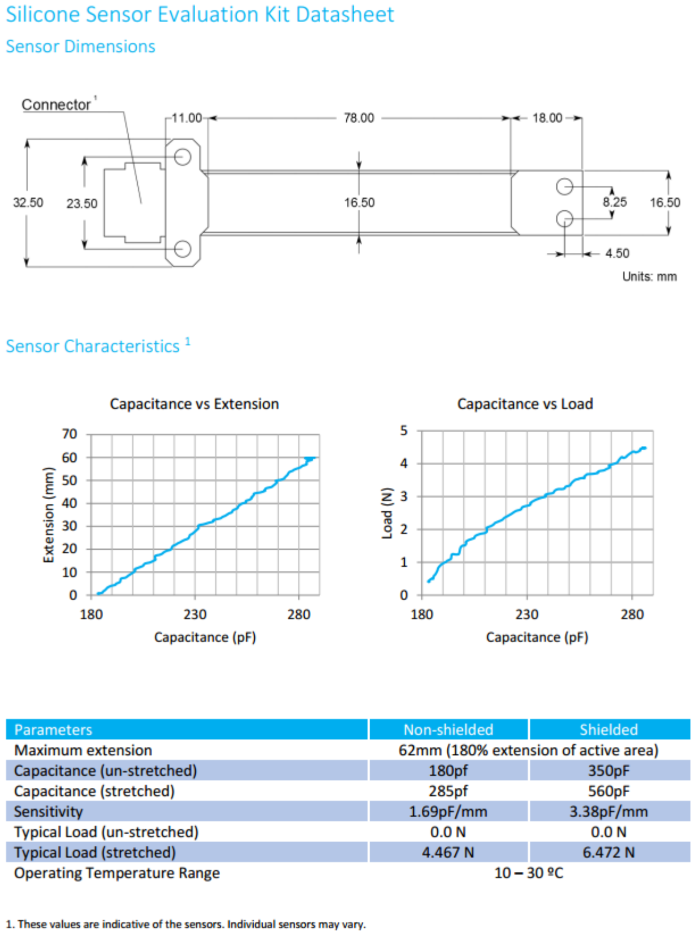


Figure 3.6: Silicon Bend Sensor from Stretchsense and Circuit



Silicone Sensor Evaluation Kit Datasheet

Sensor Dimensions

Connector 1
11.00
78.00
18.00
32.50
23.50
16.50
8.25
16.50
4.50
Units: mm

Sensor Characteristics 1

Capacitance vs Extension
70
60
50
40
30
20
10
0
Extension (mm)
180
230
280
Capacitance (pF)
Capacitance vs Load
5
4
3
2
1
0
Load (N)
180
230
280
Capacitance (pF)

Parameters	Non-shielded	Shielded
Maximum extension	62mm (180% extension of active area)	
Capacitance (un-stretched)	180pf	350pF
Capacitance (stretched)	285pf	560pF
Sensitivity	1.69pF/mm	3.38pF/mm
Typical Load (un-stretched)	0.0 N	0.0 N
Typical Load (stretched)	4.467 N	6.472 N
Operating Temperature Range	10 – 30 ºC	

1. These values are indicative of the sensors. Individual sensors may vary.

Figure 3.7: Silicon Bend Sensor from Stretchsense Charateristics

high cost. Those flex sensors are ideal for our application; costs for integrating this technology are too high also during the production phase respecting our target price. Stretchense itself is selling a glove with inside its capacitive flex sensors, prize for a pair of gloves is about 7.150$. This means 3.575$ per glove which uses just 5 capacitive flex sensors with two sensing zone to measure bending of the first two phalanges. Stretchsense's glove architecture is visible in Figure 3.8.

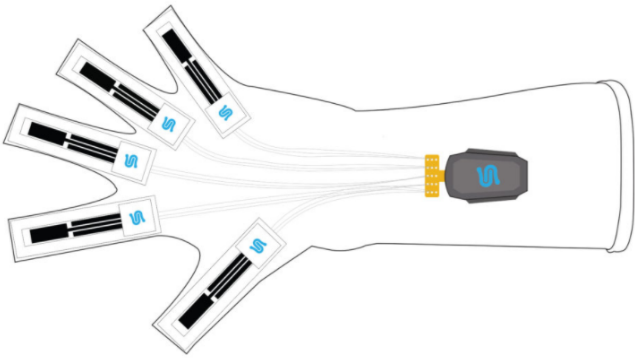


FIGURE 3.8: Stretchsense's MoCap Glove

Moreover capacitive flex sensorsfrom Stretchsense can understand splay movement thanks to configuration of this particular splay sensor. In the diagram a splay sensor is showed with relevant channels: A, B and C. Channels A and C are will help us to determine if splay movement is occurring. Channel B is placed above and between channels A and C to capture bending at the middle knuckle. If index finger moves to the left, channel A decreases in length and channel C increases. Conversely, when the finger moves to the right, channel A increases and channel C decreases. Looking at the difference enables us to determine the finger's lateral position.

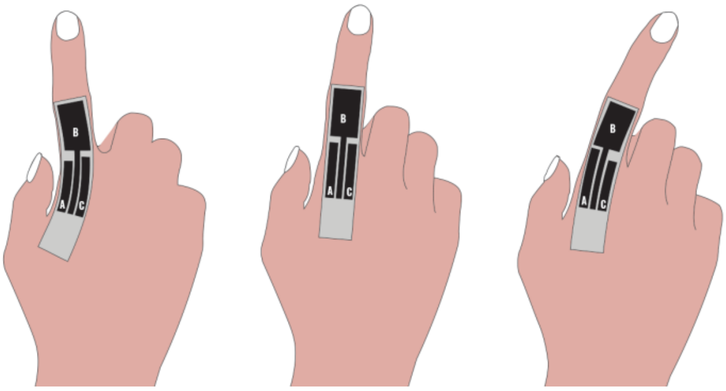


FIGURE 3.9: How Splay Sensor works

3.3.3 Sensors with Stainless Steel Soul

Main article about yarn-based stretchable sensor arrays (YSSAs) is [69]. It is an innovative and original work. This study proposes YSSAs as low cost and reliable

sensors.
What is really interesting in the article is how YSSAs are built and made. In Figure 3.10 is possible to understand sensor's components and relevant materials.
Sensors consist ofby (in order from inside to outside):

- Soul of Staineless Steel,
- Strends of Polyester which surround stainless steel soul,
- A rubber membrane,
- Spiral of Polyester wire,
- Membrane of polydimethylsiloxane (PDSM) sleeve.

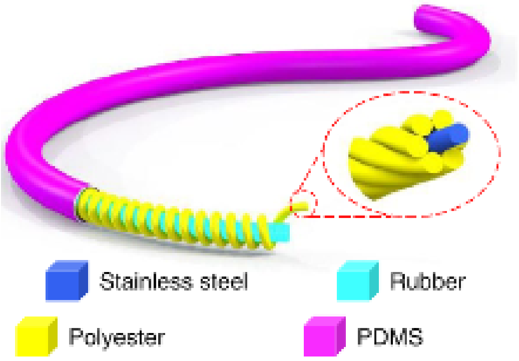


FIGURE 3.10: Yarn-Based Stretchable Sensor Array - Schematic illustration

YASSAs sensors shows also a linear growth considering Stress (deformation) - Voltage relation. As illustrated in Figure 3.11. Experiment is about bending the YASSA sensor 1 time per second. Authors also declare a stretch–release response time of less than 15 milliseconds. In a gesture recognition application, this technology will be more than enough to recognize gestures.

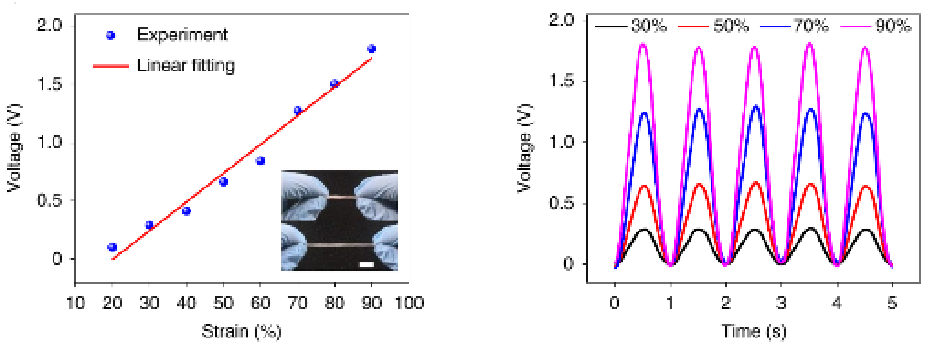


Figure 3.11: (Left) Linear fitting of stress-voltage - (Right) Generated voltage signals of the YASSA sensor

3.4 Inertial Measurement Units

During all the duration of this thesis a lot of focus was about IMUs understanding, choosing the best IMU believing that is the best on the consumer market. This project mainly aims to build a real product so a commercial IMU needs to be used. Highly cost IMUs (>30 USD) will determinate a high cost for Talking Handsto be on the market. The target is a product under the price of 1000 USD.
In the current version of Talking Hands two IMUs are used. In future versions eleven IMUs BNO080are expected to be used. High cost on sensors means do not obtain a relatively low cost product in production phase.

In this section IMU technology is explained for a better understanding about why we choose this technology as main actor for this system.

An inertial measurement unit IMU is an electronic device that measures and reports a body's specific force, angular rate, and sometimes the magnetic field surrounding the body, using a combination of accelerometers and gyroscopes, sometimes magnetometers. IMUs are typically used to manoeuvrer aircraft, including unmanned aerial vehicles (UAVs) and spacecraft, like satellites and landers. [70–73].

3.4.1 9 or 6 Degree of FreedomIMUs

Typically a critical point is choosing the right IMUmodule for the particular application and which IMU is right for your application. Six Degree of Freedom IMU does not use the magnetometer and this might be an advantage since magnetometers are highly sensible to the environment, i.e. the measures can be corrupted by the presence of external magnetic fields or due to the closeness to magnetic-iron materials.

However, a fundamental question arises: is the 6 Degree of Freedom IMU suitable for our purposes? Clearly, a partial loss of information occurs, and hence the key point is establishing whether acceleration and angular velocity are sufficient to recover the orientation or not. While from a theoretical point of view the answer is positive, in practice the outcome is not sure.

With a 6 Degree of Freedom IMU drift starts to be huge after 2-3 minutes. A rude way to solve the problem is resetting the IMU every time the drift starts to be relevant, putting the dataglove in the known initial position then resetting the IMUs. That is no acceptable for our application and we decided to integrate the magnetometer in the IMUs, using two 9 Degree of Freedom IMUs.

Lets now introduce the problem called Gyro drift: not going back to zero-rate value when rotation stops. Gyro drift occurs in inertial navigation systems (INS, in area navigation/RNAV). An INS system consists of the inertial platform, interior accelerometers and a computer.

There are two sorts of gyro drift:

- **Leveling Gyro Drift**: the random rotation of the gyroscope around the axis will tend to shift the platform away from the horizontal causing an oscillation action, which the accelerometers try to correct. This oscillation, depending on its period, will cause velocity errors. Velocity errors will be reflected on position errors.

- **Azimuth Gyro Drift**: Azimuth gyroscope drift is caused from small position errors. However, gyro drift about the azimuth axis (pitch, lateral axis) produces small errors compared to the initial misalignment errors in azimuth.

To correct the Azimuth and Leveling gyro drift more information are needed. A magnetometer can be used to correct it.
Merging all those information in a sensor fusion algorithm e.g. AHRS (Attitude Heading Reference System) algorithm is possible to have a more reliable system.

The best IMU solution for us in terms of money, precision, reliability and repeatability of the measures are BNO080 or BNO055 from Bosch Sensortech.

Before arriving to IMU solution, different approaches have been tried. First of all we tried to understand if it was possible to use accelerometers to track hand position, integrating acceleration two times, obtaining position. We previously clarified why is not possible to use a commercial accelerometer to obtain position and velocity and a more detailed explanation is available in [47]. After that, we also tried to use accelerometers as finger trackers, replacing flex sensors. We found that it is also not possible, we tried in an empirical way proposing an experiment to clarify this concept, the reader is addressed to Appendix A.

3.4.2 BNO055

BNO055 is a System in Package (SiP), integrating a triaxial 14bit accelerometer, a triaxial 16bit gyroscope with a range of ±2000 deg per second, a triaxial geomagnetic sensor and a 32bit microcontroller running the company's BSX3.0 FusionLib software. With a size of 5.2 x 3.8 x 1.1 mm^3, it is significantly smaller for our purposes.

Bosch declares that BNO055 is particularly suitable in applications like:

- Augmented reality
- Navigation
- Gaming
- Fitness and well-being
- Context awareness

It is worth to note that the role of IMUin those application is very close to our case, and this is a further confirmation of a good choice.
BNO080is really similar to BNO055, reader is addressed to download BNO080's datasheet for further and detailed information.

3.5 Processing Units

A Microcontroller or MCU (Micro Controller Unit) is a small electronic component which is the brain of nowadays embedded systems.
In Data-Glove based applications microncontrollers are used as main component to collect data from sensors, running a gesture recognition algorithm or a part of it. Today in Data-Gloves we usually find more microncontrollers, especially thanks to the huge growth of System On a Chip (SoC) technology. A SoC is a system were the microncontroller may be just a part of it, in the same integrated circuit we found also more peripherals, every component of the SoC is produced on the same manufacturing process. A bit different from SoCs, are Sistem in Packages (SiPs). The difference between SiPs and SoCs is the manufacturing process; indeed SiPs do not store components in one integrated circuit (substrate), but has multiple chips inside the same package, a SoC instead integrates all the needed peripherals just in one substrate. SiP's Dies (a small block of semiconducting material) contains integrated circuits (ICs) which may be stacked vertically on a substrate as in Figure 3.12. They are internally connected by fine wires that are bonded to the package; those ICs may also be connected to passive components (as resistors or capacitors) and mounted on the same package. This means that a complete functional unit can be built in a multi-chip package, and few external components need to be added to make it work. For a better understanding the reader can refer to [74].

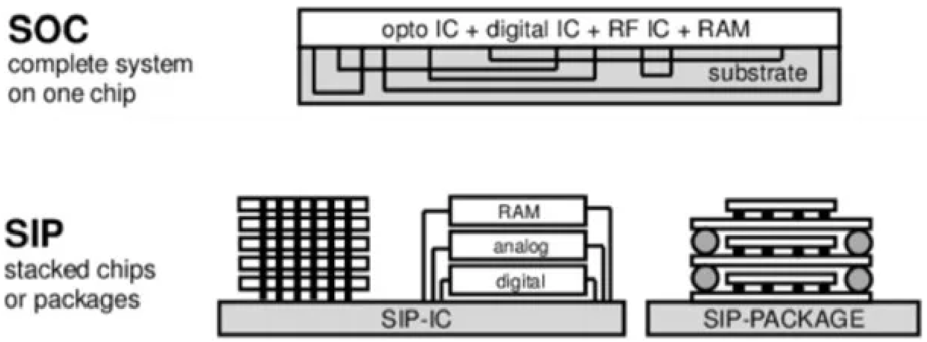

SOC
complete system on one chip
opto IC + digital IC + RF IC + RAM
substrate
SIP
stacked chips or packages
RAM
analog
digital
SIP-IC
SIP-PACKAGE

FIGURE 3.12: Comparison among SOC (System-On-Chip), and SIP (System-In-Package) from [74]

For example in our Data-Glove application we have a SiP component: the IMU (BNO055). BNO055 is a system in package because it integrates a triaxial 14-bit accelerometer, a close-loop triaxial 16-bit gyroscope, a triaxial geomagnetic sensor and a 32-bit microcontroller (ARM Cortex M0) running the BSX3.0 FusionLib software.

The previous introduction about microncontroller, System in Package (SiP) and System On a Chip (SoC) is done for a better understanding of the technology behind Talking Hands' architecture, underlining today's trend to decentralize computation also on an embedded system. IMUs perform all the computation in its own System in Package (SiP) obtaining precise data about orientation with its own microncontroller and its own sensors.

Developers and engineers do not have to face and manage hardware/firmware integration of different kind of sensors (magnetometers, gyroscopes and accelerometers) but can easily relay on filtered data directly from the SiP. It is possible to say that nowadays in a Data-Glove application the main microncontroller is not chosen because of its computational performances. It is chosen in relation of how many different SiP it is able to manage. For example in Chapter 5 the microncontroller will be changed because the old ARM Cortex M3 is no longer able to manage the new architecture with 11 IMUs connected using the I2C interface. It is not possible to connect 11 BNO080 to the same I^2C channel because BNO080 can be configured only with 2 different I^2C addresses (0x4A or 0x4B), so to manage 11 IMUs 6 different I^2C channels are needed and every channel will host maximum 2 IMUs.

Computational power is no longer a problem if the gesture recognition algorithm runs on another device and gesture's data are stored also in another device or an external memory, for example on a smartphone or on a personal computer. In fact lot of researches use just an 8-bit microncontroller as the Atmel AVR ATmega328P of an Arduino board. This microncontroller has 14 digital IO pins, 6 analog inputs and a 16MHz crystal oscillator, not at all comparable to the BNO055's or BNO080's microncontroller.

3.6 Talking Hands Prototypes

One of the biggest challenges during the development phase was the creation of design and mechanics. Talking Hands has been thought to guarantee portability and comfort without compromising user's tactile sensitivity.

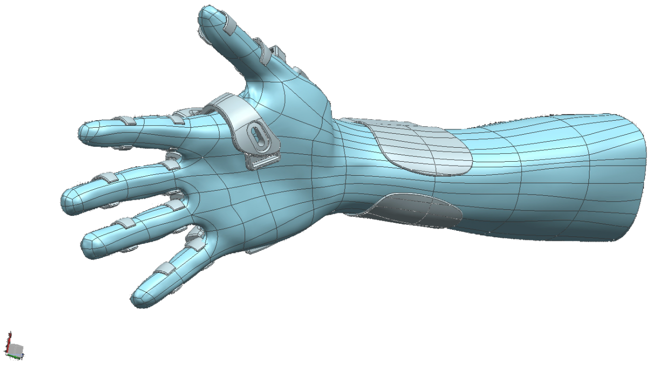


FIGURE 3.13: Palm

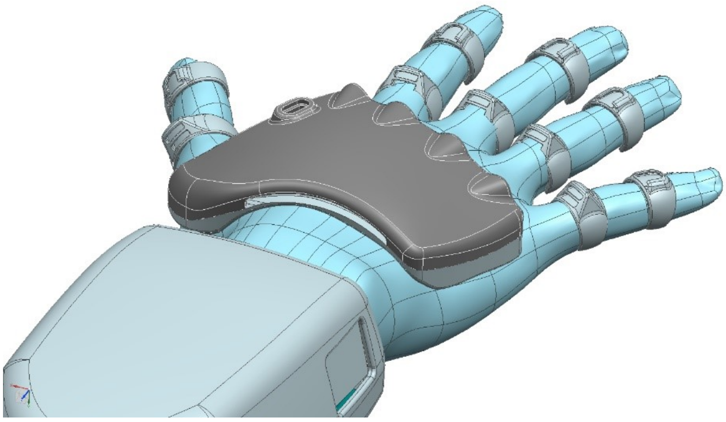


FIGURE 3.14: Back of the Hand

In this section different prototypes will be described, showing many difficulties encountered during the design phase and how those problems have been faced.
The structure of the first prototype was thought to allow all the possible movements to test the entire system in a real situation. It has been our first test bench.
In the first prototype shown in Figures 3.14, 3.13, 3.15 the main objective was to create a structure which allows integration of sensors (IMUs and flex sensors) leaving the user free to move any articulation of his hand and forearm.
Design a slot for forearm's hardware components has been quite easy: a lot of space is available on the forearm and it is possible to consider it as a fully-rigid body. All the hardware components regarding the forearm have been inserted in Part c) of Figure 3.15. Arm part is really bulky because an Arduino Due has been used as prototyping platform, a huge volume has been taken by it.

Housing all hardware components on the hand-part has not been so easy as for the forearm. During design phase of the hand part two major technical problems have been faced:

1 IMU integration in the hand structure: the IMU needs a rigid structure to be mounted on, if the exoskeleton-hand-part slides on the user's hand, microcontroller will receive false variations about hand's orientation. IMU must be integral to the rigid body where it is mounted.

2 Allow flex sensors sliding while finger bends. In the meanwhile flex sensors should not be subject to torsion, otherwise data could be interpreted erroneously as flexion.

About point 1, a rigid structure has been created following the hand shape. Inside Hand Part (visible in Figure 3.15 b) an IMU BNO055 has been placed with all the wiring needed to make flex sensors, IMU and reset button work. About point 2, flex sensors come out from the hand part thanks to rails positioned in proximity of knuckles, after coming out from those rails, flex sensors continue to be driven by tracks on the rings (see Figure 3.15 a).

As for the hardware architecture is possible to summarize the mechanical system in three parts:

- A set of rings for housing the different flex sensors (Figure 3.15 a);
- The back of the hand, including all cable connections and IMU (Figure 3.15 b);
- The arm part connected to the previous via a fabric strap, through which the cables pass; it has the space required to insert the microprocessor, an IMU, the bluetooth communication module and the battery (Figure 3.15 c).

The first complete prototype is visible in Figure 3.16.

