Politecnico di Torino

FACOLTÀ DI INGEGNERIA
Corso di Laurea Magistrale in Mechatronic Engineering

MASTER'S DEGREE THESIS

Hand gestures classification with a thin film stretchable electronic skin

fill

Candidato:
Matteo Virone
Matricola 230932

Relatore:
Prof. Marcello Chiaberge

Correlatore:
Dott. Mahmoud Tavakoli

Anno Accademico 2017/2018

fill

Sommario

Il riconoscimento dei gesti della mano si è espanso notevolmente nelle ultime decadi, soprattutto per quanto riguarda l'interfaccia uomo-macchina, questo grazie agli sviluppi riguardanti le tecnologie meccatroniche necessarie a realizzarlo. Numerosi dispositivi di classificazione dei gesti della mano sono stati sviluppati ultimamente, con prestazioni soddisfacienti in termini di affidabilità e varietà nel riconoscimento dei gesti della mano. Tuttavia, questi prodotti presentano degli svantaggi significativi: un costo elevato ed una complessità eccessiva, mentre l'obiettivo del progetto corrente è la realizzazione di un sistema basato su un sensore capacitivo a basso costo e di facile uso. In questa tesi, per ottenere le condizioni descritte sopra, sono stati adottati un largo insieme di caratteristiche estratte (features extraction set) ed una macchina a vettori di supporto come classificatore (support vector machine: SVM).Il sistema così realizzato risulta essere in grado di riconoscere fino a 5 gesti della mano con soddisfacente affidabilità: chiusura, apertura, zoom-in, zoom-out e "ok". Questi sono solo un esempio di come si possano aggiungere gesti utili per migliorare l'interfaccia uomo-macchina. Inoltre è stato aggiunto un sesto gesto, il quale permette di bloccare il sistema con una "combinazione" scelta dall'utilizzatore e quindi evitare usi indesiderati del sistema.

Abstract

The hand gestures recognition has expanded considerably in recent decades, especially with regard to the human-machine interface, thanks to the developments concerning mechatronic technologies necessary to achieve it. Numerous devices to classify hand gestures have been developed lately, with satisfactory performances in terms of reliability and variety in the recognition of hand gestures. However, these products have significant disadvantages: a high cost and an excessive complexity, while the objective of the current project is the realization of a system based on a low cost capacitive sensor. The hardware used in this project is composed of a multi-layer printed electronics circuit which forms a mesh of sensors composed of conductive rows and columns. Such mesh allows detection of the human hand proximity by each of the rows and columns. Such hardware is then used along with a SVM based classification algorithm in order to detect five hand gestures, including closing, opening, zoom-in, zoom-out and "ok". Results shows that this combination is able to classify these gestures with an overall classification accuracy of 91%, measured during 3 sessions with 3 different users.These are just some examples of how it is possible to add useful gestures to improve the human-machine interface. In addition, a sixth gesture has been added, which allows the system to be locked with

a "combination" chosen by the user and therefore to avoid unwanted use of the system.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This chapter introduces the contents of this dissertation, providing the reasons that have led to this project, and giving the objectives to be achieved.

1.1 Overview and Motivations

Hand gesture has been one of the most common and natural communication media among human being.

The hand gesture, during daily life, is a natural communication method mostly used only among people who have some difficulty in speaking or hearing.

Gesture is also a symbol of physical behavior or emotional expression. It includes body gesture and hand gesture. It falls into two categories: static gesture [4] [5] [6] [7] and dynamic gesture [8] [9] [10] [11]. For the former, the posture of the body or the gesture of the hand denotes a sign. For the latter, the movement of the body or the hand conveys some messages.

Reffering to the recent advances and interest in alternative methods for HMI, gesture recognition can be also used as a tool of communication between computer and human [12] [13] [14]. It is greatly different from the traditional hardware based methods and can accomplish human-computer interaction through gesture recognition. Gesture recognition determines the user intent through the recognition of the gesture or movement of the body or body parts.Gesture recognition has become a hot topic for decades. In the past decades, many researchers have strived to improve the hand gesture recognition technology. To do so, researchers have focused on development of appropriate hardware, such as armbands, for detection and classification of gestures.