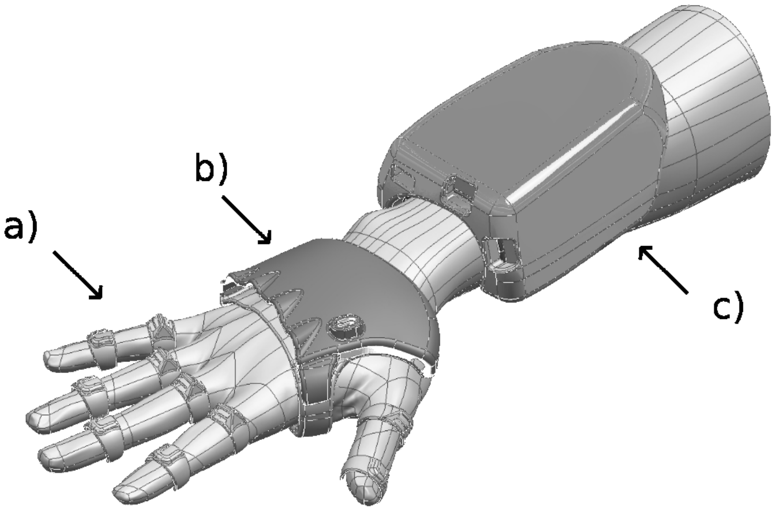


FIGURE 3.15: Cad of Talking Hands. a) System of Rings b) Hand Part c) Arm Part

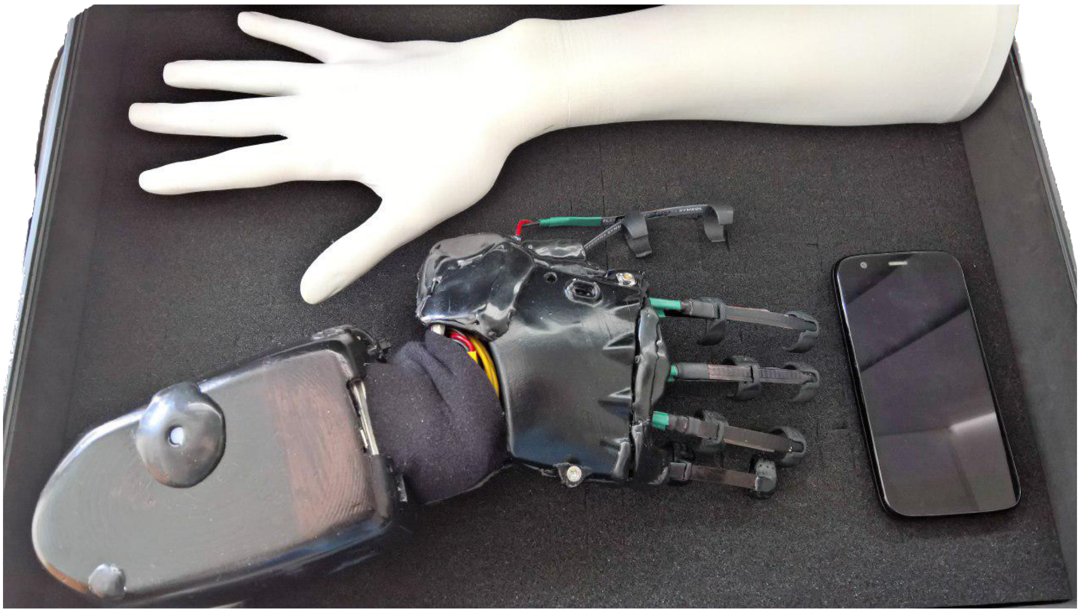


FIGURE 3.16: Real First Prototype of Talking Hands

The second iteration started from the prototype previously described, this new one, maintains functionality of the original one, but it is lightweight and the wearability is also improved. Transparent and grey part visible in Figures 3.17 and 3.18 are 3D printed using a Formlabs Form 2 resin SLA 3D printer. Black part on the hand is 3D printed using an SLS industrial printer; flexible nylon has been used as material.

In Figure 3.18 is possible to see 4 transparent part mounted on the black one. There are guides allowing flex sensors to slide without being jammed while user closes/opens one or more fingers. This part replaces rails described in the previous prototype.

On the bracelet a 1.800mAh battery were mounted on the right side, a custom PCB has been created. Connection between the arm part and the hand part are done using flexible silicone wires which prevent the cables from breaking. If we use standard cables to wire the system, after a brief usage wires which connect flex sensors usually break up. In the previous prototype fractures to wires were really common, using this particular silicon wires this problem is no longer present, also after one year of usage.

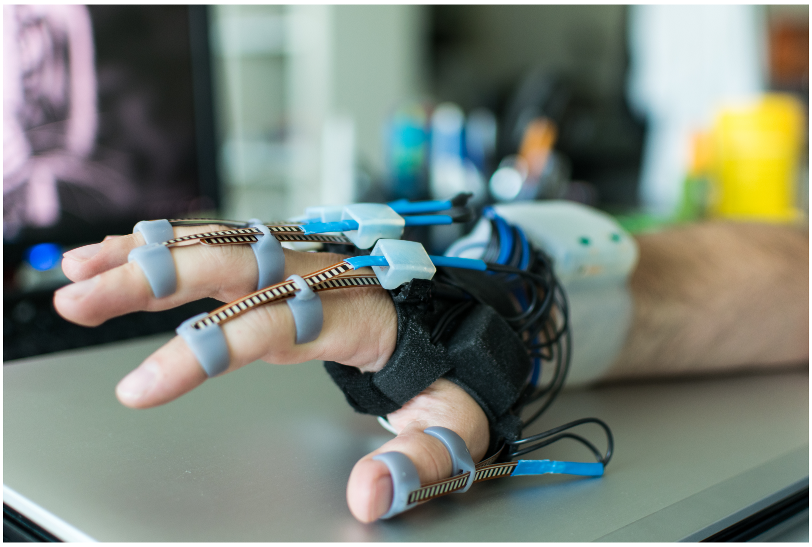


FIGURE 3.17: Lateral view of the second prototype

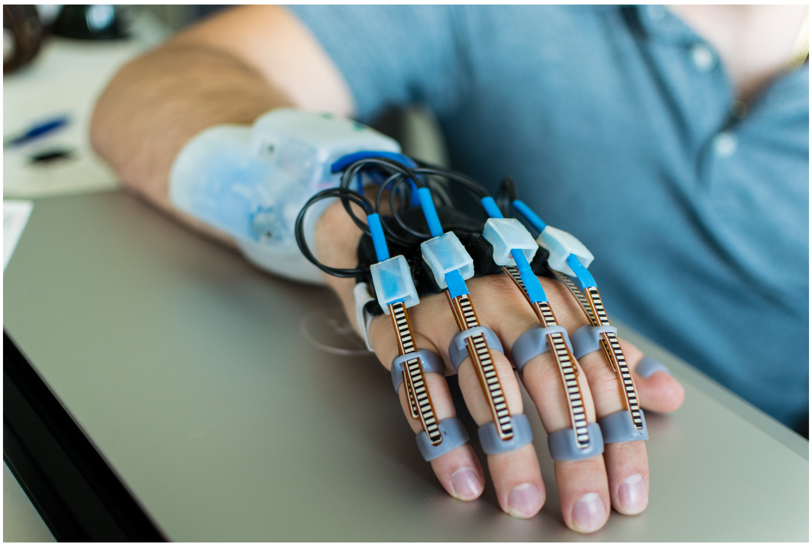


FIGURE 3.18: Top view of the second prototype

Third and last built prototype, shown in Figure 3.19, has new custom PCB on the arm part. A Flexible PCB has been realized trying to remove mechanical system which allows flex sensors to be used without any external accessories.
Flexible PBC is visible in Figure 3.19 on the hand. A haptic motor and relevant driver have been added to the system as described in Section 3.2, with the possibility to use more than 100 different waveforms as haptic feedbacks for Talking Handssystem state. For example if Talking Handscorrectly translated a sign into speech the glove will give as feedback a short vibration. This haptic feedback is very important for hearing impaired users.

The textile glove has been produced to test new hardware without loosing to much time on mechanical development, this solution has been temporary taken for hardware and software evaluation process.
The Flexible PCB is anchored to the textile glove thanks to velcro positioned between the FPCB and the glove. Rails for flex sensorsthis time are made directly inside glove's fingers.

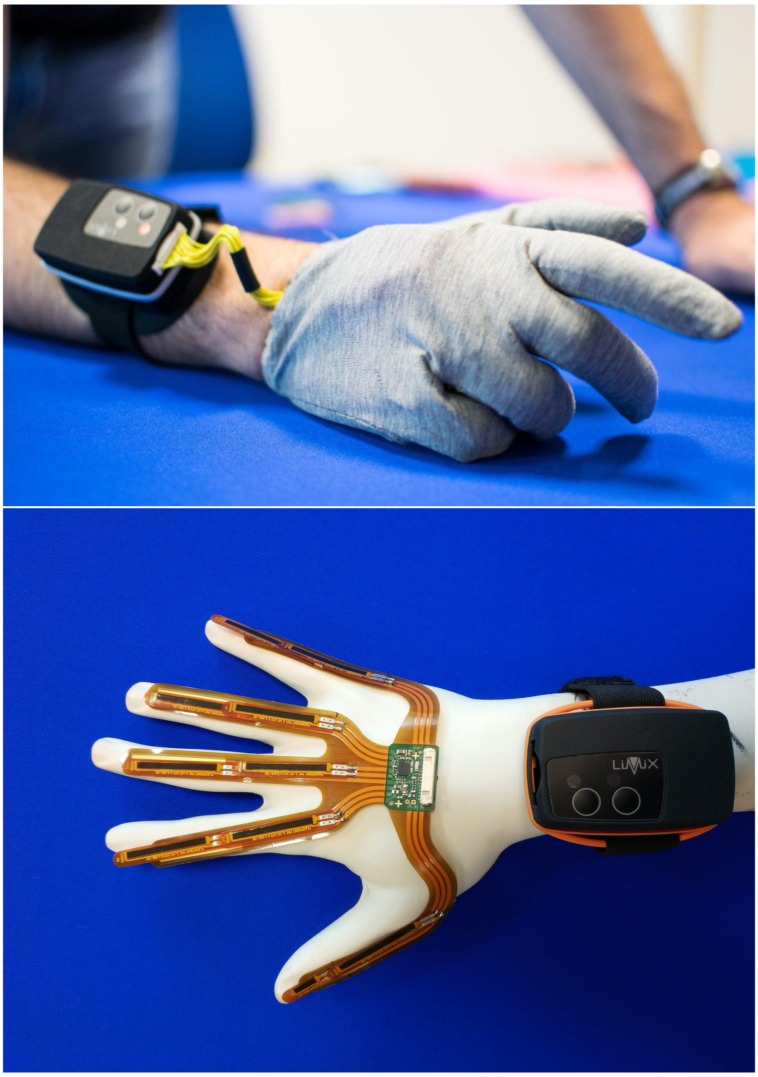


FIGURE 3.19: Current Prototype with Naked Hand Part

Starting from the current prototype, the main objective is to hide the electronic parts as much as possible while keeping the glove structure lightly, in terms of visual impact (user acceptance) and physical dimensions. [1] We will consider different ways and combinations of styles, materials and shapes, to arrive at the final goal. We will analyse the pros and cons of using for example a more organic design, we will consider the use of combinations of rigid and flexible materials, trying to reduce the parts and create a monocoque object, easy to clean and wear.

Naturally, we will take into account relevant negative feedback of the textile glove, which for example limits the use of touch and makes its use uncomfortable in certain periods (e.g. use a textile glove during summer), taking into account also more difficult targets, such as patients with ASD (Autism Spectrum Disorder), considering that people with ASD usually hate to wear gloves, a more lightly and charming device will be preferred especially by youngest people with ASD. The same process will also be done for smartphone's application. We must take into account the diversity of end users, with respect to age, skills and residual skills. Therefore, we will simplify the architecture, designing a more immediate and natural user experience and eliminate all those more technical functions that might lead to confusion the user.

3.6.1 Talking Hands Ergonomics

Until now, we thought about mechanical structure design of Talking Hands just with the aim of test former prototypes' hardware. A solution built with the aim of enhance user experience has not been described. In this subsection we will take care also about Talking Hands' ergonomics and user experience.

Ergonomics is the science that deals with the interaction between elements of a system (e.g. humans, environment, technological components) and the function for which they are used, in order to improve the satisfaction and overall performance of the system. In practice, it is that science which deals with the study of interaction between people and technologies. As a multidisciplinary science, in fact, it finds application in three main areas:

- Physics,
- Cognitive,
- Organizational.

Starting from the current prototype, we tried to find the right compromise between functionality and form, taking into account the dimensional and morphological constraint given by the hardware. Main goal of the design phase is to lightening the structure, without prevent the user's touch (sense). A first sketch has been developed using organic geometry, simply adapting device's shape to the hand shape, this sketch is visible in Figure 3.20.

[1] This study has been conducted with master student Michele Di Carlo, Industrial Designer.

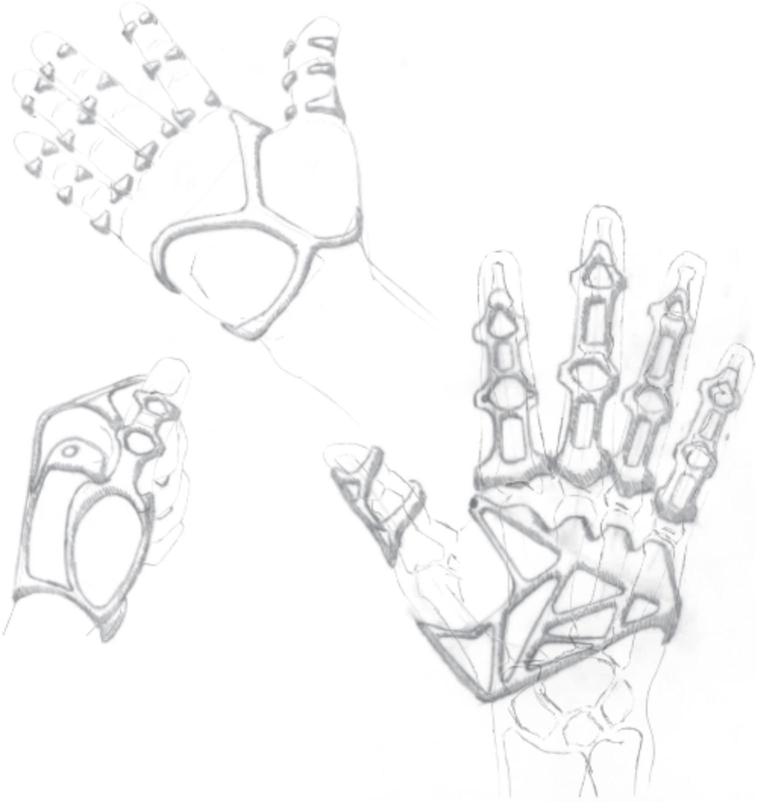


Figure 3.20: First Sketch - Organic Geometry

The sketch developed using organic geometry has been a good first iteration, especially because a huge quantity of material has been removed, structure is lighter, but it still have hardware exposed in plain sight. The same problem will remain in the sketch drawn during the second iteration (see Figure 3.21) with the combination of rigid materials (rings and back of hand) and flexible materials (connections between rings and back of hand). The idea was to use auxetic geometries managing different hands' sizes, making the structure softer where fingers bend.

With respect to the current textile prototype, we began to work on a more minimal shape that covers flex sensors, leaving much of the palm area free.We have therefore followed this path trying to free the area of the fingertips as well. Regarding materials, a combination of silicone and textile materials can be used to insert the flex sensors (green parts in Figure 3.22).

Thus we arrived at the development of the first real concept, which featured a minimal design shown in Figure 3.22.

Although the concept was intended to solve the aesthetic problem and the problem related to the poor breathability of the fabric glove, it did not solve some key points relating to the various user groups: Product limits use of the fingertips, wrapping almost the entire surface of the fingers. Such an impactful wearable object could cause a refusal for a more complex user such as people with autism spectrum disorders, for the majority of people with ASD, it would be unthinkable to put their hand into a similar glove.

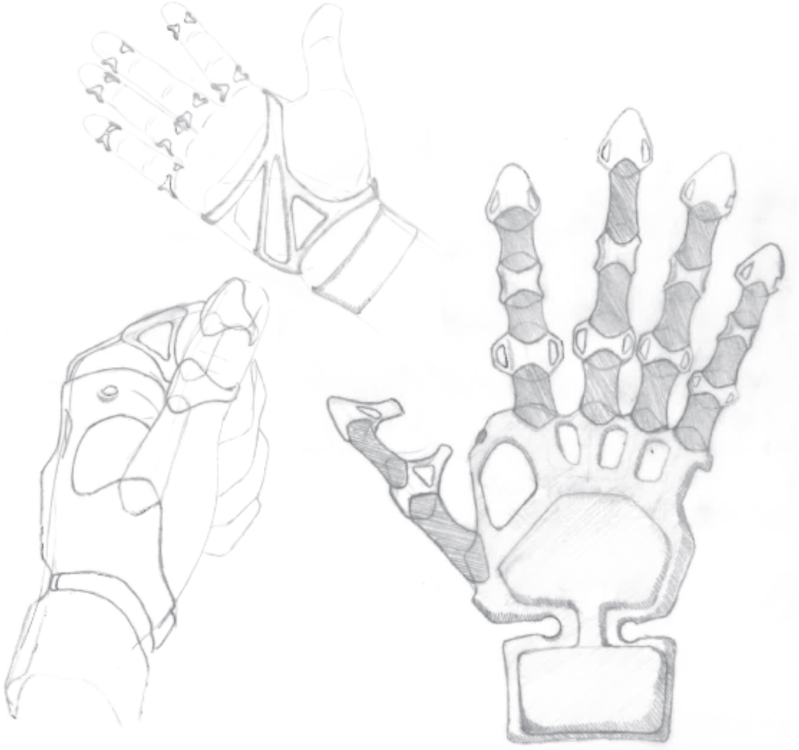


FIGURE 3.21: Second Sketch - Auxetic Geometry

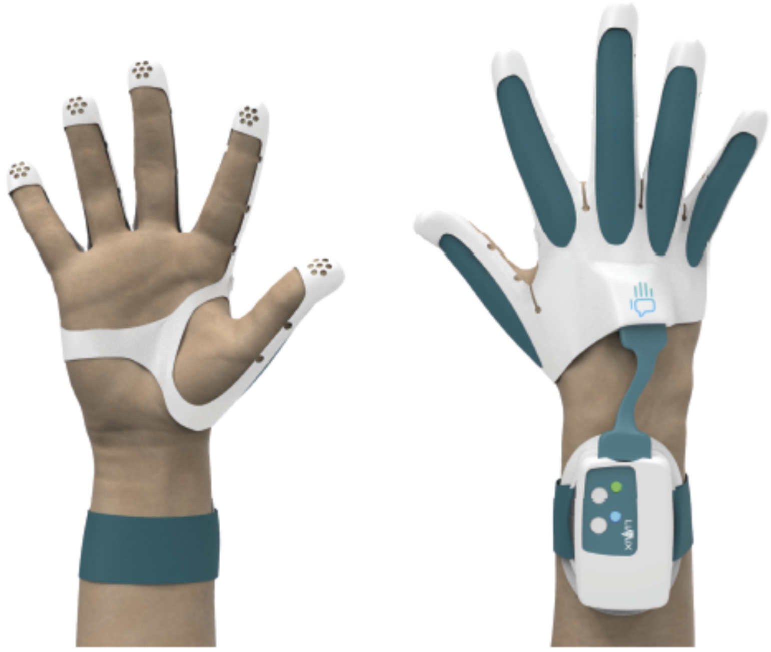


FIGURE 3.22: First Render, using flex sensors

3.7 Smartphone Application

In this section we will describe the Talking Hands' smartphone application.

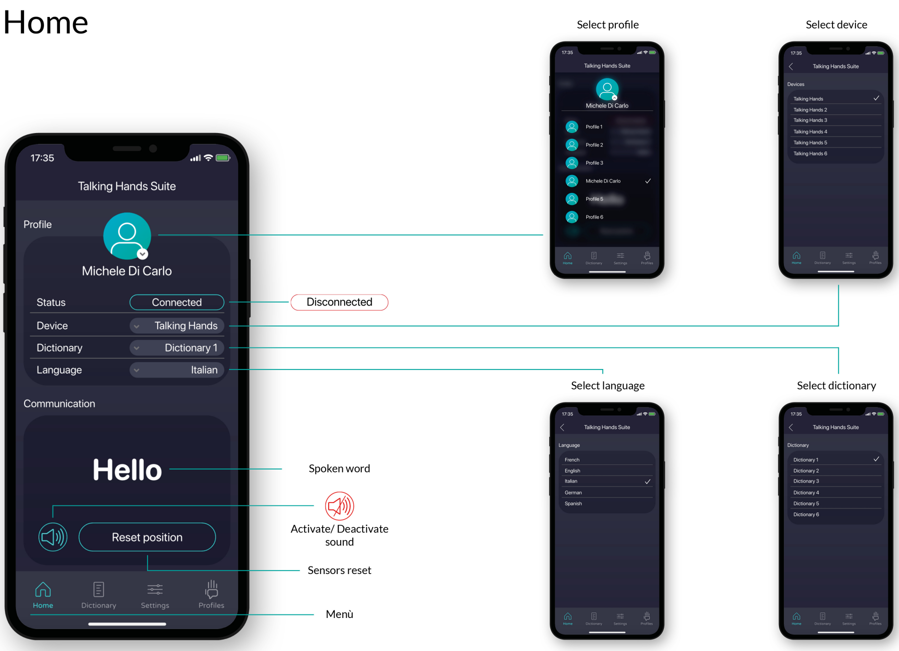


FIGURE 3.23: Home screen Architecture

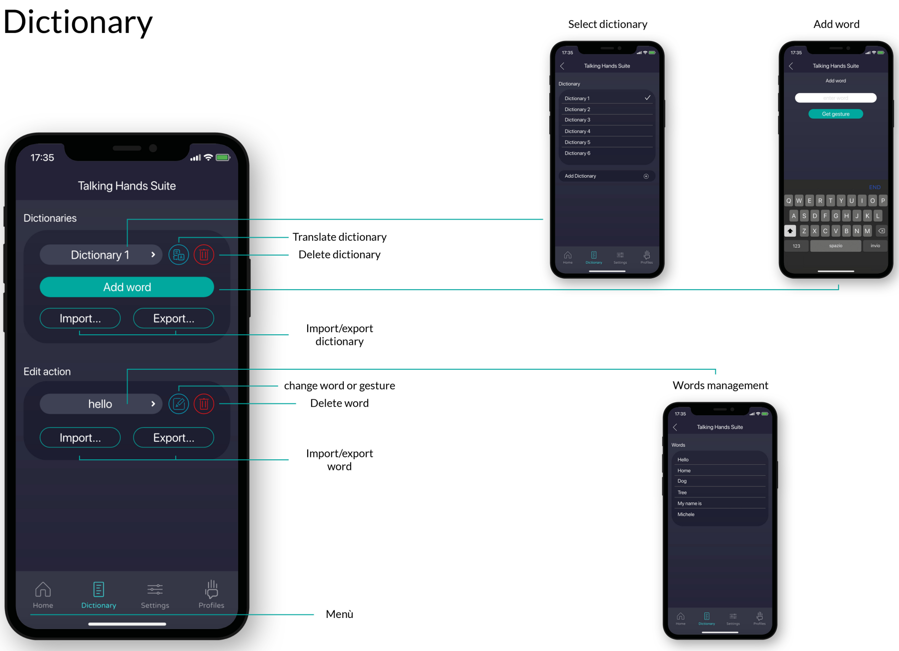


FIGURE 3.24: Dictionary screen Architecture

In Figure 3.24 is possible to see how application's architecture has been developed. The user in home screen have the possibility to

- connect/disconnect his own device;
- choose dictionary he/she wants to use;
- Mute/unmute the system;
- Reset initial orientation (see Section 3.8.1);
- Select spoken language;

This application has been created also to be used by a speech therapist. The doctor can also select different patients using the "select profile" button. This feature can also be useful for a family which has one or more people that needs to use Talking Handsor for an hospital/clinic.
When the user select "dictionary" he is redirected to another main page of the system; in this environment the user can record gesture associating to each gesture a number, letter, an entire phrase or a word creating his own dictionary of gestures. He can also import/export dictionaries, change meaning to a gesture or change the word associated to a gesture without delete the word. It is similar to the rename function we have on PCs, we can rename the word associated to a gesture and moreover we can substitute the gesture associated to a word just clicking on relevant buttons. See Figure 3.24. Users have also the possibility to import/export dictionaries, in this way users can share dictionaries. This will be really useful for a speech therapist: if he needs to do the same exercises with more than one patient, he can import an already existing dictionary made by another person. This feature will be also meaningful in case the user want to change smartphone or operative system. The entire database of gestures will be preserved.

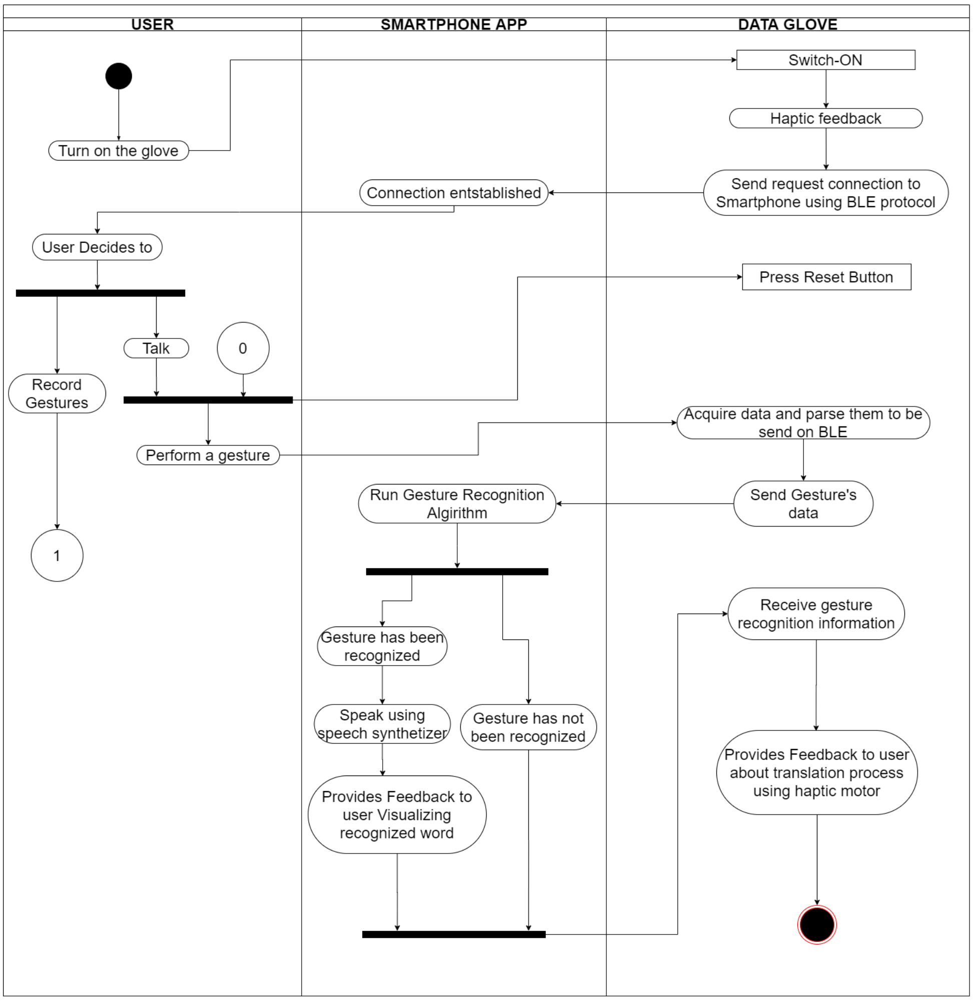
3.8 Interaction

Talking Hands' mechanical structure and hardware architecture have been described, but to guarantee usability, interactions between the user and Talking Hands must be clearly defined. Relying on physical and cognitive ergonomics, interactions between the system and the user have been developed in parallel in order to optimize the whole system. In this section the two main activities: Record Gesture" and "Talk", are summarized respectively in Figure 3.25 as activity diagrams. The round labels 0 and 1 in Figure 3.25 describe how the two diagrams are connected.

Three stages of interaction will be illustrated:

- Interaction between the user and the data glove;
- Interaction between the device and the application;
- Interaction between the final user and the whole system.

Each stage is represented by a column in activity diagrams. In Subsection 3.8.1 we are going to illustrate how reset procedure works, this solution has been found to remove external accessories as camera etc. to understand the initial orientation of Talking Hands system.


USER
SMARTPHONE APP
DATA GLOVE
Switch-ON
Haptic feedback
Turn on the glove
Connection entstablished
Send request connection to Smartphone using BLE protocol
User Decides to
Press Reset Button
Talk
0
Record Gestures
Acquire data and parse them to be send on BLE
Perform a gesture
Run Gesture Recognition Algirithm
Send Gesture's data
1
Receive gesture recognition information
Gesture has been recognized
Speak using speech synthetizer
Gesture has not been recognized
Provides Feedback to user about translation process using haptic motor
Provides Feedback to user Visualizing recognized word

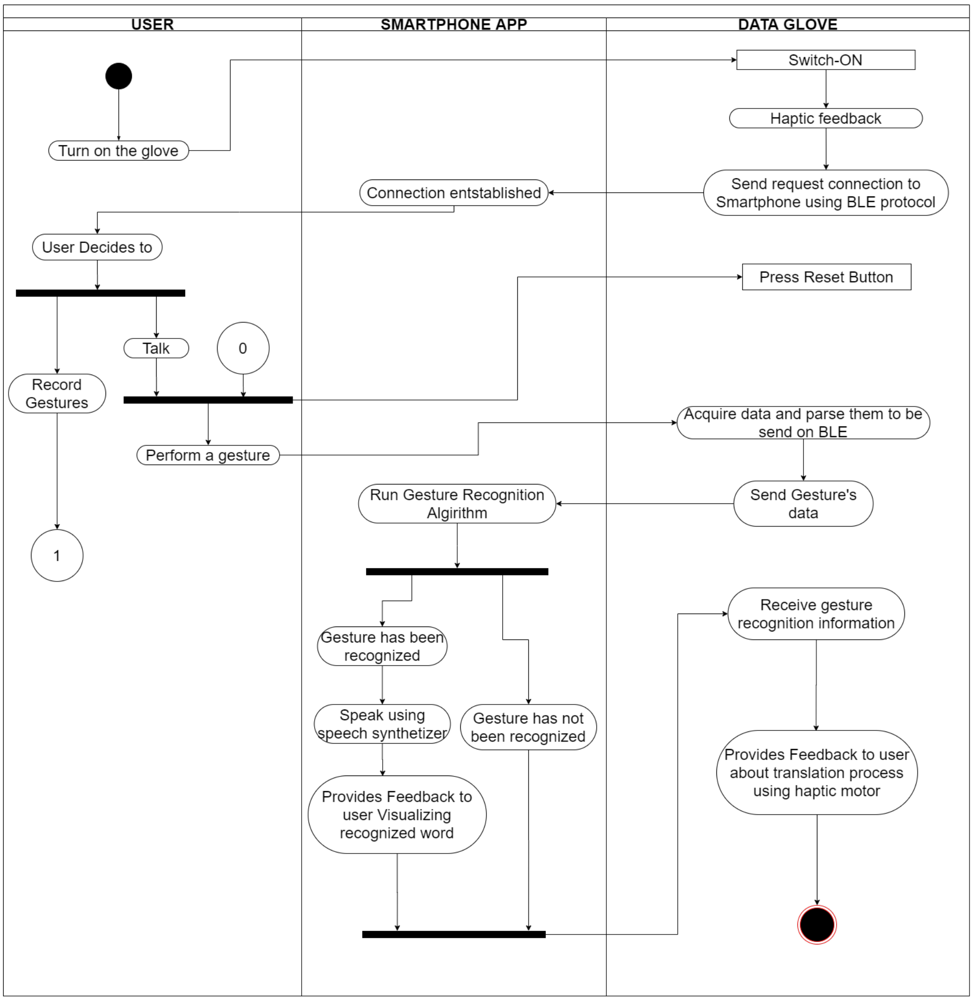

USER
SMARTPHONE APP
DATA GLOVE
Switch-ON
Haptic feedback
Turn on the glove
Send request connection to Smartphone using BLE protocol
Connection entstablished
User Decides to
Press Reset Button
Talk
0
Record Gestures
Acquire data and parse them to be send on BLE
Perform a gesture
Run Gesture Recognition Algirithm
Send Gesture's data
1
Receive gesture recognition information
Gesture has been recognized
Speak using speech synthetizer
Gesture has not been recognized
Provides Feedback to user about translation process using haptic motor
Provides Feedback to user Visualizing recognized word

FIGURE 3.25: Interaction Activity Diagram - Talk and Record Gesture

3.8.1 Reset procedure

One of the fundamental interactions for the correct functioning of Talking Hands is IMUs' reset procedure. Whenever the user rotates from his initial reference system (frame) to another point in space, the IMU will detect rotations respect the initial frame. Gestures have been recorded in relation to the initial frame, if the user moves/rotates from the initial frame without the aim of performing a gesture, sensors will detect a false rotation. This means that the gesture recognition algorithm will fail in recognizing gestures.

A practical example is reported below: In our dictionary we have different gestures already recorded, including "hello" simply raising the hand close to the shoulder, as shown if Figure 3.27a. If we do not move from initial frame and than we perform "hello" gesture, the system will correctly recognize it. If we rotate 180 degree around gravity axis, and then we perform the gesture "hello" the system will not recognize it, because all data are shifted with a rotation of 180 degree as in Figure 3.27c.

For this reason, a reset button has been inserted, one physical and one digital on the app as shown in Figure 3.26. The first was inserted on the glove taking into

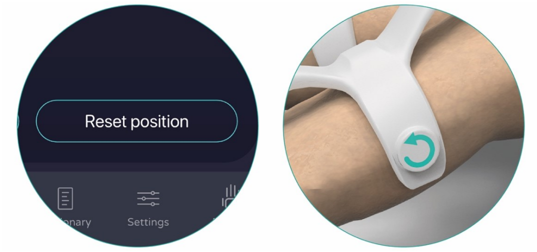
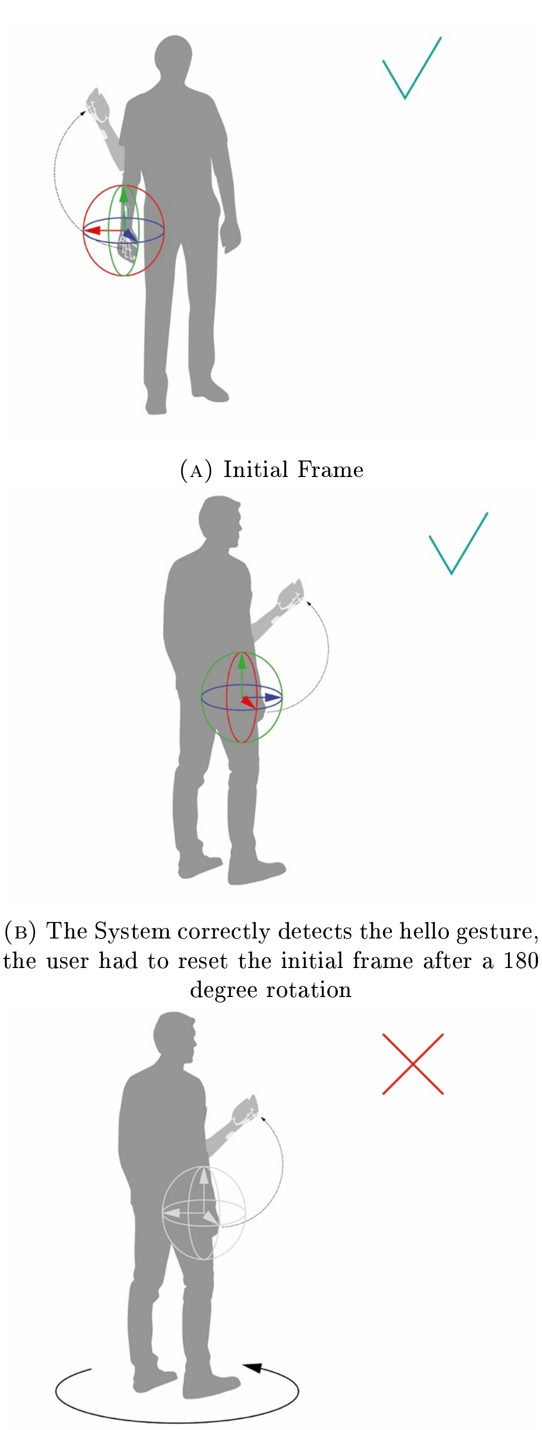


Figure 3.26: Reset Button Position

consideration an independent user. In the second case, we are talking about non-autonomous users for which the speech therapist or tutor will manage reset procedure from the smartphone.



(A) Initial Frame

(B) The System correctly detects the hello gesture, the user had to reset the initial frame after a 180 degree rotation

(C) The System does not detect the hello gesture, user did not reset the initial frame after a 180 degree rotation

FIGURE 3.27: Explanation of Reset Procedure

3.9 Discussion

Respect to other works Talking Hands differentiates mainly because aims to solve problem in the real market and not only for research purposes. A lot of attention has been gave to Human Machine Interaction and Design. Talking Hands as wrote in

Section 3.1 must be wearable and it does not have to influence user's tact/movements. Other studies did not reach this degree of detail and consistency in every single piece of hardware or mechanical structure. Nevertheless it is also true that Talking Hands system cannot recognize with an high percentage of accuracy (>90%) a large set of gesture in the same dictionary (>200 gestures). What has been done in this research is finding a trade-off between cost, usability and performances. This device will be sold at less than 1.000$ per piece, imagine that Cyberglove™, used in a lot of researches (see 2.1) is sold at 13.750$.

Chapter 4

Gesture Recognition Process

Talking Hands is able to collect data of hand and arm orientation and fingers' flexion. Once data have been collected, filtered and properly processed thanks to firmware written on Data-Glove's memory, data will be ready as input for two different algorithms: static or dynamic gesture recognition algorithm. They are respectively described in Sections 4.1 and 4.3. We remind that in case of static gesture recognition, configurations of hands and arms do not involve any motion. Dynamic gestures, like "Good morning" and "House" in American Sign Language for example, express their meaning also through movements. The main difference between static and dynamic signs classification lies in absence/presence of a temporal structure. As previously discussed, a dynamic sign is characterized by spatial and temporal information: the same hand configuration can have two (or more) different meanings according to the performed movement. As a consequence, an algorithm for the classification of dynamic signs must be able to store the information coming from sensors while a gesture is performed.

In case of static gestures, parameters to be analysed are the following:

1. Hand orientation:

   Palm orientation

   Proximal Phalanges

   Intermediate Phalanges

2. Forearm orientation

In case of dynamic gestures we can update the list in this way:

1. Hand orientation and movements:

   Palm orientation and movements

   Proximal Phalanges

   Intermediate Phalanges

2. Forearm orientation and movements

3. Movements of all the previous elements in the list, for example about forearm rotation performing a gesture:

Backward or forward

Clockwise or counter-clockwise

Left or right

This means that in case of dynamic gestures movements are really important, vice versa they are totally ignored in static gesture recognition. This because in dynamic gesture recognition moving an articulation clockwise or counter-clockwise can change a gesture's meaning.

The strategy to simultaneously use those algorithms is described in Figure 4.1.

In this Chapter we will compare Talking Hands also with some studies about Sign language Recognition, but is important to remember that Talking Hands is a wearable device for gesture recognition that is oriented to support people with vocal impairments in ordinary life. This device does not translate sign languages; it will be used as an AAC device and gestures do not have a predefined meaning, the word associated to a gesture will be chosen directly by the user. Talking Hands has been thought as a user-friendly device with a great portability that allows to vocal impaired people a basic interaction with everyone. These goals are achieved through software design solutions that allow to simplify the different tasks.

The main goal of software parts is to develop a system which allows users also to customize their vocabulary and to be able to fully customize their own dictionary autonomously, without having a standard reference as a Sign Language. In case of people with Autism Spectrum Disorder or Down Syndrome, it will be really hard to teach them a sign language; an easiest approach is to record gestures already used from those patients inside Talking Hands application. People with some motor diseases will face many problems to use a sign language, also if they know it. For example some user cannot move fingers so they must use their signs to talk and Talking Hands has to face this situation too.

Strategies and methods adopted to solve these problems customizing dictionaries are explained in Subsection 4.1.1.

4.1 Static Gestures Recognition

The work described in this section (Static) has been done using the first and second prototypes of Talking Hands, in evolution with the hardware and design of Talking Hands system. Also the software and the firmware have continuously been integrated and developed.

There are many limitations in gesture recognition systems. The first challenging tasks is the collection of movements data. In Talking Hands data acquisition of user's movement has to be simple enough to realize a portable system. Second, the translation of a meaningful gesture must be conducted in real time. The translation within

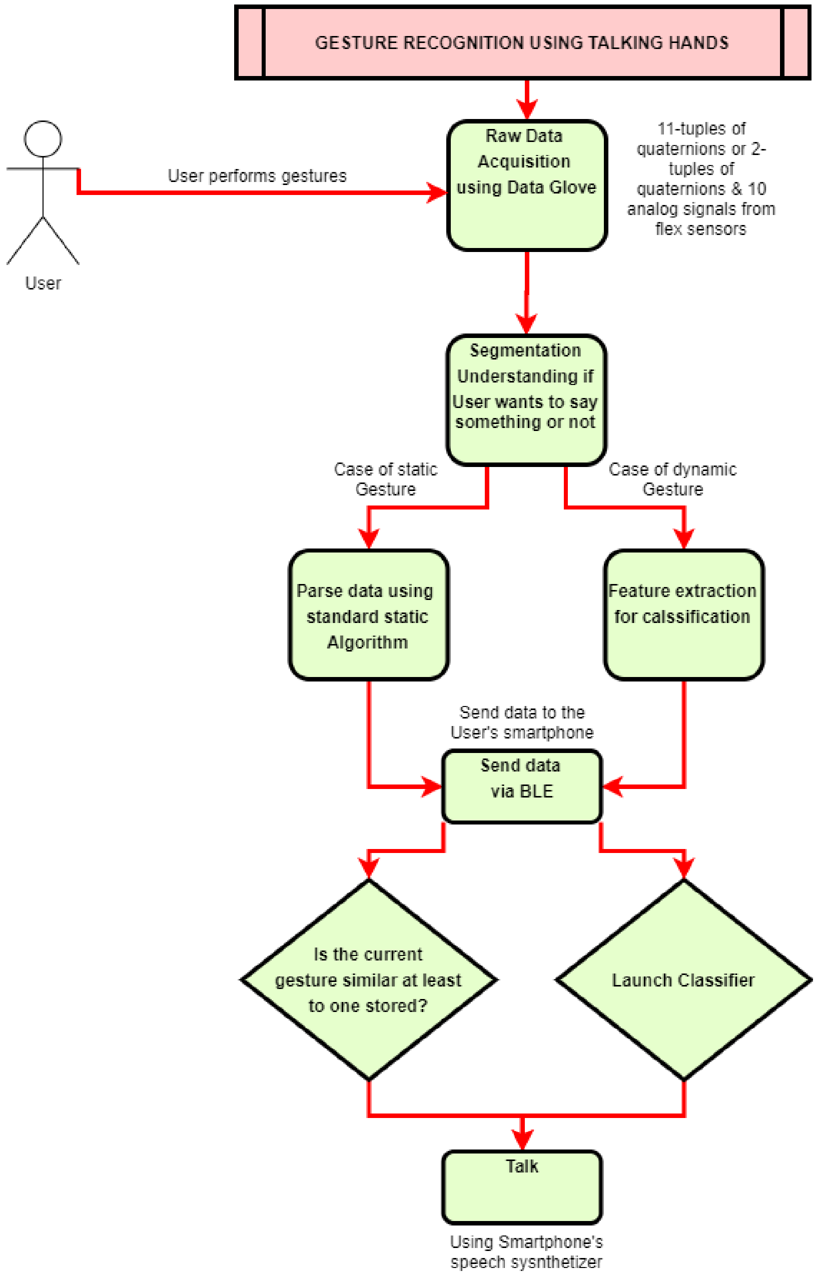

GESTURE RECOGNITION USING TALKING HANDS
User
User performs gestures
Raw Data Acquisition using Data Glove
11-tuples of quaternions or 2-tuples of quaternions & 10 analog signals from flex sensors
Segmentation Understanding if User wants to say something or not
Case of static Gesture
Case of dynamic Gesture
Parse data using standard static Algorithm
Feature extraction for calssification
Send data to the User's smartphone
Send data via BLE
Is the current gesture similar at least to one stored?
Launch Classifier
Talk
Using Smartphone's speech sysnthetizer

FIGURE 4.1: Gesture recognition using Talking Hands System

a very large set of gestures, e.g. an entire sign language, needs an heavy computation that can only be achieved in real time with powerful hardware and software systems. Furthermore, a system to translate gestures in voice should reconstruct phrases' grammar structure, because also sign languages for example are very different from their respective spoken languages. Different papers and studies of last decades face with these challenging tasks.

Most of the works in gesture recognition use advanced mathematical tools, such as Neural Networks ([75]), Hidden Markov Models ([24], [29]), Support Vectors Machines ([67]), Fuzzy C-Means Clustering ([76]). Talking Hands uses a simpler solution and the gesture recognition is based on a deterministic approach, using a distance function defined on the space of the sensors data. Nevertheless, this approach reaches a high

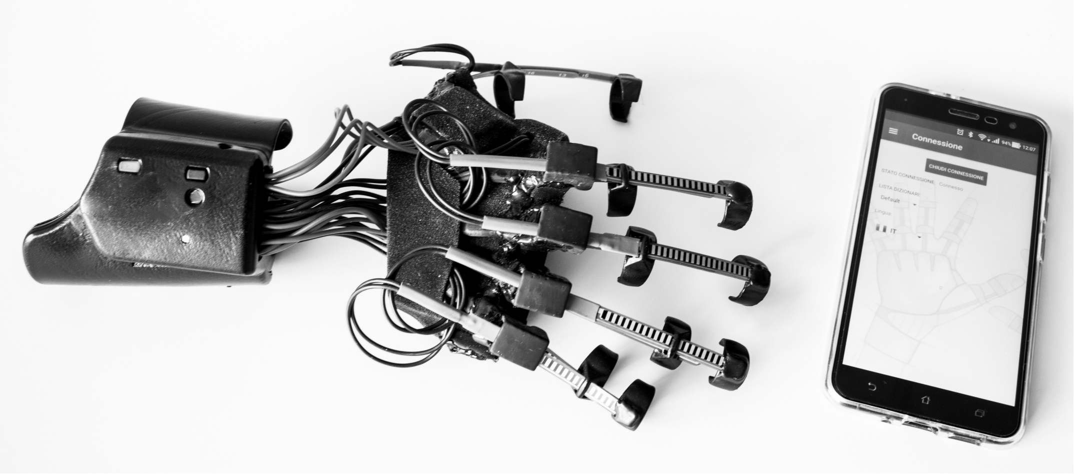


Figure 4.2: Talking Hands Set

level of translation, both on recognition rate and on the number of gestures point of view, to some extent better than the other modelling solutions. The proposed solution can not handle with dynamic gestures and this is the major drawback, but we will discuss also a dynamic gesture recognition approach in Section 4.3. However, a satisfying communication experience can be offered to the final user also with the Static approach.