Hand gesture recognition research has gained a lot of attention because of its applications for interactive human-machine interface and virtual environments.

With the use of hand gesture recognition systems, people can interact with computers in a more intuitive mode. Hand gesture recognition owns wide applications in sign language recognition [15] [16] [17] [18], computer games [19] [20], virtual reality [21] [22] and HCI systems [23] [24]. There were numerous gesture recognition methods established for tracking and recognizing numerous hand gestures. Each one of them has their advantage and disadvantage.

Wearable hardware is one of these, in which in order to interface with the computer system, users need to wear them. The best example for the wearable technology is the instrumented gloves. These electronic gloves have some sensors, and thanks to these sensors they provide information related to location of the hand, position orientation of fingers etc. Output results of data gloves are good but they are expensive [6].

The most recent ones are the optical markers. The optical markers detect the location of hand or tips of fingers by projecting Infra-Red light and reflect this light on screen. These systems also offer a worthy result but need a very complex configuration. Nowadays some new methods have been proposed for hand gesture recognition, such as Image based systems which needs processing of image structures like texture, color etc. The approaches on optical markers are expensive and have very difficult configuration [6]. Also the technique based on image processing is vulnerable to diverse illumination situation, color texture modifying, which leads to variations in observed outcomes [25]. However, the performance of gesture recognition directly based on the features extracted by image processing is relatively limited. Recently, there have been an increasing number of gesture recognition research using vision-based methods. Although the performance has improved as the appearance of advanced sensors, like Microsoft Kinect sensors, the relatively higher price of such devices is still an obstacle to the large-scale application of gesture-based HCI systems. They are also bulky for the time being and cannot be integrated for instance in a touchpad. Besides, such advanced sensors perform even more unreliably than optical cameras in some certain environment. For instance, the attenuation of infrared ray in water could largely limit the use of those like Microsoft Kinect sensors in water with a good light condition.

The most common is the touchpad, which in recent years has not been improved, remaining able only to detect the touch. Consider how with a thin film of capacitive sensor is possible to implement, in addition to the touchpad, other functions, such as gesture recognition, without the need to touch it.

The nature of gesture recognition is a classification problem.

There are lots of approaches to handle 2D gesture recognition, including the orientation histogram [26], the hidden Markov model [27], particle filtering [28], support vector machine (SVM) [29], etc. [30] [31]. Most of those approaches need preprocessing the input gesture image to extract features. The performance of those approaches depends a lot on the feature learning process.

This dissertation is focused on a hand gestures recognition through another type of sensor, capacitive sensor. Our system shown in Figure 1.1 is composed by the capacitive sensor and the board to send the data.