The software of Talking Hands consists of two main modules: one on the glove (firmware) and one on the smartphone. The firmware pre-processes the sensors data and establishes if the user is performing a meaningful gesture. The smartphone receives data from the glove and uses the speech synthesizer to talk. The translation of the gesture into a text word can be implemented both on the firmware and the smartphone application, depending on the product version. For example in the very first prototype the gesture recognition algorithm has been implemented directly on the glove and the smartphone was used only as speaker.

4.1.1 Translation through Scenarios

Also if Talking Hands does not translate entire sign languages, it can guarantee to a deaf person (or to a vocal impaired) a good communication through a scenario translation, that is one of the most important novelty of this work.

We define a *Scenario* as a set of gestures that Talking Hands can translate in a single session. Hence the system can translate the gestures of a scenario at time, that can be selected through the smartphone application. The user can switch among the scenarios on-line, i.e. during the usage without the need of re-initializing.

This approach leads to some important advantages. In a scenario approach, gestures can be easily recorded by the user through the smartphone application. The user can associate a gesture to a word, a letter, a sound or to an entire phrase and then the gesture is assigned to one or more scenarios, as shown Figure 4.3. This approach enlarges the set of gesture that the system can translate, without losing reliability. Hence, the same gesture can have more than one translation in different scenarios.

Moreover, similar gestures would not be misunderstood if they are not in the same scenarios. Since the number of possible scenarios is limited only by the memory of the smartphone, the user can have a huge set of gestures, where the limitation is due mainly to the cognitive load of the user which has to remember and use properly the scenarios and their gesture. In the actual prototype, the maximum number of gestures in each scenario is about $40 - 50$, but this limitation is due only to the correlations that occurs among large set of gestures. Thanks to these advantages, the translation through scenarios offers a good communication for the vocal impaired.

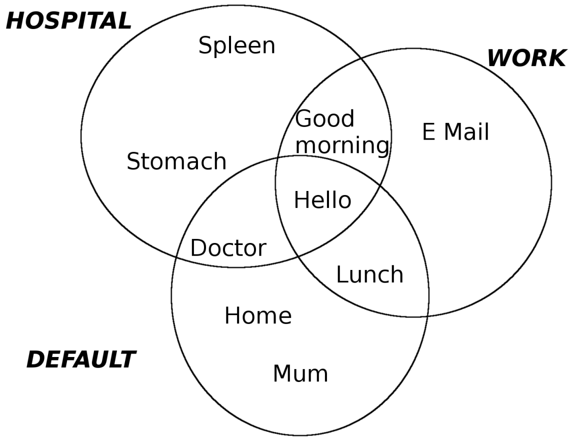


FIGURE 4.3: Schematic representation of three simple scenarios

4.1.2 Distance Function

The algorithm uses a distance on the space of vectors that represent the gestures both to establish if the user is performing a gesture and to link the gesture to its translation. This distance is defined to be both accurate and robust, i.e. if the data-glove has one or two broken sensors, the translation still works.

We formally introduce a proper notation to give a clear presentation of the distance function. We use quaternions to have information about orientations of the hand and of the forearm, with respect to the initial position, using the results in [77]. This choice avoids the well-known gimbal-lock problem of the Euler angles and uses less bytes than the Direct Cosine Matrix (DCM). The data coming from sensors at each sample time consist of 18 integers: 4 for each quaternion of the two IMUs and one for each flex sensor. We denote the quaternions coming from the IMUs of hand and forarm with $\boldsymbol{h} = (h^1, h^2, h^3, h^4)$ and $\boldsymbol{a} = (a^1, a^2, a^3, a^4)$ respectively. According to the BOSCH BNO055 data-sheet, the values of h^i and a^i are in $[-2^{14}, 2^{14}]$, where 2^{14} is a scale factor. We remark that, since we are using quaternions that describe rotations in the 3D space, the following identity holds

$$\|\boldsymbol{h}\| = \sqrt{(h^1)^2 + (h^2)^2 + (h^3)^2 + (h^4)^2} = 2^{14}$$

so we define $\mathcal{Q} = \{\boldsymbol{h} \in \mathbb{Z}^4 : \|\boldsymbol{h}\| = 2^{14}\}$. The flex sensors data are denoted with $\boldsymbol{f} = (f^1, \dots, f^{10})$ and their values are in $[0, 1000]$: if $f^i = 0$, the respective finger joint is totally bent. So we define $\mathcal{F} = [0, 1000]^{10}$.

A data package coming from sensors during a single loop of the micro controller is denoted with

$$\boldsymbol{s} = (\boldsymbol{h}, \boldsymbol{a}, \boldsymbol{f}) \in \mathcal{Q}^2 \times \mathcal{F} = \mathcal{S}$$

which is a 18 dimensional vector. With this notation, a distance function is a function

$$d : \mathcal{S} \times \mathcal{S} \to \mathbb{Z}_{\geq 0}$$

and it has to be equal to zero if and only if $\boldsymbol{s}_1$ and $\boldsymbol{s}_2$ describe the same gesture. The straightforward definition of euclidean distance is meaningless in $\mathcal{S}$, due to the two quaternion $\boldsymbol{h}, \boldsymbol{a}$ components. Hence, the two quaternions $\boldsymbol{h} = (h^1, h^2, h^3, h^4)$ and$-\boldsymbol{h} = (-h^1, -h^2, -h^3, -h^4)$ represent the same orientation, but their euclidean distance is not equal to zero. To overcome this issue, a proper distance function in $\mathcal{Q}$ must be used. Following the results in [77], we define

$$\varphi : \mathcal{Q} \times \mathcal{Q} \to [0, 1000] \qquad \varphi(\boldsymbol{h}_1, \boldsymbol{h}_2) = int\left[\frac{2000}{\pi} \arccos\left(\frac{|\boldsymbol{h}_1 \cdot \boldsymbol{h}_2|}{2^{28}}\right)\right] \tag{4.1}$$

where $|\boldsymbol{h}_1 \cdot \boldsymbol{h}_2|$ indicates the absolute value of the standard dot-product. The multiplication factor $2000/\pi$ is introduced to have the same order of magnitude between the distances of quaternion and flex sensors, while the normalization factor of 2^{28} derives from (4.1.2). The distance between two data vectors $\boldsymbol{s}_1 = (\boldsymbol{h}_1, \boldsymbol{a}_1, \boldsymbol{f}_1)$ and $\boldsymbol{s}_2 = (\boldsymbol{h}_2, \boldsymbol{a}_2, \boldsymbol{f}_2)$ is defined as

$$d(\boldsymbol{s}_1, \boldsymbol{s}_2) = \varphi(\boldsymbol{h}_1, \boldsymbol{h}_2) + \varphi(\boldsymbol{a}_1, \boldsymbol{a}_2) + \sum_{i=1}^{10} |f_1^i - f_2^i|,$$

which is the sum of quaternions and flex sensors distances. The overall distance computation requires: 27 sums, 10 multiplications, 10 comparisons, 2 arccos evaluations. This function is very accurate and it is equal to zero if and only if $\boldsymbol{s}_1$ and $\boldsymbol{s}_2$ have exactly the same orientations and flex sensors values. However, it is not fault tolerant: if there is a broken sensor, the distance function does not recognize it and the whole system fails. To overcome this issue, we introduce a threshold $M \in \mathbb{Z}_{\geq 0}$ and we consider two vectors $\boldsymbol{s}_1$ and $\boldsymbol{s}_2$ the same gesture if

$$d(\boldsymbol{s}_1, \boldsymbol{s}_2) < M$$

Tuning the parameter M, we trade off accuracy and robustness of the system.

4.1.3 Gesture Recognition Algorithm

The gesture recognition algorithm of Talking Hands is simple, deterministic and it can run in the microprocessor of the data-glove. It is a real-time checking algorithm, i.e. it processes and checks all data coming from the different sensors continuously. Hence Talking Hands does not need any external pc. In Figure 4.4 the high level flow chart of the algorithm is shown.

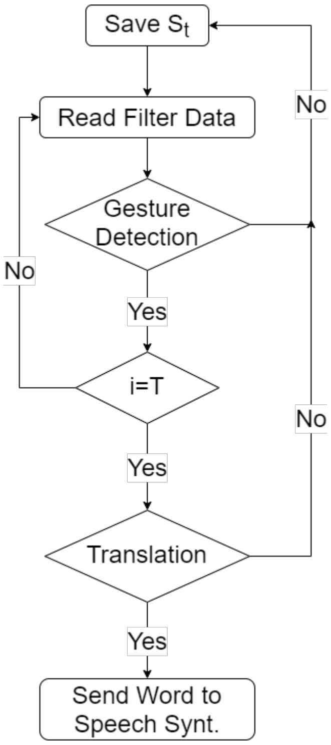


FIGURE 4.4: High Level Flow Chart of Talking Hands

The most important parts of the algorithm are the gesture detection, which establishes if the user is performing a meaningful gesture, and the translation, which links the gesture with the corresponding output. The high level flow chart of the gesture recognition algorithm is composed by the following steps:

1. Filter data to clear sensors noises;

2. Gesture Detection: determine if the user is performing a gesture. To achieve this goal, the algorithm computes the distance in time of the sensors data. If the distance is larger than a given threshold, it deduces that the user is moving from a gesture to another and is in a transition phase: in this case, the algorithm restarts. More precisely, the system saves the vector $\boldsymbol{s}_\tau$ of the current gestures, where τ indicates the actual loop. In the next micro controller loops, the algorithm checks if

$$d(\boldsymbol{s}_\tau, \boldsymbol{s}_{\tau+i}) < M \qquad \forall i = 1, ..., T$$

where $T \in \mathbb{Z}$ is the number of loops that the same gesture has to occur to be considered. Consequently, T is a parameter of the system: increasing too much this value, the system would have a delay in translation; with a too low value, the system could not distinguish among gestures and transition phases.

If $\exists j \in 1, \dots, T$ so that $d(\boldsymbol{s}_\tau, \boldsymbol{s}_{\tau+j}) > M$, the system overwrite the value of $\boldsymbol{s}_\tau$ with $\boldsymbol{s}_{\tau+j}$.

3. Translation: it links the gesture with its meaning. A scenario can be considered a finite collection of couples $(\boldsymbol{s}, w)$, where $\boldsymbol{s} \in \mathcal{S}$ is the characteristic gesture of the gesture and w_i is the string of its translation. Hence we define a scenario as $S = \{(\boldsymbol{s}_i, w_i)\}_{i=1,\dots,N}$, where N is the number of gestures in the scenario S.
   - if the actual gesture s_τ is sufficient close to the characteristic vector of a recorded gesture, i.e. $\exists i = 1, \dots, N : d(\boldsymbol{s}_\tau, \boldsymbol{s}_i) < M$, the algorithm associates $\boldsymbol{s}_\tau$ with the translation of the closer recorded gesture, i.e. w_j with $j = \arg\min_{i=1,\dots,N} d(\boldsymbol{s}_\tau, \boldsymbol{s}_i)$.
   - otherwise the algorithm restarts, since no translation is achieved;
4. The achieved translation w_j is sent to the speech synthesizer.

The point 2 and 3 of the previous description can be also switched: firstly the actual data are translated to the closer recorded gesture if the distance is lower than a certain bound; then, if the same translation is maintained for a certain time, it is sent to the speech synthesizer. This last solution is computational expansive since it requires the comparison of each sampled data with the whole set of gestures. Moreover, if the translation is performed in the smartphone application, this requires to send all the sensors data to the smartphone with a continuous communication. Hence, determining if the user is performing a gesture before the effective translation is preferred. This allows to compute only a distance between the actual sensors data and the past ones in each micro controller loop. If the distance is lower than a certain bound M for a certain time T, only the last data are used for the translation, so only a sample of sensors data is sent to the smartphone, reducing the communication load of the Bluetooth module.

4.1.4 Tests

The first prototype of Talking Hands has been carefully tested.

The translation tests are executed as follows. In each test the user performed the entire set of 40 gestures for five times, for a total of 200 translations. The percentage of successful translations is shown in Figure 4.5. The prototype achieves more than 90% of accuracy. The fails are both lack of translations and word misunderstanding. The second test reported in Figure 4.5 achieved a translation rate of almost 80%, but with a broken sensor: this demonstrates the robustness of the system. Talking Hands has an operating time of 6 hours with a 3.7V 1100mAh Li-Ion Battery.

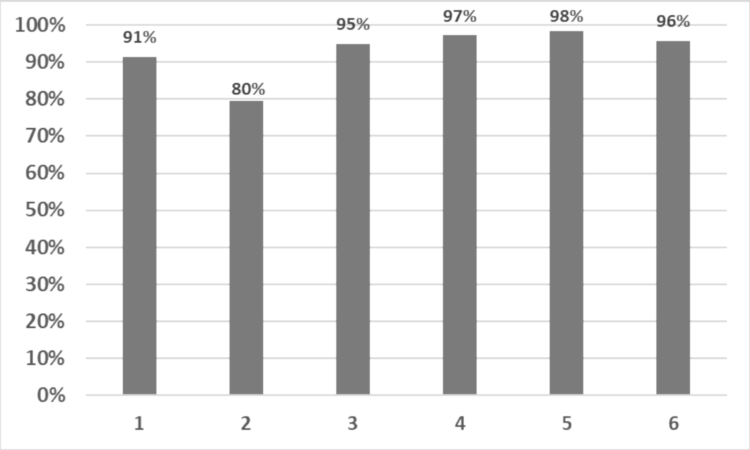


FIGURE 4.5: Successful Translation Histogram

4.2 Considerations on proposed Static Gesture method

Proposed method is preferred for a real application aiming to consumer market. Here we are going to see strength and weak points of the proposed method. One of the strengths is that this method requires a low computational effort. No training is needed respect to Machine learning methods, user needs only to record a gesture using the smartphone application, every microprocessor or smartphone processor will be able to run the gesture recognition algorithm, also cheap smartphones' processors. This guarantees a 100% compatibility with users' smartphone. Using this method Talking Hands' battery life is not compromised because using BLE protocol a gesture is sent only when the device (glove) detects the intention of the user to perform a gesture and connection is not always active because a continuous streaming of information is not needed. Furthermore considering reached accuracy using this method it has been considered good enough for the final application. As weakness there is no possibility to recognize dynamic signs and similar gestures may be hard to be correctly recognized if not performed exactly as in the recording phase, because of the distance function.

4.3 Dynamic Gestures Recognition

This chapter introduces a complete work-flow for the translation of dynamic isolated gestures based on data acquired from a data-glove. A gesture recognition system based on a wearable device represents indeed a more efficient solution with respect to cameras or position trackers for helping speech-impaired people on a daily basis. This study presents experimental results, comparing different machine learning classifiers and discussing their performances both in terms of translation accuracy and computational time. The feature extraction and classification performances of the proposed work flow have been also tested using a public database and compared with other works in the literature, showing improved results. The reported analysis suggests a multi-layer perceptron neural network as the most suitable classifier for the realization of a wearable gesture translation system. Using data of Talking Hands, we perform the classification among 30 dynamic gestures and compare the results obtained by the following machine learning methods: nearest neighbours, linear Support Vector Machine (SVM), random forest, neural networks and naive Bayes classifier. In addition, we have tested the same methods on a publicly available dataset of 95 gestures, comparing our results to similar results available in the literature. Tests provided satisfactory results, showing that our hardware/software solutions outperforms the accuracy levels of translation provided by other studies, even on the same data. Work described in this chapter (Dynamic) has been done using third prototype (described in Section 3.6) of Talking Hands system.

In a real world application, a SL sentence consists of several true-gesture sequences and non-gesture sequences, also called movement epenthesis (ME). As consequence, a SL translation system must perform two subsequent tasks: 1) segmentation, namely splitting a sentence into true-gesture sequences and ME sequences; 2) classification of each true-gesture sequence.

Segmentation

To better understand the segmentation task, let us consider Figures 4.6 and 4.7. In Figure 4.6 a data sequence coming coming from the IMU of the hand module is shown. This data refer to subsequent gestures performed by an user. However, it is not obvious at first glance how many gestures are performed, and where one ends and the consecutive begins. Figure 4.7 shows the same sequence after the segmentation process: six dynamic gestures have been detected, highlighted by the green boxes.

In order to split a data-streaming between true-gesture sequences and ME sequences, a threshold on the velocity computed on the quaternion coming from IMU of the hand module is exploited. Indeed, if a dynamic gesture is occurring, the user is moving his/her hand and a variation on the quaternion could be detected.

The distance function (4.1.2), which has been implemented to compare two quaternions, is not suitable for this segmentation process. Indeed, the quaternions coming from the IMUs could have some computational errors, so they could slightly differ

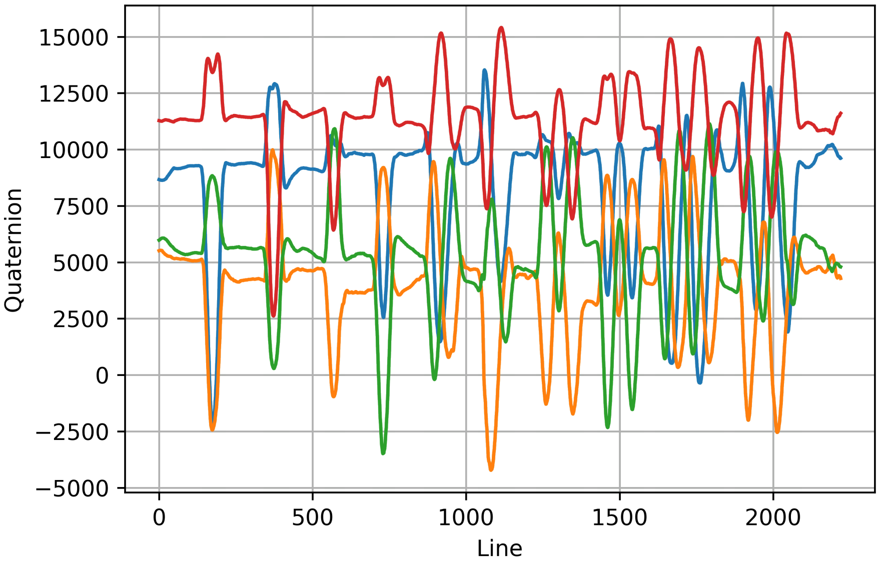


FIGURE 4.6: Data without Segmentation

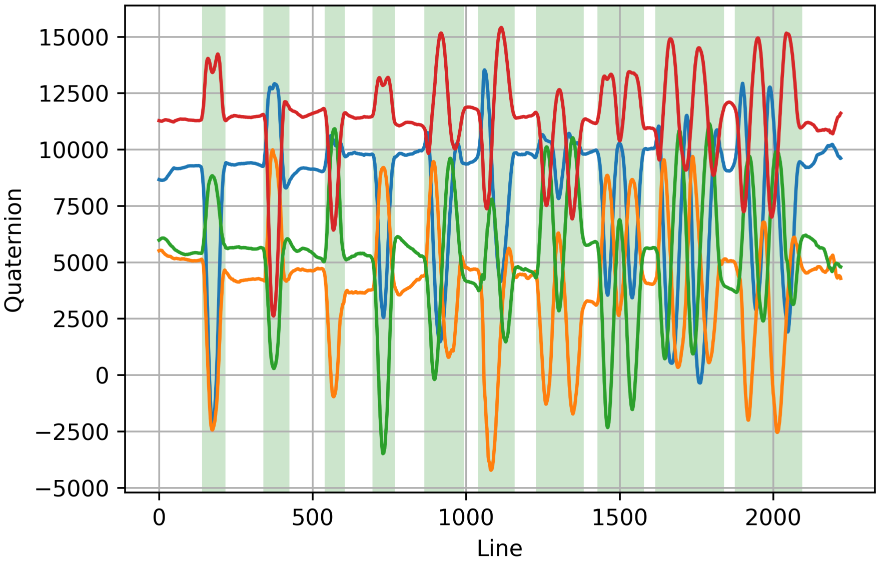


FIGURE 4.7: Segmentation of a data sequence with six gestures

from the unit norm. In that case, the distance (4.1.2) is not reliable. This can be seen in Figure 4.8: even if the system is still, a little velocity is detected.

Differently respect the recognition phase, during the segmentation process we are interested on the distance between two consecutive quaternions. Hence, they can differ only for a small amount and the Euclidean distance could lead to better result. The velocity computed with the Euclidean distance is reported in Figure 4.9. The different gestures are isolated setting a threshold on the velocity signal.

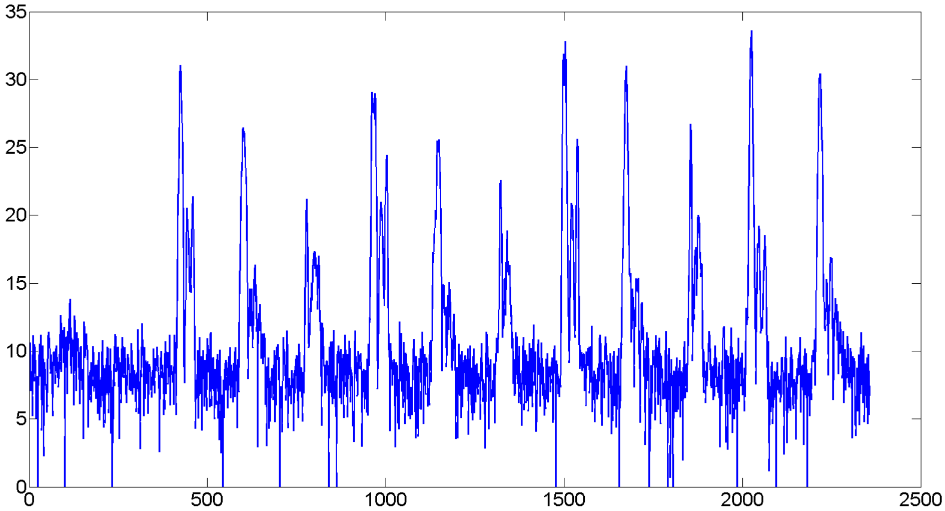


FIGURE 4.8: Velocity computed from the data sequence of quaternion using the distance function (4.1.2).

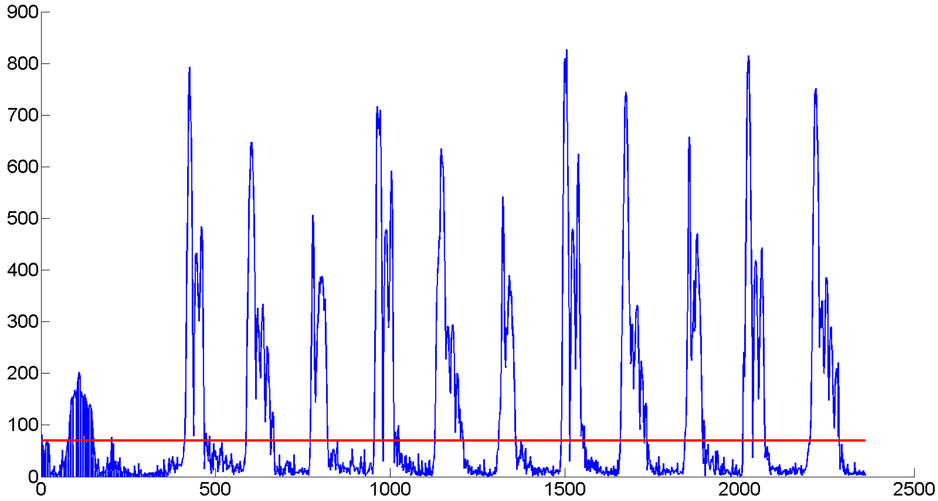


FIGURE 4.9: Velocity computed from the data sequence of quaternion using the Euclidean distance function. The red line indicates the threshold used for the segmentation.

Feature Extraction

Data acquired from Talking Hands cannot directly be used to classify gestures. Indeed, each gesture has to be described by a fixed length vector to implement the different machine learning methods. Moreover, since a data sequence relative to one second consists of about $(8 + 10) \times 70 = 1260$ numbers, classification performed on such a heavy vector could be too computational demanding for a mobile application.

In some works available in the literature (e.g. [63]), the feature vectors refer to meaningful data about the gestures, such as histograms of hand position, configuration and velocity or time and distance of gestures. In this work a representation of gesture data has been chosen using the coefficients of some fitting curves instead. In particular, a cubic spline with three knots has been used for each one of the eight data sequences of the two quaternions. As well known, a spline is a function defined piecewise by polynomials. More formally, let $\hat{q}_{ij}(t_k)$, with $i = 1, 2$ and $j = 1, \ldots, 4$, be one of the component of a quaternion, where t is an integer that varies from 0 to T. To ease the presentation, indexes have been dropped denoting the component simply as $\hat{q}(t_k)$. A fitting spline $q(t)$, namely a piecewise function defined by polynomials, is built as follows

$$q(t) = \begin{cases} \alpha_{00} + \alpha_{01}t + \alpha_{02}t^2 + \alpha_{03}t^3 & \text{if } 0 \leq t < c_1 \\ \alpha_{10} + \alpha_{11}t + \alpha_{12}t^2 + \alpha_{13}t^3 & \text{if } c_1 \leq t < c_2 \\ \alpha_{20} + \alpha_{21}t + \alpha_{22}t^2 + \alpha_{23}t^3 & \text{if } c_2 \leq t < c_3 \\ \alpha_{30} + \alpha_{31}t + \alpha_{32}t^2 + \alpha_{33}t^3 & \text{if } c_3 \leq t \leq T \end{cases}$$

where c_i, $i = 1, 2, 3$ are three uniform distributed points, called knots. Each one of these polynomial functions can be found using the least squares error metric, thus minimizing

$$e = \sum_{t=0}^{T} \left(q(t) - \hat{q}(t)\right)^2 .$$

The constraints have been added that $q(t)$ must be continuous and with continuous first and second derivatives. With these constraints, the function $q(t)$ is given by $16 - 9 = 7$ parameters that will constitute the feature vector that describes $\hat{q}(t)$. Hence, the information of a gesture coming from the two quaternions is described by $7 \times 8 = 56$ numbers, regardless of the duration of the gesture.

We adopted a polynomial of third degree for each data sequence coming from the 10 flex sensors, so they are described by $4 \times 10 = 40$ numbers. As consequence, each gesture is described by a 96-dimensional vector.

We tested the following classifiers for gestures translation: nearest neighbour, linear SVM, random forest, neural network and naive Bayes.

The nearest neighbours classifier uses a k-nearest neighbours algorithm with $k = 3$ and a standard euclidean metric on the feature vectors. The linear support vector machine uses a C-support vector classification to avoid overfitting, with a penalty parameter $C = 0.025$ [78]. The random forest classifier uses 100 trees with a maximum depth of 5 [79]. The neural network is a multi-layer perceptron with one hidden layer constituted by 100 nodes and logistic sigmoid function, namely $f(x) = 1/\left(1 + e^{-x}\right)$, as activation function [80]. The naive bayes classifier implements a gaussian naive bayes algorithm for classification [81].

All these classifiers have been implemented using the Python library Scikit-learn [82], and their respective classes are listed in Table 4.1.

Experiment 2 is based on the data acquisition from [83]. For each hand [83] has recorded those kind of data:

TABLE 4.1: Scikit-learn classes of implemented classifiers

Method	Scikit-learn library	Class
Nearest Neighbours	neighbors	KNeighborsClassifier
Linear SVM	svm	SVC
Random Forest	ensemble	RandomForestClassifier
Neural Network	neural_network	MLPClassifier
Naive Bayes	naive_bayes	GaussianNB

a x position expressed relative to a zero point set slightly below the chin. Expressed in meters.

b y position expressed relative to a zero point set slightly below the chin. Expressed in meters.

c z position expressed relative to a zero point set slightly below the chin. Expressed in meters.

d roll expressed as a value between -0.5 and 0.5 with 0 being palm down. Positive means the palm is rolled clockwise from the perspective of the user. To get degrees, multiply by 180.

e pitch expressed as a value between -0.5 and 0.5 with 0 being palm flat (horizontal). Positive means the palm is pointing up. To get degrees, multiply by 180.

f yaw expressed a value between -1.0 and 1.0 with 0 being palm straight ahead from the perspective of the user. Positive means clockwise from the perspective above the user. To get degrees, multiply by 180.

g Thumb bend measure between 0 and 1. 0 means totally flat, 1 means totally bent. However, the finger bend measurements are not very exact.

h Forefinger bend measure between 0 and 1. 0 means totally flat, 1 means totally bent. However, the finger bend measurements are not very exact.

i Middle finger bend measure between 0 and 1. 0 means totally flat, 1 means totally bent. However, the finger bend measurements are not very exact.

j Ring finger bend measure between 0 and 1. 0 means totally flat, 1 means totally bent. However, the finger bend measurements are not very exact.

k Little finger bend measure between 0 and 1. 0 means totally flat, 1 means totally bent. However, the finger bend measurements are not very exact.

Some considerations need to be undertaken:

1. Talking Hands system does not need and does not acquire data about position compared to a,b,c in the list 4.3.

2. Talking Hands system does not use orientation expressed in Euler angles compared to d,e,f in the list 4.3.

3. Talking Hands system acquire also data about forearm orientation and not just about palm orientation, both expressed in quaternions.

4. Talking Hands system does not acquire bending measures between 0 and 1 as in points: g,h,i,j,k of the list 4.3. Talking Handssystem acquire finger's flexion (bending) of two phalanges: proximal and metacarpal using an 10 bit ADCs to convert analog signal coming from flex sensor in a range starting from 0 to 1023.

5. Talking Hands system utilizes just one glove.

Especially from point 1 in list 4.3 is possible to understand that position does not influences results expressed in Section 4.5.

4.4 Experimental

In this section, the results of two different experiments are presented. In Experiment 1 data from Talking Hands has been used during training and test phases. Each sign has been detected using the previous described segmentation process. Then the feature vectors were computed as described in Section 4.3. In Experiment 2 data from a public database of signs [83] has been used to train and test the different methods.

Since in Experiment 2 the data are not acquired with the Talking Hands' hardware, this experiment has been conducted to compare the Talking Hands' gesture recognition algorithm with others in literature, regardless of the data source. Moreover, using a publicly available database, the comparison with other isolated signs translation processes that have been tested on this database (e.g. [64]) is highly reliable.

Experiment 1. We used data collected from Talking Hands. The data-glove has been wired connected to a pc and computation has been performed off-line using Python. The pc used for the tests is an *Asus K550JK* (hardware specification: Intel Core i7-4710 HQ 2.5 GHz, 8GB RAM). This database consists of 27 gestures classes that do not refer to any signed-word. For example, some gestures are the following: a clockwise (or counterclockwise) rotation of the arm; two clockwise (or counterclockwise) rotations of the arm; a right-left (or left-right) movement of the hand; two right-left (or left-right) movements of the hand; an up-down (or down-up) movement of the hand; two up-down (or down-up) movements of the hand; waving hand; a clockwise (or counterclockwise) rotation of the hand. A total of 6 instances per gesture has been collected from each of the five users involved in this experiment. The final dataset consists of 810 gestures. The segmentation and feature extraction processes described in Section 4.3 have been applied.

Experiment 2. In this experiment we used a publicly available database [83]. This dataset consists of 95 different signs of Australian Sign Language, with 27 samples per sign, captured from a native signer using high-quality position trackers. This was a two-hand system and this is the biggest difference with the our dataset in experiment 1. Moreover, two Flock-of-Birds magnetic position trackers provided 6 degrees of freedom of each hand, i.e. roll, pitch and yaw as well as 3D position. Therefore, differently from Talking Hands, this system used some external device to acquire precise data about the position. Two sensors gloves provided data about fingers' bending. Hence, 22 attributes were acquired. The sample frequency was about 100Hz.

In both experiments, each classifier has been tested using the "hold out" method with three different test sizes, namely 40%, 20% and 10%. The remaining samples were used in training process. For each test size, a Monte Carlo validation has been applied with 100 runs, choosing randomly the training set of each gesture class at each run.

At each run, and for each classifier, we collected the results about classification accuracy, classification time and training time to perform some statistical analysis to compare the classifiers.

Moreover, with the dataset of Experiment 1, we conducted other tests to study the user-indipendence of the proposed method.

4.5 Results and Discussion

4.5.1 Experiment 1

Table 4.2 reports all the statistics about experiment 1. For a better understanding, the box plots in Figure 4.10a shows the statistics on the performances of the considered classifiers when a test size of 10% is used. The box extends from the lower to upper quartile values of the data, with a line denoting the statistical median. The whiskers extend from the box to show the range of the data and the outliers are indicated with circles past the end of the whiskers.

We notice that nearest neighbours and random forest classifiers performed better than the others in terms of classification accuracy (Figure 4.10a). In particular, the Random Forest classifier reaches the highest average of translation accuracy (99.7%). The statistical Wilcoxon rank sum test was performed to compare the mean values reached by the classifiers. The null hypothesis that the classification methods have equal medians at the 5% significance level was not rejected when comparing Nearest Neighbours, Linear SVM and Random Forest, indicating that these three methods are equivalent and have better performances than the other two. The Wilcoxon rank sum test also showed that Neural Network and Naive Bayes are equivalent in terms of classification accuracy.

The results of the experiment 1 with test sizes of 20% and 40% are quite similar to those of 10%, indicating that the number of training samples does not particularly effect the translation accuracy. This is very important in the perspective of a commercial product, since it is infeasible to obtain large training datasets of signs for each user. Our experiment with a test size of 40% achieved a translation accuracy of 99.3% for the Random Forest classifier. However, the results of the three tests are not equivalent from a statistical point of view. Indeed, the Wilcoxon rank sum test conducted on the three tests of each classifier showed that they have different medians.

The neural network classifier is faster than the others in terms of translation time, but its training time lasts longer. However, the translation time is much more important in a real world application, since the training process could be performed offline. Moreover, the training process of the neural network lasted 4.1 seconds on average and this is a reasonable time for a commercial sign language translation system.

Table 4.2: Comparison in experiment 1

Test Size	Method	Classification Accuracy				Classification Time (milliseconds)				Training Time (milliseconds)			
		Mean	St. Dev.	Min	Max	Mean	St. Dev.	Min	Max	Mean	St. Dev.	Min	Max
10%	Nearest Neighbours	99.5%	0.88%	96.6%	100.0%	6.8	1.05	6	14	2.4	0.49	2	3
	Linear SVM	99.5%	0.81%	96.6%	100.0%	3.8	0.67	3	7	22.6	2.16	21	41
	Random Forest	**99.7%**	0.59%	97.7%	100.0%	8.4	1.07	7	16	264.6	44.59	240	560
	Neural Net	98.8%	1.15%	95.5%	100.0%	1.0	0.26	0	2	4059.4	590.87	2319	6549
	Naive Bayes	99.1%	0.95%	96.6%	100.0%	2.5	0.56	2	4	3.0	0.52	2	7
20%	Nearest Neighbours	99.3%	0.69%	97.6%	100.0%	11.1	0.60	10	13	2.1	0.35	2	3
	Linear SVM	99.3%	0.62%	97.6%	100.0%	6.2	0.45	5	7	19.1	0.64	17	21
	Random Forest	**99.5%**	0.46%	98.2%	100.0%	9.7	0.50	9	11	231.1	3.36	227	247
	Neural Net	98.4%	0.91%	95.9%	100.0%	1.5	0.50	1	2	3609.4	551.37	2009	4225
	Naive Bayes	98.9%	0.69%	95.9%	100.0%	3.3	0.48	3	4	2.8	0.37	2	3
40%	Nearest Neighbours	98.9%	0.61%	96.7%	100.0%	19.7	0.77	18	22	1.7	0.44	1	2
	Linear SVM	99.2%	0.44%	98.2%	100.0%	10.1	0.53	9	11	14.6	0.55	14	16
	Random Forest	**99.3%**	0.36%	98.2%	100.0%	12.9	0.44	12	14	206.4	4.18	200	232
	Neural Net	98.0%	0.76%	96.4%	99.4%	2.4	0.49	2	3	3083.4	447.35	1457	3487
	Naive Bayes	98.5%	0.74%	95.8%	100.0%	5.1	0.33	4	6	2.6	0.49	2	3

4.5.2 Experiment 2

As discussed before, we used a publicly available dataset for the second experiment. For each sign, we extracted the feature vector using a cubic spline fitting for the position and the orientation data, while a polynomial fitting has used on fingers' bending data. Therefore, each gesture was described by a vector of dimension $((7 \times 6) + (4 \times 5)) \times 2 = 124$.

TABLE 4.3: Comparison in experiment 2

Test Size	Method	Classification Accuracy				Classification Time (milliseconds)				Training Time (milliseconds)			
		Mean	St. Dev.	Min	Max	Mean	St. Dev.	Min	Max	Mean	St. Dev.	Min	Max
10%	Nearest Neighbours	92.9%	3.36%	84.2%	98.6%	105.8	6.29	96	139	12.6	2.00	11	30
	Linear SVM	79.4%	6.72%	61.1%	91.2%	137.6	14.03	123	178	981.2	20.07	941	1041
	Random Forest	92.6%	3.74%	82.8%	98.6%	24.8	1.51	23	36	943.0	29.63	903	1037
	Neural Network	**97.4%**	1.78%	91.2%	99.6%	2.0	0.36	1	3	20211.4	1021.99	17577	22690
	Naive Bayes	86.3%	5.47%	69.1%	94.4%	18.7	0.79	18	23	10.7	0.76	9	13
20%	Nearest Neighbours	92.0%	2.14%	85.1%	96.8%	256.1	90.35	194	474	12.3	4.35	9	27
	Linear SVM	78.3%	4.10%	67.0%	86.8%	316.6	109.73	242	562	3806.4	28457.56	836	285530
	Random Forest	91.6%	2.47%	85.6%	97.0%	60.3	18.79	45	114	992.3	234.84	844	2571
	Neural Network	**97.4%**	1.03%	94.7%	99.5%	4.3	1.75	3	9	19000.6	7743.48	12965	36567
	Naive Bayes	85.1%	3.20%	75.1%	90.5%	47.7	24.18	33	131	14.4	5.42	11	32
40%	Nearest Neighbours	91.0%	1.67%	86.0%	94.0%	470.9	146.25	265	712	11.3	4.28	6	25
	Linear SVM	78.0%	2.46%	72.3%	83.2%	620.4	219.92	324	1221	780.8	171.71	548	1081
	Random Forest	90.5%	1.80%	83.8%	94.4%	149.5	52.82	77	278	1042.5	284.16	696	2558
	Neural Network	**97.1%**	0.88%	94.3%	98.8%	7.9	3.12	4	20	24259.8	8395.39	11590	35454
	Naive Bayes	82.4%	3.37%	70.7%	88.4%	134.4	52.49	60	198	16.9	5.95	9	35

Table 4.3 shows the statistics on the results obtained with the Monte Carlo cross validation. In this experiment, the neural network classifier performed better than the others in terms of classification accuracy in all the three tests, reaching a 97.4% of accuracy with a test size of 10%. The Wilcoxon rank sum test showed that each classifier was not equivalent to the others.

Also in this case, the neural network is the fastest classifier, with a translation time of about 3 milliseconds. Comparing the translation times, all the classifiers took more time in the second experiment than in the first one. However, the Neural Network increased less than the others, remaining on the same order of magnitude. This is a very important feature from the perspective of future applications, and makes the neural network classifier more suitable than the others to be applied for a translation between a large set of signs. Moreover, it had the smallest performance variation among the compared classifiers, making it the most reliable one. The only drawback of the Neural Network is the large increasing of the training time, which can last more than 20 seconds. However, we think this could be still a reasonable time in a real world application.

4.5.3 User Independence

Other tests have been conducted to test the proposed method on user-independent scenarios. In particular, we divided the dataset of Experiment 1 among the five users. Then, we train the classifiers using the data of four users and test them on the gestures performed by the remaining user. Other tests have been conducted using the data of a single user as train set and the remaining data of the four users as test set. The results are shown in Table 4.4.

The accuracy levels of classifiers trained on a dataset of four users are slightly less than the ones on Experiment 1. However, Nearest Neighbours, Linear SVM and Random Forest classifiers reached an accuracy level of about 96%, which is comparable to other results in literature (see section below for more details). The accuracy of the Naive Bayes classifier has significantly decreased, meaning that this classifier is highly user-dependent.

As expected, the accuracy levels of classifiers trained on a dataset of just one user, which are reported in the righ-hand side of Table 4.4, are worst than the others. In particular, we notice that the Naive Bayes classifier is unable to manage the user-independent scenario, scoring 13.6% on average, while the classifiers that performed better in this case are the Nearest Neighbours and the Linear SVM.

TABLE 4.4: User independence results

Method	Train data of four users					Mean	Train data of one user					Mean
Nearest Neighbours	97.6%	93.3%	97.5%	99.4%	95.2%	**96.6%**	89.2%	80.8%	87.8%	85.1%	85.5%	**85.6%**
Linear SVM	99.4%	92.1%	96.3%	99.4%	96.4%	**96.7%**	91.3%	82.0%	89.1%	90.2%	89.3%	**88.3%**
Random Forest	99.4%	90.9%	99.4%	96.3%	95.2%	**96.2%**	80.3%	73.5%	77.8%	75.3%	71.3%	**75.6%**
Neural Network	97.0%	85.4%	96.3%	97.6%	93.3%	**93.8%**	84.3%	75.8%	86.0%	84.3%	71.6%	**80.2%**
Naive Bayes	93.3%	85.4%	92.6%	93.9%	56.4%	**82.9%**	10.7%	16.8%	10.3%	14.9%	16.6%	**13.6%**

4.5.4 Comparison with other works

TABLE 4.5: Comparison of translation accuracy with similar papers

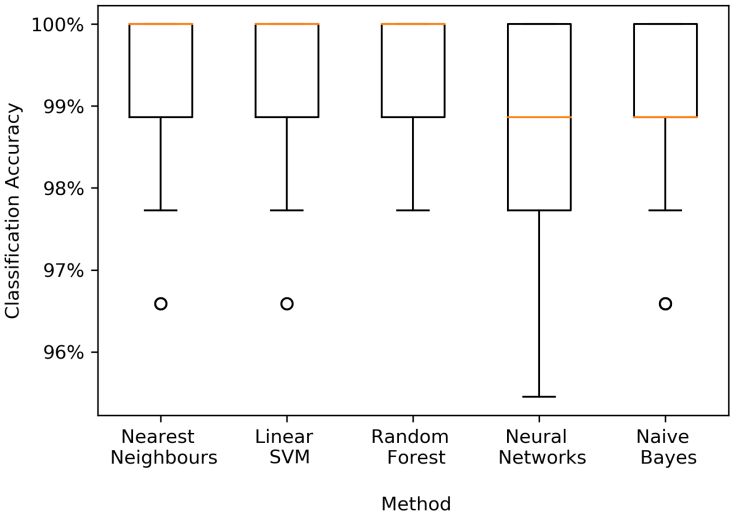
Work	Acquisition Method	Classification Method	# Gestures	# Users	Accuracy
Shukor et al. [60]	data-glove	distance function	9	4	78%
Saggio et al. [**saggio2020**]	data-glove	convolutional neural network	10	7	98%
This work	**data-glove**	**proposed method**	**27**	**5**	**99.7%**
Kumar et al. [**kumar2018**]	Microsoft Kinect	hidden markov model	30	8	84%
Mittal et al. [**mittal2019**]	Leap Motion	long-short term memory neural network	35	6	89.5%
Oz and Leu [63]	data-glove	neural network	50	≤ 6	92%
Kumar et al. [**kumar2017b**]	Microsoft Kinect and Leap motion	HMM and bidirectional LSTM neural network	50	≤ 10	98%
Kosmidou and Hadjileontiadis [61]	EMG and accelerometers	intrinsic mode entropy model	60	3	93%
Gamage et al. [25]	camera	gaussian process dynamical Model	66	1	≈85%
Rozado et al. [64]	position tracker	hierarchical temporal memory model	95	1	91%
This work	**position tracker**	**proposed method**	**95**	**1**	**97.4%**

Table 4.5 reports the classification accuracies of the proposed method and those of other similar papers, together with the acquisition and classification applied and the number of gestures among which the translations occurred.

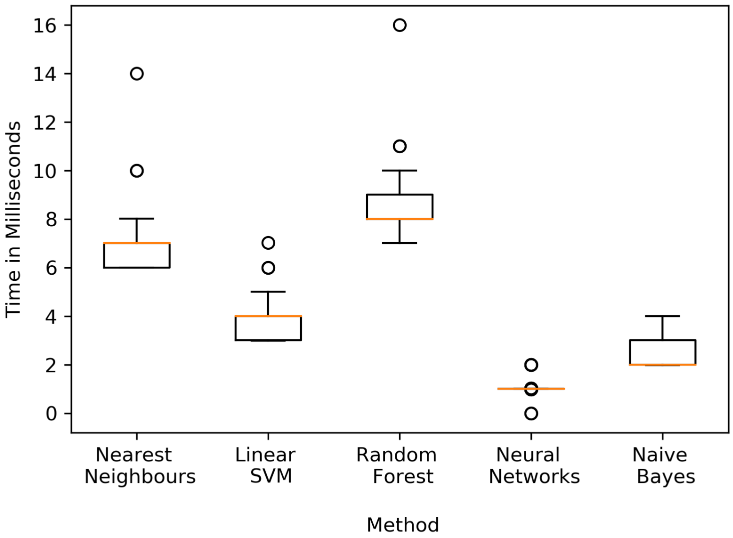
The works differ for the acquisition methods applied, the number of gestures among which the classification occurs and the number of users involved in the study. As a consequence, the comparison can be conducted only on the recognition's accuracy, which gives a loose indication on the overall performances of the systems developed in these works. Talking Hands obtained the highest performance for the translation of isolated dynamic signs, reaching an average of 99.7% accuracy among 27 gesture classes with a Random Forest classifier. The dataset used in [64] and in our second experiment is the same, thus the comparison with that work is more meaningful than the others. As previously stated, since the dataset is the same, this comparison is only between the gesture recognition algorithm implemented. Hence, we only compare the feature extraction process and the different classifiers previously described in this chapter with the analogous steps of [64].

The accuracy obtained with our methods is 97.4% and it is statistically higher then the one obtained in [64], which is 91%. This is quite remarkable, since our method does not directly rely on any temporal classification technique, like HMM or LSTM. Indeed, the temporal information of the different gesture categories is captured by the coefficients of the splines that fit the data sequences. Instead, the algorithm of [64] relies on a Bayesian-like paradigm known as hierarchical temporal memory (HTM) and each gesture category is modelled as a single HMM. Moreover, the algorithm presented in [64] are not feasible for a mobile application running on a microprocessor and a smartphone, differently from the algorithm presented in this chapter for Talking Hands.

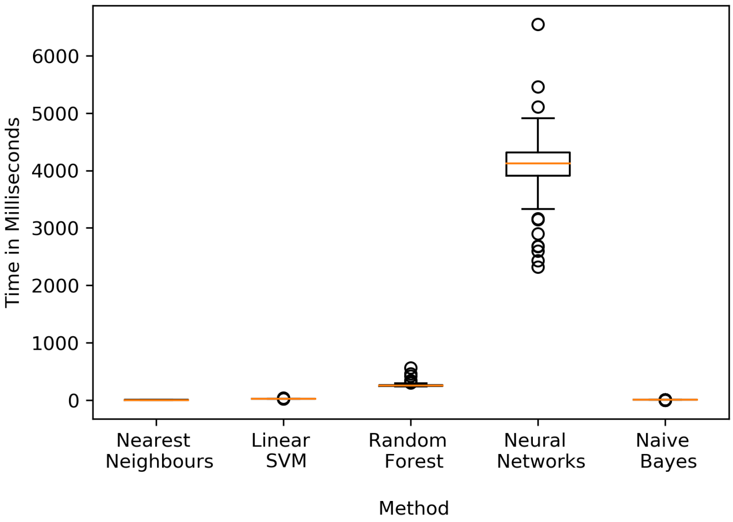
The results about the time of classification and training are missing in [64], hence a discussion on them is not possible.



(A) Statistic of experiment 1 with a test size of 10% - Classification Performance



(B) Statistic of experiment 1 with a test size of 10% - Translation Time



(C) Statistic of experiment 1 with a test size of 10% - Training Time

Chapter 5

New Platform

In this Chapter the new Talking Hands platform will be described. This solution will not be physically realized in time for the end of this PhD thesis, what we are going to illustrate is the new hardware architecture and new mechanical design. This new new configuration comes out after almost one year of tests and feedback from final users and clinicians.

5.1 Future Hardware

Current prototype (described in Section 3.6), although successfully demonstrated its functionalities, but it is not enough reliable and usable to be turned into a commercial product so far. New version of Talking Hands will achieve higher performances, improving sensors' data reliability, mainly designing a new hardware solution.

First of all, lets explain why Talking Hands hardware architecture will change:

- Flex sensors' pins easily break Flexible PCB's tracks (see Figure 5.1) if glove is not worn correctly. We are talking about glove visible in Figure 3.19.
- Flex sensors does not ensure repeatability, measures will change every time user remove/wear the glove. Moreover resistive flex sensors do not have an accurate flexion measurement.
- Flexible PCB constrains usage of a textile Data-Glove as wearable solution. Flex sensors and Flexible PCB do not allow to easily customize sizes for children and depends on user's phalanges length. This is a problem also for glove's sizes, and an FPCB printed for an adult must be different for a kid. This means that there is a need to produce different sizes of: Flexible electronic printed circuit; Flex sensors; textile gloves.
- Flex sensors are very expensive even for large quantity orders, IMUs are cheaper and performs better as finger tracker;
- Flex sensors needs to be calibrated, mapping the range of minimum/maximum flexion user is able to perform. Each user should have his own flexes' configuration, this augment system complexity.

- BNO055 (IMU) needs to be calibrated for a correct usage, replacing it whit a new BNO080/085 will remove the calibration procedure because those IMUs have a built in auto-calibration procedure, in this way the user does not need to perform specific movements before starting to use Talking Hands.

- Textile glove removes user's tact.

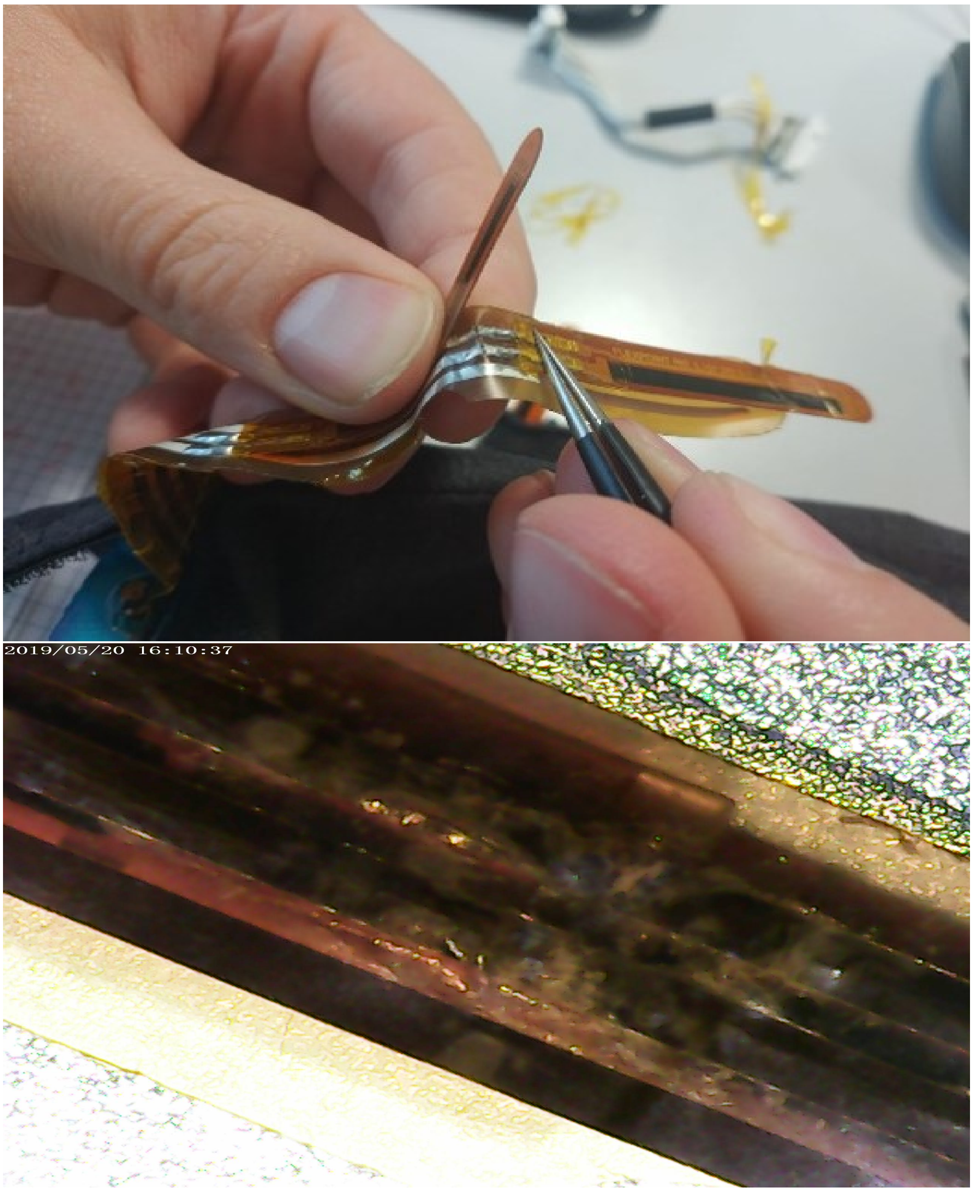


FIGURE 5.1: Broken F-PCB's tracks after few hours incorrect usage, flex sensor's pin has been stressed during usage, white surface means broken bus

To solve those problems a new fingers' system composed by nine IMUs, one for each finger's phalanx (proximal and intermediate) excepted for thumb has been created. Thumb has only one IMU because it does not have an intermediate phalanx, thumb's IMU will be placed on top of distal phalanx, because the distal phalanx is the most expressive (talking about gestures), because thumb's proximal phalanx just follows movements of thumb's distal phalanx. Use two IMUs for the thumb will be redundant, augmenting costs, complexity and wearability. IMUs' configuration about finger module has been illustrated in Figure 5.2.

Also the schema about the hardware architecture and microcontroller - peripherals communication has been updated (see Figures 5.3 and 5.4).

Using new platform's hardware architecture problems of previous list will be solved, a fully wearable industrial prototype with enough durability and reliability to be used in real world applications will be developed. The advantages with respect to the current solution are outstanding:

- Fully Adaptable and wearable: IMUs will be mounted in the glove itself and will be fully adaptable to any person, overcoming the problem related to the size of the glove and the size of the bending sensors.

- Durability: the welding of the circuit connecting the bending sensors to the acquisition unit are the weakest points and currently subject to breaks and malfunctioning after usage (see Figure 5.1). Eliminating this part will make Talking Hands usable for long term.

- Sensitivity and reliability: the use of IMUs will increase precision and sensitivity in detecting flexions and finger movements. This was already tested in a functional prototype.

- Usability: the use of flexible multi-sensing platform integrated in a wearable bracelet integrated in the e-Glove - only requiring housing for the battery - will strongly improve the usability and wearability of the device for patients.

- Cost: using standardized flexible electronics and substituting the bending sensors with IMUs, the production cost for will be reduced by 20-25%, from current 250€ to less than 200€ / unit. It seems stranger but buying >1000 IMUs will guarantee a cost of 5€ per IMU comparable to flex sensors. Plus having a more precise fingers tracking and without the need of producing an FPC, but encapsulate IMUs inside fingers' rings.

Also software improvements will be performed in the new version. In Section 4.3 strategy and method behind dynamic gesture recognition have been explained. In new version of Talking Hands another objective is to port software developed for PCs into smartphones' applications, developing a new app which integrates dynamic gesture recognition implementing a simple solution that will not compromise the final user experience.

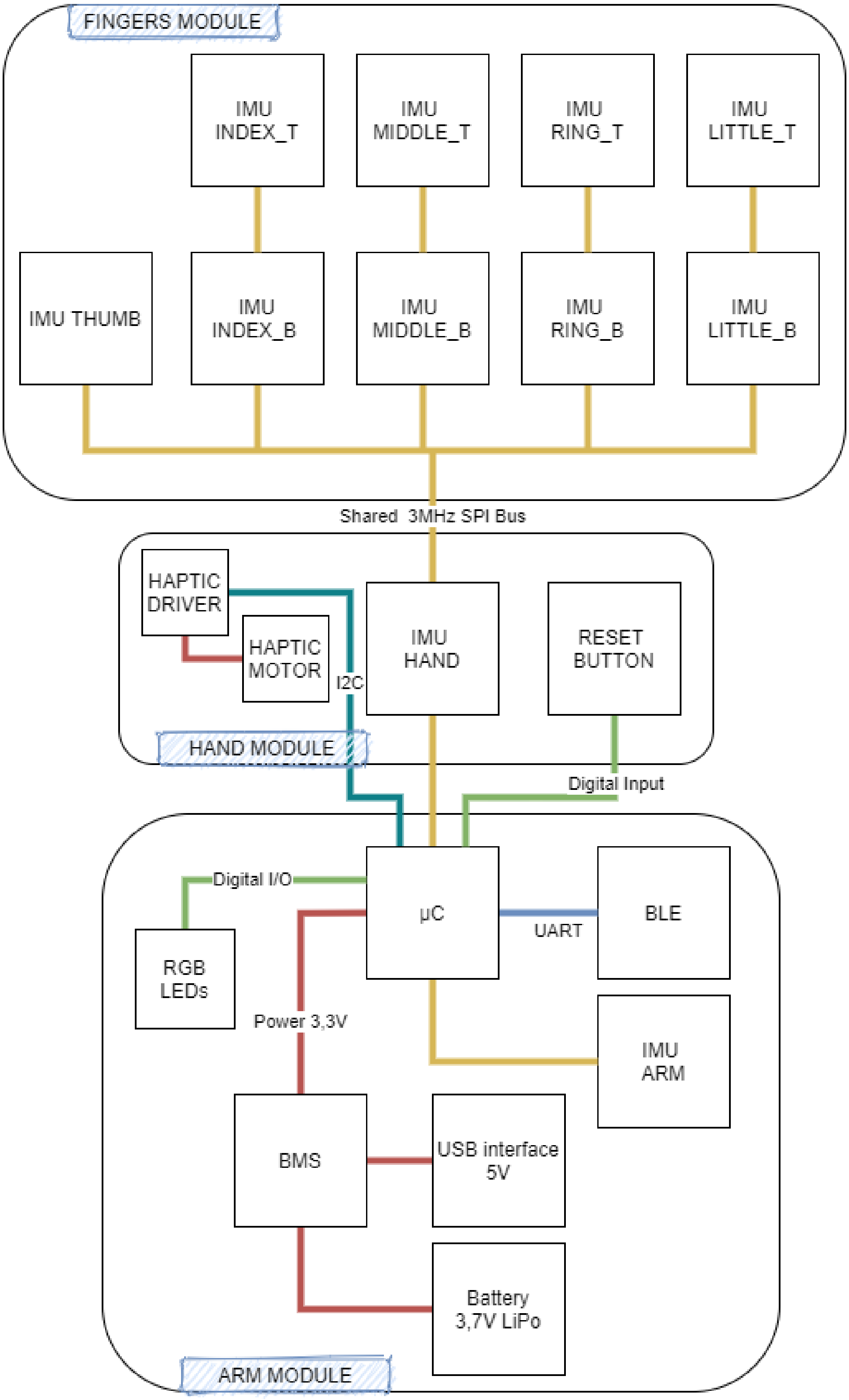


FIGURE 5.2: Full IMU Architecture

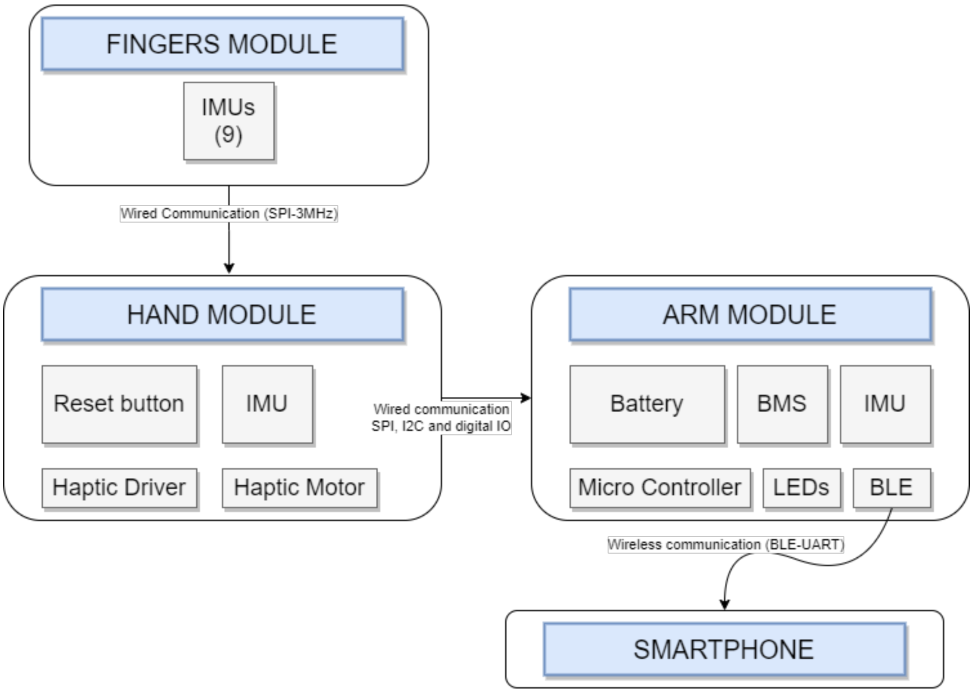


FIGURE 5.3: New Hardware Architecture

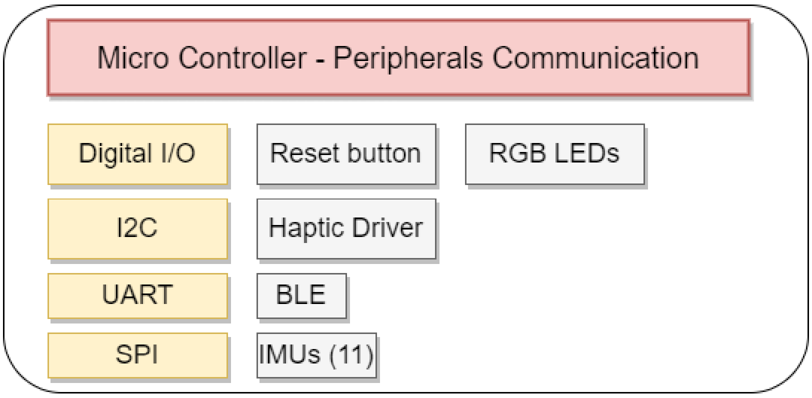


FIGURE 5.4: New microcontroller - Peripherals Communication

5.2 Future Design

Starting from the design solution described in Subsection 3.6.1 and considered the new hardware proposed in Section 5.1, the bulky flex sensors will be replaced by small sensors (IMUs) that can be positioned on fingers' phalanges (proximal and intermediate). This was a huge step forward, which allowed us to work further on decreasing the material on fingers' surface.

This step brought device's design from the "glove" hypothesis to the "exoskeleton" hypothesis. This study aimed to reduce hardware's surface on fingers to the minimum possible surface, see Figures 5.5 and 5.6. Even if the two alternatives go back a little to the first prototypes, in combination with the new hardware solution, they were the key to accomplish the final objective. In this way area of the palm is not totally covered as in the previous cases, releasing fingers from all the material that up to this point was necessary to contain internal hardware components.

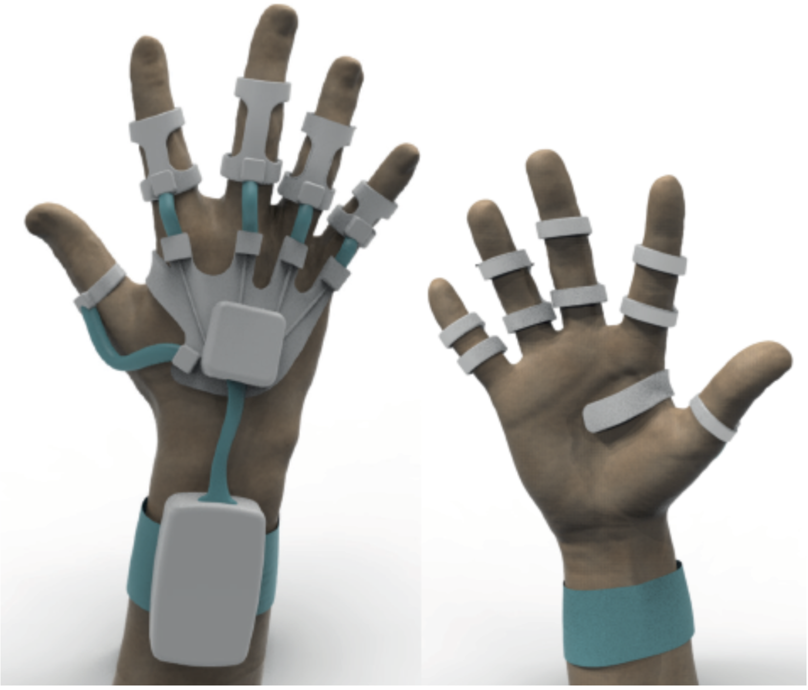


FIGURE 5.5: Second Render, using IMUs

Focusing of the arm part, bracelet's dimensions were excessively bulky, so we tried to reorganize the layout of the internal components, eliminating some parts that are redundant or superfluous as buttons, moreover soft materials, in combination with a rigid core allows usage of the glove on any type of arm. As a last point we worked on the variable thicknesses of the glove, which follows the internal hardware; in the end we opted for the stylistically "cleaner" solution, that is the alternative you can see in Figure 5.6.

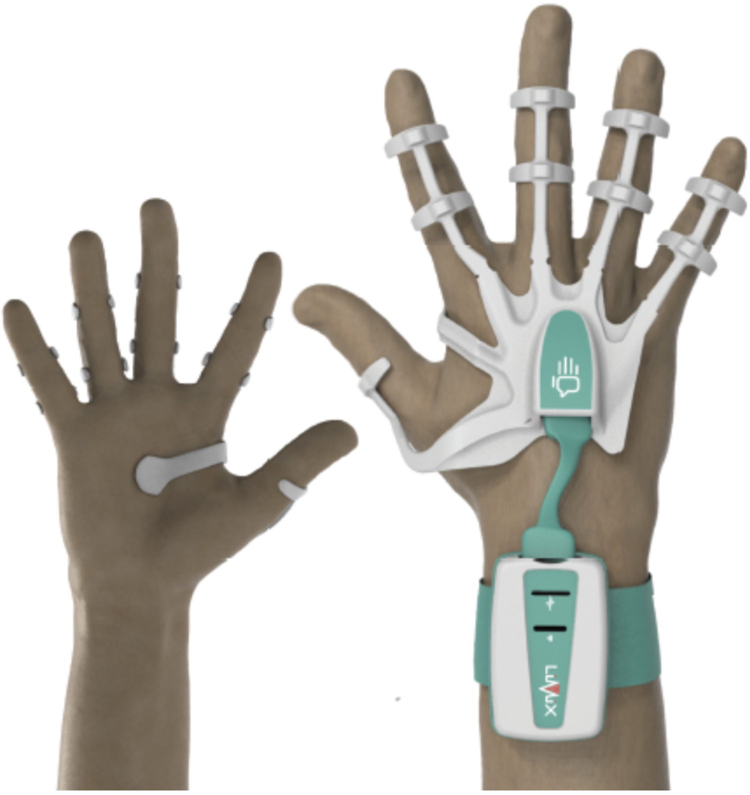


FIGURE 5.6: Last Render, using IMUs

Bracelet's previous version had significant dimensions, not optimized for the hardware inside. We worked on both the interaction and components to make the bracelet's body less bulky as possible. Buttons at the top have been eliminated and a 1100 mAh battery has been inserted, replacing the old 1800 mAh to recover volume; inside the layout of the components has been optimized to save more space. In addition, the part relating strap's attachment has been modified, recovering other space in height we have therefore gone from a footprint of 68 x 42 x 36 mm to a footprint of 68 x 42 x 19 mm. External dimensions are visible in Figure 5.7. About length and width of the bracelet, main constrain is the size of main electronic board.
To attach the arm part to user's forearm a watch band has been used, removing problem to produce also this part. A comparison between current prototype and new bracelet design is visible in Figure 5.8

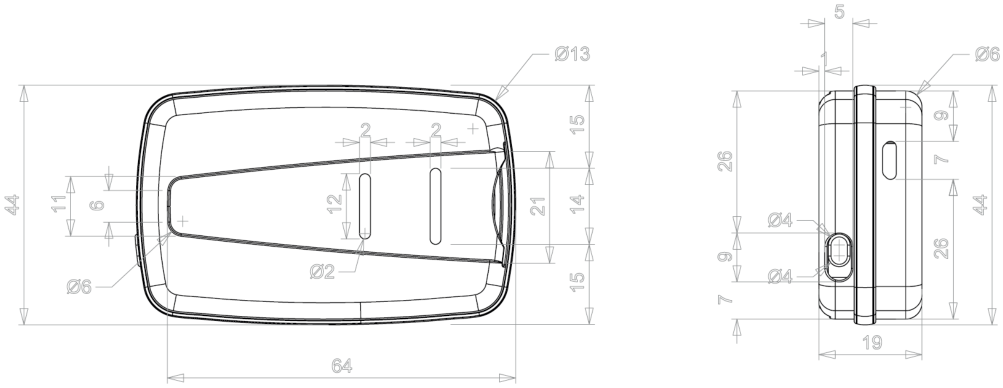


Figure 5.7: Bracelet Dimensions

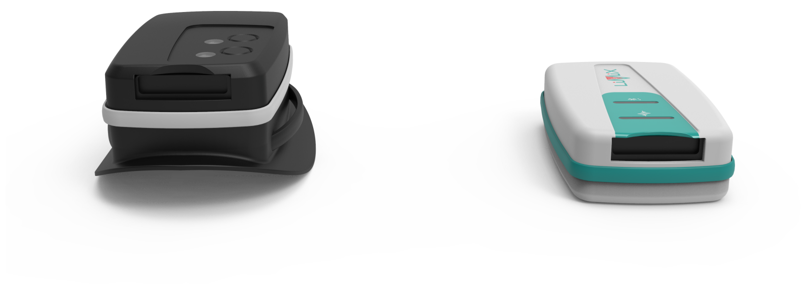


Figure 5.8: Bracelets Comparison (render)

Based on the ergonomic study of the hand, we hypothesized three sizes based on the percentiles of a man, a woman and a child. The three sizes will respectively cover a range that goes from the 5th percentile of a 6-year-old child to the 5th percentile of a woman (S) from the 5th percentile of a woman to the 50th percentile of a woman (M) and from the 50th percentile of a woman to the 95th percentile of the man (L), see Figure 5.9

A further study needs to be conducted about rings sizes; probably the best solution will be to create 2 or 3 sizes of silicon rings which may adapt to a wide range of fingers.

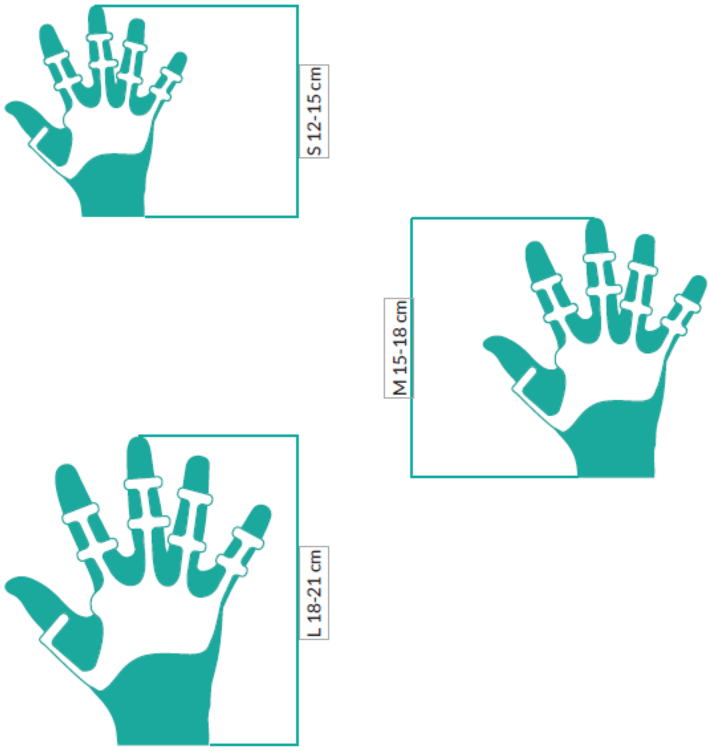


Figure 5.9: Talking Hands sizes

Chapter 6

Conclusion and future work

This thesis presented a work-flow for the translation of static and dynamic signs from a custom sensory glove, called Talking Hands. Different algorithms (static and dynamics) and classifiers have been tested, some of them, especially Random Forest and Neural Network, performed better than the other works in literature. This work shows that high performances of gesture recognitioncould be obtained without any external camera or position tracker, acquiring all the data from a wearable device. Indeed, even if explicit information about the position of the hand is not available, the data about the orientation allow a high translation accuracy. The presented results suggest that a completely wearable gesture recognitiondevice system can be possibly realized. Though some steps are still missing to achieve this goal. Indeed, it should be recalled that all the translation process for dynamic gesture recognition (described in Section 4.3) should be implemented on a smartphone application. Thanks to the powerful processors available today, the computation load is not an issue, and the main problem is represented by the communication channel, since all the data about the dynamic gestures should be sent to the phone in real time. For this reason, further studies will be carried out on compressing data, trying to maintain the BLE antenna or upgrade it to the next BLE generation standard. Otherwise, a WiFi module can be used to manage a huge amount of data. Several system tests must be done to ensure product usability. Those test will start with post-stroke patients with aphasia in collaboration with KOS-Care group.

Appendix A

Accelerometer as finger tracker?

This brief appendix reports results study of Analog Devices' ADXL362 accelerometer for the acquisition of data relating to finger flexion, understanding if ADXL362 or similar accelerometers are suitable as finger tracker for Talking Hands project. A simple Hardware Architecture used to conduct this experiment is shown in Figure A.1.

This experiment was conducted using two ADXL362 embedded in two different rings which can be seen in Figure A.2, one placed on the intermediate phalanx and the other one on the proximal phalanx.
Data about linear acceleration are acquired by a microcontroller, the microcontroller sends through serial communication (UART1152008N1) real time data to a PC, where data are stored.

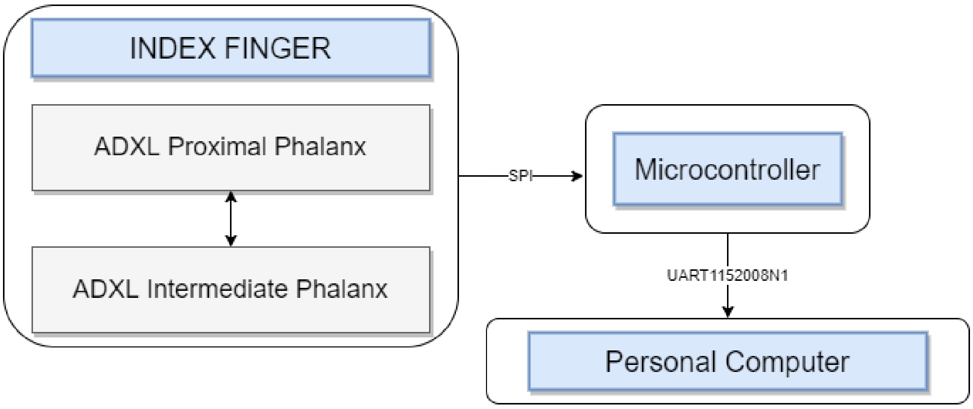


FIGURE A.1: Hardware Architecture of accelerometers as finger tracker

Figure A.3 shows the raw data of the accelerometer reading during the flexion of the fingers in three conditions (shown in A.2):

- Palm facing left, fingers up,
- Palm facing left, front fingers,
- Palm down.

During each phase of the experiment, three total flexions of the index finger has been performed at a very low speed.

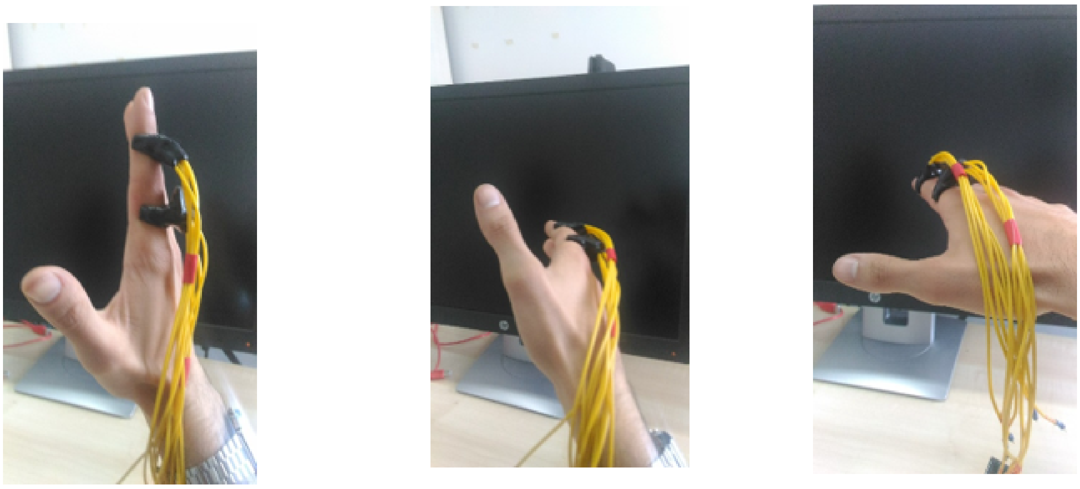


Figure A.2: Hand position in the experiment: (Left) Palm facing left, fingers up (Center) Palm facing left, front fingers, (Right) Palm down

This is not possible in the second condition: in the central part of the graph, or that relating to the second condition, accelerometer's values do not show significant changes during the three flexions. Flexion can not be recognized, Figure A.4 shows data about fingers' flexion in the critical case.

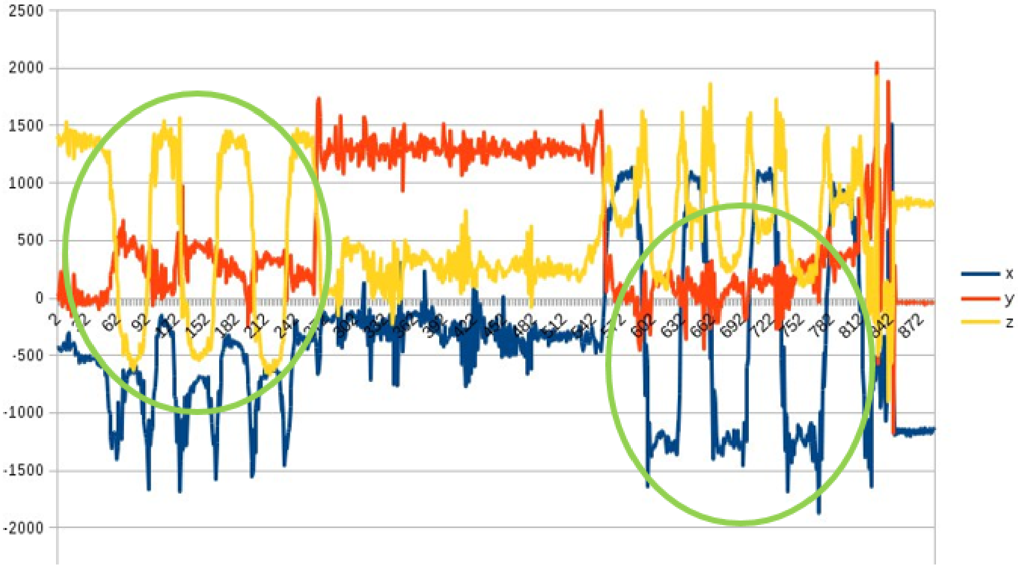


Figure A.3: Data sequence of the three conditions

The problem is that with hand configured as in Figure A.2 (Center) finger's flexion are almost ignored. In conclusion, results were not satisfactory. Certain finger's movements cannot be easily and precisely discriminated by using only accelerometers, otherwise ambiguous we can get ambiguous results, misunderstanding fingers' position.

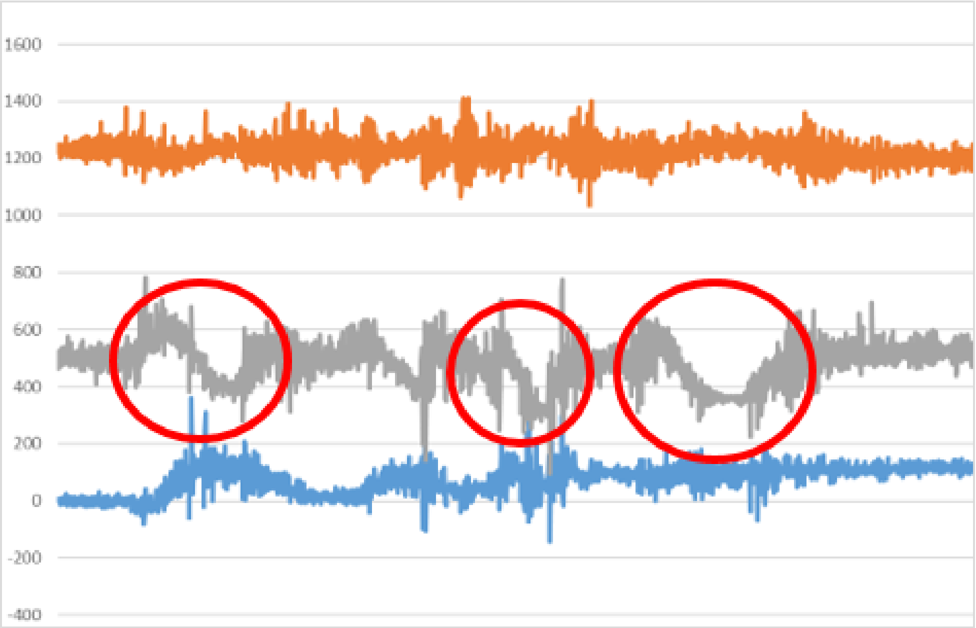


Figure A.4: Finger's flexion in the critical case

Appendix B

Gesture detection system within relative reference systems

Aim of the experiment:
This experiment tries to solve the problem of recognizing gestures of an user who drives a vehicle, just using an IMU. Considering the forearm as a rigid-link, an IMU will perfectly tracks forearm's orientation. Starting from the assumption that it is possible to recognize the orientation of any object that is moving within a reference system, but having another reference system, e.g. a person walking inside an bus, in this experiment we will recognize user's forearm movements while the user is driving a car.

Solution:
In this application Talking Hands system's arm part has been used, its architecture is described in Section 3.2. Meaningful gestures are expressed considering only the user's forearm orientation. Detecting the gestures of a user in a static position is possible, simply by exploiting the data relating to the orientation of the user's arm in space, as already explained in Chapter 4. This is no longer possible when the user is no longer static but he is moving, e.g. walking or driving a car.
Today in literature a solution which solves this problem just using Inertial Sensors like IMUs, accelerometers etc has not been found. This solution solves the problem by using only inertial measurement units. The proposed solution can be generalized to n-relative reference systems. To be clearer, let's explain this with a practical (but absurd) example : let's assume that the final user is driving a car inside a submarine. There are three relative reference systems:

- Submarine: which affects the movements of the car and the user;
- Car: which undergoes the movements of the submarine and influences those of the user.
- User: who undergoes the movements of the submarine and the car.

We have one IMU for each system: submarine; car; user. So we have also relevant data about systems' orientation. IMU on user's forearm will be embedded in a device called

AGIS (Automotive Gesture Interaction System). This device is visible in Figure B.1. It will be a simple bracelet the user can wear.

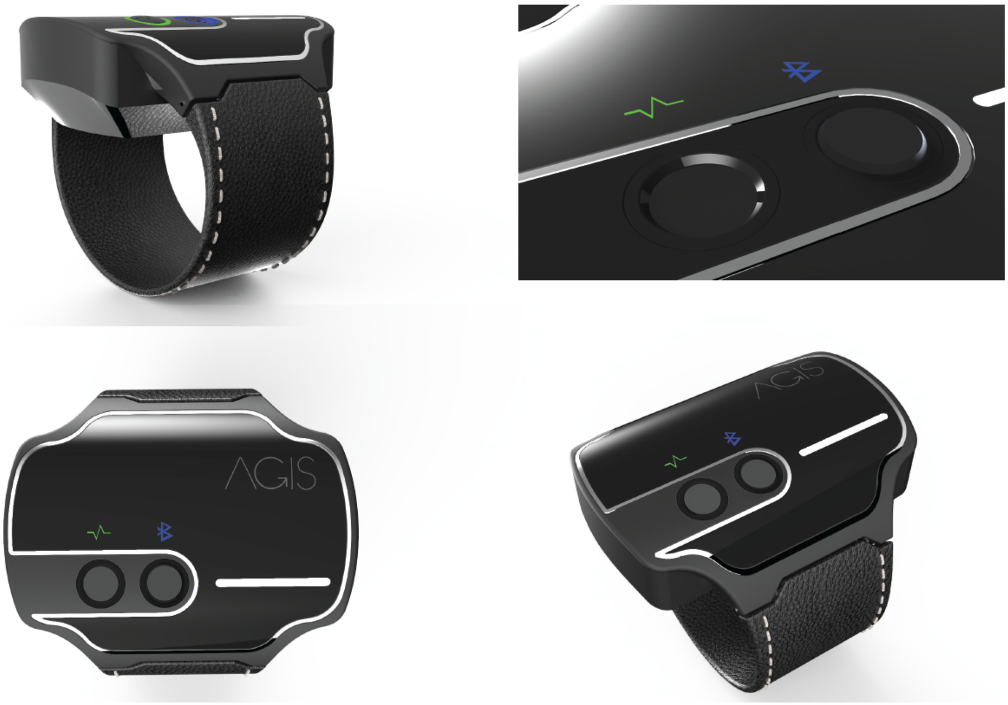


Figure B.1: AGIS render

For a better understanding refer to B.2. This reasoning can be extended to n-relative reference systems, it is not bounded at 3 different reference systems. In our experiment we tried with 2-relative reference system: a human inside a car. To recognize gestures in a car while driving, a real time correction is needed. Available data are:

- user's forearm orientation, thanks to the IMU placed on the user's forearm thanks to AGIS.

- car's orientation, placing an IMU placed on user's car thanks to a smartphone (which embed an IMU) on car's dashboard using a smartphone holder as shown in B.2.

IMU's output for our system are quaternions. According to mathematical theory, quaternions are the elements of a four-dimensional algebra on the field of real numbers. They constitute a vector space and a generic quaternion q is written as:

$$q = a\mathbf{1} + b\boldsymbol{i} + c\boldsymbol{j} + d\boldsymbol{k}$$

where a, b, c, d are real numbers, and **i**, **j**, **k** are symbols that can be interpreted as unit-vectors pointing along the three spatial axes. The canonical basis for this vector space is:

$$\mathbf{1} := <1,0,0,0>, \ \boldsymbol{i} := <0,1,0,0>, \ \boldsymbol{j} := <0,0,1,0> \text{ and } \boldsymbol{k} := <0,0,0,1>$$

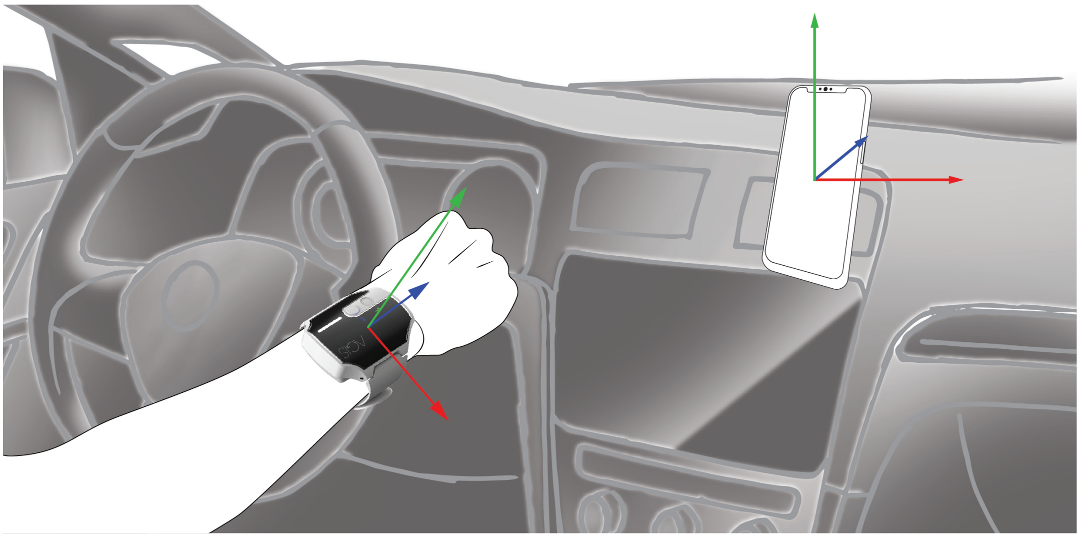


FIGURE B.2: Example of user's and car's reference system

Given a quaternion q its conjugate is defined as: $q^* = a - b\boldsymbol{i} - c\boldsymbol{j} - d\boldsymbol{k}$
The norm of a quaternion whose norm is:

$$\|q\| = \sqrt{qq^*} = \sqrt{a^2 + b^2 + c^2 + d^2}$$

A unitary (or normalized) quaternion is defined as a quaternion that has norm equal to 1.
The inverse of a quaternion q is: $q^{-1} = \frac{q^*}{\|q\|}$.

Any rotation in three dimensions can be represented as a combination of an axis and an angle of rotation. Quaternions represent a simple way to encode this axis-angle representation in four numbers and apply the corresponding rotation to a position vector representing a point relative to the origin in $\mathbb{R}^{\mathbf{3}}$.

A quaternion rotation can be represented using the formula:

$$q = \; e^{\frac{1}{2}\vartheta(bi+cj+dk)} = \cos\tfrac{1}{2}\vartheta - (bi + cj + dk)\sin\tfrac{1}{2}\vartheta$$

where ϑ is the rotation angle and the vector $< b, c, d >$ represents the rotation axis.

When the user's IMU is turned on, IMU starts to detect its orientation, returning quaternion that indicate the axis and the angle of rotation respect to the reference. With the user's IMU positioned on the wrist, changes in the orientation of the user's forearm will be detected. Remembering that if the user walks, moves, or even just rotates on himself, the orientation detected by the IMU will obviously change.

Using a matching method between quaternions, a gesture recognition algorithm has been developed starting from the algorithm described in 4.1. From Talking Hands we bring also the strategy to reset orientation every time the user performs a body rotation, as described in Section 3.8.1. In fact, the same gesture repeated with the body rotated generates different quaternions, which if compared with those memorized for the associated gesture lead to not detecting the gesture, despite the movement made by the user being the same. This happens because external rotation is applied also to the user's arm. For example if I am standing and wearing AGIS with arm parallel to the ground and y axes of AGIS's IMU perpendicular to the ground, if I apply a rotation to my body about 180 degree, this means AGIS's IMU will receive also a rotation of 180 degree on the y axes.

There are two possible approaches to this type of problem. The simplest and most direct one involves the use of a button that, connected to AGIS and pressed by the user, resets the reference system every time the body rotates with respect to the initial reference system of the bracelet. Same as Talking Hands application (3.8.1).

Inside a car, previous solution is not advisable: while driving, in every curve, climb or descent the car will change the orientation of the user compromising the recognition algorithm. This means the user must care more to reset the button then driving.

Using a second IMU we can address this problem. In fact, a second IMU integrated in the car provides us information on the orientation of the car system (using an IMU placed on a smartphone as in B.2), while the AGIS's IMU give us information about user's movements.

With car's quaternions, we can calculate the reverse rotation to correct user's orientation reference system. By doing this we obtain quaternions representing the user's movements independently from the car's movements.
In this way the matching is effective and the recognition of gestures occurs without the user bothering to make these corrections clicking a reset button. This correction is totally autonomous.

Since IMU's reading is continue, it will not refer to the instant in time, so let q_C quaternion sent by the car and q_A sent by AGIS in the same instant.
Lets refer to q_C^r e q_A^r as quaternions which represents the two reference systems, quaternions' values when IMUs are turned on (car and AGIS), we can call these quaternions also initial frames.

The resulting rotation represented multiplying the two quaternions is equivalent to the succession of the two rotations of the multiplied quaternions: in fact, by multiplying a quaternion by its inverse we obtain the unit vector, which represents the rotation of zero angle, therefore the absence of rotation.

In this way $q_{Cc} = {q_C^r}^{-1} q_C$ represents current car's orientation corrected respect to the reference car's initial frame. In fact if the car has not been moved since starting the system, we have:
$q_C = q_C^r \implies {q_C^r}^{-1} q_C = \mathbf{1}$.
If we suppose that the user did not move the arm yet, quaternion q_{Cc} represents also AGIS's rotation since AGIS is inside the car anchored to user's arm.

Lets calculate the inverse of q_{Cc}:
${q_{Cc}}^{-1}$: it is the quaternion which will correct AGIS's orientation.
$q_{Ac} = {q_{Cc}}^{-1} q_A$ is the quaternion that indicates AGIS's orientation respect to q^r_A, independently from car's movements.

Quaternion q_{Ac} is the data used in the gesture recognition algorithm.

This gesture recognition algorithm has been implemented to simplify the user experience inside a smart car. After connecting the device to a smartphone, the user can forget his smartphone and control all multimedia and telephone commands using gestures. The algorithm that compensates car's changing orientation allows the user to answer a phone call just moving his wrist as in a predefined gesture, or controlling the multimedia system's volume while changing music tracks. This experiment has been conducted in a real situation, 5 different gestures where recorded: volume up; volume down; answer/end call; next track; previous track. In real test situations 99% of accuracy has been reached about gesture recognition.

Appendix C

List of Publication

Articles referring to Talking Hands are: [84], where hardware is described together with data about its translation efficiency with reference to static gestures, while preliminary results about translation of dynamic gestures have been presented in [85].
Two articles were published also in medical journal, in a short abstract published for Life Span and Disability Journal [86] there is a short description of Talking Hands-for non verbal communicators. [19] is focused on Talking Hands helping people with non-verbal Autism Spectrum Disorder. They must use an augmentative and alternative communication (AAC) system that helps with ordinary speech, such as picture exchange communication systems, text systems, and voice output devices. Sign language (also a form of AAC) has been recommended for people with ASD because it has many advantages as a communication system. In [19] we describe a possible impact of Talking Hands for people with ASD. We analyse the following advantage: Talking Hands could increase the possibility to talk naturally after using Talking Hands as therapy.
Last publication is about Talking Hands used for dynamic gesture recognition, including also experimental results shown in section 4.4, comparing different machine learning classifiers and discussing their performances both in terms of translation accuracy and computational time. This word has been published in [87].

Bibliography

[1] G. Pizer, K. Walters, and R. P. Meier. “We Communicated That Way for a Reason”: Language Practices and Language Ideologies Among Hearing Adults Whose Parents Are Deaf”. In: *The Journal of Deaf Studies and Deaf Education* vol. 18, (1 2013), pp. 75–92.

[2] J. F. Alexa Kuenburg Paul Fellinger. “Health Care Access Among Deaf People”. In: *The Journal of Deaf Studies and Deaf Education* vol. 21, (1 2016), pp. 1–10.

[3] S. J. E. uonion of Deaf. *Deaf and Employment in crisis*. 2013. URL: `https://www.eud.eu/news/deaf-and-employment-crisis` (visited on 07/03/2013).

[4] N. Gale. “D/Deaf and disabled trans experiences in Europe-”. In: *Transgender Europe, November 2017* (2017). URL: `https://tgeu.org/wp-content/uploads/2018/02/Oppression-Squared.pdf`.

[5] V. de Andrade. *Deafness carries a huge cost burden: economic as well as personal*. 2017. URL: `https://www.wits.ac.za/news/latest-news/in-their-own-words/2017/2017-03/deafness-carries-a-huge-cost-burden-economic-as-well-as-personal.html` (visited on 03/03/2017).

[6] asha.org. *Speech Sound Disorders-Articulation and Phonology*. 2013. URL: `https://www.asha.org/PRPSpecificTopic.aspx?folderid=8589935321&section=Incidence_and_Prevalence`.

[7] M. Vidović, O. Sinanović, L. Sabaskić, A. Haticić, and E. Brkić. “Incidence and types of speech disorders in stroke patients”. In: (2011). URL: `https://pubmed.ncbi.nlm.nih.gov/22649878/`.

[8] C. G. Liselotte Kjellmer Elisabeth Fernell and F. Norrelgen. In: (2018). URL: `https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6157542/`.

[9] E. M. Wilson, L. Abbeduto, S. M. Camarata, and L. D. Shribergd. “Estimates of the prevalence of speech and motor speech disorders in adolescents with Down syndrome”. In: (2019). URL: `https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6604065/`.

[10] R. P. M. H. J. P. Tanja Makkonen Hanna Ruottinen. “Speech deterioration in amyotrophic lateral sclerosis (ALS) after manifestation of bulbar symptoms”. In: (2018). URL: `https://pubmed.ncbi.nlm.nih.gov/29159848/`.

[11] G. Conti-Ramsden, K. Durkin, U. Toseeb, N. Botting, and A. Pickles. “Education and employment outcomes of young adults with a history of developmental language disorder”. In: (2018). URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5873379/.

[12] M. Stransky and M. Morris. “Adults with Communication Disabilities Face Health Care Obstacles”. In: (2018). URL: https://doi.org/10.1044/leader.FTR1.24032019.46.

[13] M. McKee, M. L. Stransky, and A. Reichard. “Hearing loss and associated medical conditions among individuals 65 years and older”. In: (2018). URL: https://www.sciencedirect.com/science/article/pii/S1936657417301176.

[14] G. Bartlett, R. Blais, and R. T. andreaserrani Brenda MacGibbon. “Impact of patient communication problems on the risk of adverse events in acute care settings”. In: (2008). URL: https://www.cmaj.ca/content/178/12/1555.

[15] U. E. P. K. Michael Rudolph Frank Rosanowski. “Anxiety and depression in mothers of speech impaired children”. In: (2003). URL: https://www.sciencedirect.com/science/article/pii/S0165587603003033?via\%3Dihub.

[16] E. D. Forum. *Disability and social exclusion in the European Union – Final Study Report.* 2018. URL: https://sid.usal.es/idocs/F8/FDO7040/disability_and_social_exclusion_report.pdf.

[17] who.int. *Assistive devices and technologies.* URL: https://www.who.int/disabilities/technology/en/.

[18] asha.org. *Augmentative and Alternative Communication (AAC).* URL: https://www.asha.org/public/speech/disorders/aac/.

[19] F. Pezzuoli, D. Tafaro, M. Pane, D. Corona, and L. Corradini. “Development of a New Sign Language Translation System for People with Autism Spectrum Disorder”. In: *Advances in Neurodevelopmental Disorders* (2020).

[20] A. M. Aktham, Z. B. Bahaa, Z. A. Alaa, S. M. Maher, and M. M. B. Lakulu. “A review on systems-based sensory gloves for sign language recognition state of the art between 2007 and 2017”. In: *Sensors (Switzerland)* vol. 18, no. 7, (2018). ISSN: 14248220.

[21] R. Z. Khan and N. A. Ibraheem. “Hand Gesture Recognition: A Literature Review”. In: *International Journal of Artificial Intelligence & Applications* vol. 3, no. 4, (2012), pp. 161 –174.

[22] M. Sushmita and A. Tinku. “Gesture recognition: A survey”. In: *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews* vol. 37, (2007), pp. 311 –324. ISSN: 10946977. arXiv: 9809069v1 [arXiv:gr-qc].

[23] O. S. C. W and R. S. “Automatic Sign Language Analysis: A Survey and the Future beyond Lexical Meaning”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* vol. 27, no. 6, (2005).

[24] V. C. “American Sign Language Recognition: Reducing the Complexity of the Task with Phoneme-based Modeling and Parallel Hidden Markov Models”. PhD thesis. University of Pennsylvania, 2003.

[25] G. Nuwan, K. Y. Chow, A. Rini, and D. Serge. “Gaussian Process Dynamical Models for hand gesture interpretation in Sign Language”. In: *Pattern Recognition Letters* vol. 32, (15 2011), pp. 2009–2014.

[26] K. Daniel, M. John, and M. Charles. “A person independent system for recognition of hand postures used in sign language”. In: *Pattern Recognition Letters* vol. 31, (2010), pp. 1359–1368. ISSN: 01678655.

[27] K. Pradeep, G. Himaanshu, P. P. Roy, and D. D. Prosad. “Coupled HMM-based multi-sensor data fusion for sign language recognition”. In: *Pattern Recognition Letters* vol. 86, (2017), pp. 1–8.

[28] H. Junwei, A. George, and S. Alistair. “Modelling and segmenting subunits for sign language recognition based on hand motion analysis”. In: *Pattern Recognition Letters* vol. 30, (6 2009), pp. 623–633.

[29] Y. H. Sub, S. Jung, B. Y. J., and S. Hyun. “Hand gesture recognition using combined features of location, angle and velocity”. In: *Pattern Recognition* vol. 37, no. 4, (2001), pp. 1491–1501. ISSN: 00313203.

[30] Z. M. M. and S. S. I. “Sign language recognition using a combination of new vision based features”. In: *Pattern Recognition Letters* vol. 32, (4 2011), pp. 572–577.

[31] M. Zbakh, Z. Haddad, and J. L. Krahe. “An online reversed French Sign Language dictionary based on a learning approach for signs classification”. In: *Pattern Recognition Letters* vol. 67, (2015), pp. 28–38.

[32] N. H. Davi and Z. Cleber. “Gesture recognition: A review focusing on sign language in a mobile context”. In: *Expert Systems with Applications* vol. 103, (2018), pp. 159 –183. ISSN: 09574174.

[33] Cooper, Pugeault, and Bowden. “Reading the signs: A video based sign dictionary”. In: *Proceedings of the IEEE International Conference on Computer Vision* (2011), pp. 914–919.

[34] S. Thad, W. Joshua, and P. Alex. “Real Time American Sign Language Recognition using Desk and Wearable Computer Based Video”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* vol. 20, no. 12, (1998), pp. 1371 –1375.

[35] SignAll. *SignAll website.* URL: `https://www.signall.us` (visited on 08/10/2020).

[36] *Gesture and Sign language in Human-Computer Interaction.* Bielefel, Germany: Springer, 1997.

[37] R. A. Bolt. “Put$_That - There : VoiceandGestureatGraphicsInterface$”. In: (1980). URL: `https://www.youtube.com/watch?v=RyBEUyEtxQo`.

[38] P. A. Harling and A. D. Edwards. “Hand Tension as a Gesture Segmentation Cue”. In: (1996).

[39] M. Boulares and M. Jemni. “Automatic hand motion analysis for the sign language space management”. In: *Pattern Analysis and Applications* vol. 22, (2 2019), pp. 311–341.

[40] I. Elouariachi, R. Benouini, K. Zenkouar, and A. Zarghili. “Robust hand gesture recognition system based on a new set of quaternion Tchebichef moment invariants”. In: *Pattern Analysis and Applications* (2020).

[41] M. Fagiani, E. Principi, S. Squartini, and F. Piazza. “Signer independent isolated Italian sign recognition based on hidden Markov models”. In: *Pattern Analysis and Applications* vol. 18, (2 2014), pp. 385–402.

[42] M. Zadghorban and M. Nahvi. “An algorithm on sign words extraction and recognition of continuous Persian sign language based on motion and shape features of hands”. In: *Pattern Analysis and Applications* vol. 21, (2 2018), pp. 323–335.

[43] X. Zabulisy, H. Baltzakisy, and A. Argyroszy. “Vision-based Hand Gesture Recognition for Human-Computer Interaction”. In: ().

[44] A. Thalange and S. Dixit. “ASL Number Recognition using Open-finger Distance Feature Measurement Techniquen”. In: ().

[45] P. Kumar, H. Gauba, P. P. Roy, and D. P. Dogra. “A Multimodal Framework for Sensor based Sign Language Recognition”. In: *Neurocomputing (2017)* (2017).

[46] T. G. Zimmerman, J. Lanier, C. Blanchard, S. Bryson, and Y. Harvill. “A HAND GESTURE INTERFACE DEVICE”. In: *VPL Research, Inc. 656 Bair Island Road, Suite 304 Redwood City, CA 94063.* 1987.

[47] *CH Robotics Error Estimating Position and Velocity from acceleration.* URL: `http://www.chrobotics.com/library/accel-position-velocity`.

[48] C. Tushar, P. Ankit, V. Anvesh, and M Sameer. “Smart Glove With Gesture Recognition Ability For The Hearing And Speech Impaired”. In: Sept. 2014.

[49] N. Praveen, N. Karanth, and M Megha. “Sign language interpreter using a smart glove”. In: Oct. 2014, pp. 1–5.

[50] R. Nuwer. “Armband adds a twitch to gesture control”. In: *New Scientist* vol. 217, no. 2906, (2013), p. 21. ISSN: 0262-4079. URL: `http://www.sciencedirect.com/science/article/pii/S0262407913605424`.

[51] Bishop and C. M. *Pattern Recognition and Machine Learning.* Springer, 2006.

[52] S. Russel and P. Norving. *Artificial Intelligence, a modern approach.* Pearson, 2010.

[53] J. Gałka, M. Mąsior, M. Zaborski, and K. Barczewska. “Inertial Motion Sensing Glove for Sign Language Gesture Acquisition and Recognition”. In: *IEEE Sensors Journal* vol. 16, no. 16, (2016), pp. 6310–6316.

[54] J. Ren. “ANN vs. SVM: Which one performs better in classification of MCCs in mammogram imaging”. In: *Knowledge-Based Systems* vol. 26, (2012), pp. 144 –153. ISSN: 0950-7051. URL: `http://www.sciencedirect.com/science/article/pii/S095070511100164X`.

[55] B. Dhananjai, P. Uddaish, S. Gaurav, and S. Nitin. “Two way wireless data communication and American sign language translator glove for images text and speech display on mobile phone”. In: *5th International Conference on Communication Systems and Network Technologies, CSNT 2015.* 2015, pp. 578–585. ISBN: 9781479917976.

[56] B. JanFizza, R. Maryam, M. S. Ishtiaq, K. A. M, and S. Ahmad. “American Sign Language Translation through Sensory Glove: SignSpeak”. In: *International Journal of u- and e-Service, Science and Technology* vol. 8, (2015), pp. 131–142.

[57] S. Michael and T. Mohohlo. “A mobile application for South African Sign Language (SASL) recognition”. In: *IEEE AFRICON 2015.* 2015, pp. 1–5.

[58] D. Suraksha and D. Suman. “Low cost tangible glove for translating sign gestures to speech and text in Hindi language”. In: *3rd IEEE International Conference on Computational Intelligence & Communication Technology (CICT).* 2017, pp. 1–5. ISBN: 9781509062188.

[59] K. L. Jen, S. W. Lin, Y. P. Ju, and W. S. Jhan. “A real-time portable sign language translation system”. In: *Midwest Symposium on Circuits and Systems.* no. 1, . 2015, pp. 3–6.

[60] S. A. Zaki, M. M. Fahmi, J. M. Herman, F. B. Ali Ibrahim, M. F. Asyraf, and M. B. B. Bahar. “A New Data Glove Approach for Malaysian Sign Language Detection”. In: *IEEE Int. Symp. On Robotics and intelligent sensors (IRIS).* Elsevier Masson SAS, pp. 60–67. ISBN: 6065552305.

[61] K. V. E. and H. L. J. “Sign language recognition using intrinsic-mode sample entropy on sEMG and accelerometer data”. In: *IEEE Transactions on Biomedical Engineering* vol. 56, no. 12, (2009), pp. 2879–2890. ISSN: 00189294.

[62] H. Amoud, H. Snoussi, D. Hewson, M. Doussot, and J. Duchene. “Intrinsic Mode Entropy for Nonlinear Discriminant Analysis”. In: *IEEE Signal Processing Letters* vol. 14, no. 5, (2007), pp. 297–300. ISSN: 1558-2361.

[63] O. Cemil and L. M. C. “American Sign Language word recognition with a sensory glove using artificial neural networks”. In: *Engineering ApplicationsofArtificial-Intelligence* vol. 24, (2011), pp. 1204–1213.

[64] R. David, R. F. B., and V. Pablo. “Extending the bioinspired hierarchical temporal memory paradigm for sign language recognition”. In: *Neurocomputing* vol. 79, (2012), pp. 75–86. ISSN: 09252312.

[65] G. Dileep and H. Jeff. “Towards a mathematical theory of cortical micro-circuits”. In: *Plos Computational Biology* vol. 5, (10 2009).

[66] *Bluetooth Generic Attributes.* URL: https://www.bluetooth.com/specifications/gatt/.

[67] A. K., R. E., Y. T., A. Y., and E. L. Carpentier. "A Testing System for a Real-Time Gesture Classification Using Surface EMG". In: *20th IFAC World Congress.* 2017, pp. 11498–11503.

[68] G. Saggio, F. Riillo, L. Sbernini, and L. R. Quitadamo. "Resistive flex sensors: a survey". In: *Smart Materials and Structures* vol. 25, no. 1, (2015), p. 013001.

[69] Z. Zhihao, C. Kyle, L. Xiaoshi, Z. Songlin, W. Yufen, Z. Yihao, M. Keyu, S. Chenchen, H. Qiang, F. Wenjing, F. Endong, L. Zhiwei, T. Xulong, D. Weili, Y. Jin, and C. Jun. "Sign-to-speech translation using machine-learning-assisted stretchable sensor arrays". In: *Nature Electronics* vol. 3, (2020), pp. 571–578.

[70] *A Guide To using IMU (Accelerometer and Gyroscope Devices) in Embedded Applications.* URL: http://www.starlino.com/imuguide.html.

[71] *Description of IMU aiding from Roll isolated Gyro.* URL: http://www.starlino.com/imuguide.html.

[72] *Inertial Navigation: 40 Years of Evolution.* URL: http://www.imar-navigation.de.

[73] *Three Axis IMU.* URL: http://www.mathworks.com/access/helpdesk/help/toolbox/aeroblks/index.html?/access/helpdesk/help/toolbox/aeroblks/threeaxisinertialmeasurementunit.html.

[74] M. Jacob, W. Eric, and L. Sung. "Thermal and Crosstalk-Aware Physical Design for 3D System-On-Package". In: *Proceedings - Electronic Components and Technology Conference* vol. 1, (Jan. 2005).

[75] K. Mourakami and H. Taguchi. "Gesture Recognition using Recurrent Neural Networks". In: *ACM Conference on Human factors in computing systems: reaching through technology.* 1999, pp. 237 –242.

[76] R. H. Liang and M. Ouhyoung. "A real-time continuous gesture recognition system for sign language". In: *Proceedings of the 3rd IEEE International Conference on Automatic Face and Gesture Recognition.* 1998, pp. 558–567. ISBN: 0818683449.

[77] Huynh and D. Q. "Metrics for 3D rotations: Comparison and analysis". In: *Journal of Mathematical Imaging and Vision* vol. 35, (2009), pp. 155–164. ISSN: 09249907.

[78] C. Corinna and V. Vladimir. "Support-vector networks". In: *Machine Learning* vol. 20, (3 1995), pp. 273–297.

[79] Breiman and Leo. "Random Forests". In: *Machine Learning* vol. 45, (1 2001), pp. 5–32.

[80] H. Trevor, T. Rober, and F. Jerome. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction.* New York, NY: Springer, 2009.

[81] G. John and P. Langley. “Estimating Continuous Distributions in Bayesian Classifiers”. In: *In Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*. Morgan Kaufmann, 1995, pp. 338–345.

[82] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. “Scikit-learn: Machine Learning in Python”. In: *Journal of Machine Learning Research* vol. 12, (2011), pp. 2825–2830.

[83] K. andMohammed Waleed. *Australian Sign Language signs (High Quality) Data Set*. UCI Machine Learning Repository. 2002. URL: `http://archive.ics.uci.edu/ml/datasets`.

[84] P. Francesco, C. Dario, C. M. Letizia, and C. Andrea. “Development of a Wearable Device for Sign Language Translation”. In: *Int. Workshop on Human-Friendly Robotics (HFR2017)*. Ed. by F. Ficuciello, F. Ruggiero, and A. Finzi. Cham: Springer International Publishing, 2017, pp. 115–126. ISBN: 978-3-319-89327-3.

[85] P. Francesco, C. Dario, and C. M. Letizia. “Improvements in a Wearable Device for Sign Language Translation”. In: *Advances in Intelligent Systems and Computing*. Ed. by S. Verlag. Cham: Springer International Publishing, 2020, pp. 70–81.

[86] P. Francesco, C. Dario, C. M. Letizia, and C. Athos. “Talking Hands: a data-glove that helps non-verbal forms of communication”. In: *Assistive Technology and Disabilities Conference, Rome, Italy*. 2019.

[87] F. Pezzuoli, D. Corona, and M. L. Corradini. “Recognition and classification of dynamic hand gestures by a wearable data-glove”. In: *SN Computer Science Original Research in Pattern Recognition* (2020).