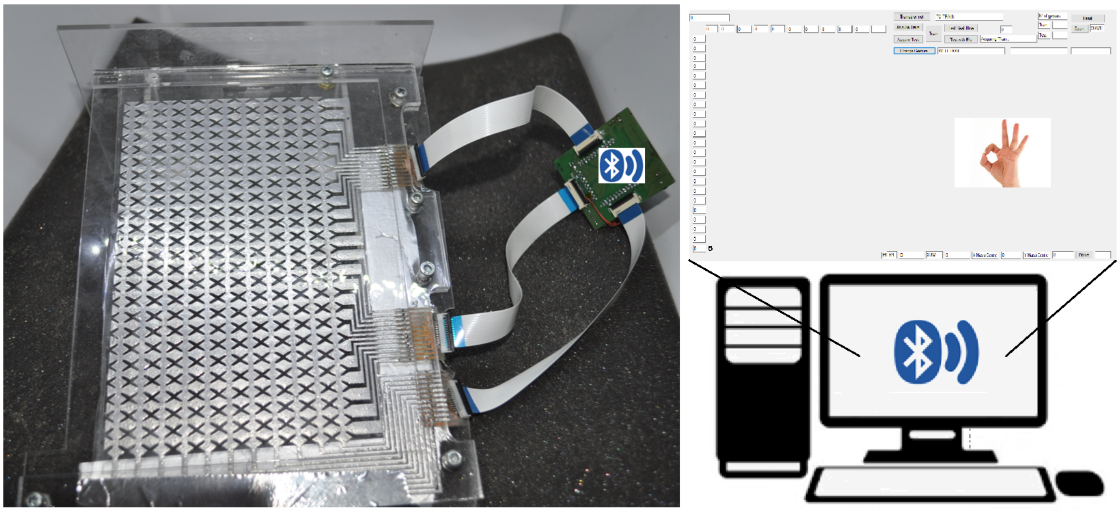


Figure 1.1: System

Capacitive proximity sensors allow not just the detection, but a distance estimation of conductive, grounded objects, such as the human body. In the past, this property has already been used to create devices that can estimate the position of one or more hands [32] [33], or the posture of a person on different pieces of furniture [34] [35]. A distinct advantage of capacitive proximity sensors is their ability to detect objects without being disturbed by nonconductive materials. They can be installed invisibly behind solid objects, which allows for an unobtrusive application. In questions of functionality and reliability, capacitive sensor-based devices used for hand tracking are comparable to gesture recognition systems that are based on other technologies such as cameras [36] and accelerometers [37]. Many of the corresponding gesture recognition software frameworks are based on algorithms that use learning-by-example [38], for instance used in conjunction with pointing devices [39], accelerometer-based input devices [37] and camera-based [36] gesture recognition systems.

Depending on the form factor of the appliance, required input dimensions and precision, size and weight, learnability and acceptable cost, the use of capacitive sensing may be an interesting alternative to conventional techniques.

As it has already been said, there are various technologies available to recognize gestures in open space. Common methods include cameras [40], depth sensors

[41] or capacitive systems [32]. This work is focusing on the latter. Compared to the other systems capacitive sensors can be employed unobtrusively, work through various materials and do not have a high computational cost [33]. However, there are also various drawbacks, including a lower resolution, limited detection distance, sensitivity towards dynamic electric fields in the immediate, environment and shielding issues in various materials [42].

In this dissertation the objective is fixed then to have a simple and intuitive classification system, based on a SVM theory, for recognizing gestures, ensuring reliability of the classification, reasonable time for calibrating the algorithm and a fast response of the system to a gesture performing. The specific goals of the dissertation are outlined in the following section.

1.2 Objectives

In order to correctly design the system able to recognize gestures, the following objectives are fixed:

- Analyze the concepts and the already developed techniques for the classification methods and how they are most appropriate to perform gesture recognition in different situations.
- Define static and dynamic gestures to be able to know which threshold is the most appropriate to acquire a gesture.
- Decide a classification method able to recognize both dynamic and static gestures.
- Decide the features to extract from the signals and the classifier to separate the 5 classes (gestures), given the requirements of the ability to recognize static gestures, dynamic gestures and complexity. More in details, these constraints are traduced in the following goals:
  1. The training set dimension acquired for each gesture is reduced to the minimum.
  2. The classifier must achieve good gestures recognition percentage.
  3. The processing time for calibration should not exceed 1 min; moreover, the algorithm must be trainable with a small training set, asking the user to perform a minimum number of gestures for the calibration.
  4. The system response to a gesture recognition (acquisition threshold crossing) must not exceed 1 s.

5. The classifier should allow use without retraining it for every session; the classification system hence must reach good performances even when the algorithm is not recalibrated, that is, session independence.

- Test the realized classification method in different contexts, to assess the predefined requirements for the design and give conclusions and suggestions for future developments.

1.3 Thesis Organisation

The thesis is divided in 5 chapters.

In chapter 1 an introduction is given to the dissertation, together with the objectives and the reasons that have led to this project.

In chapter 2 an overview is provided about the state of the art capacitive sensor and about systems already on the market able to recognize gestures.

Chapter 3 illustrates the design of the sensor,on the board used, the software, the classification methods, explaining the reasons of the SVM algorithm.

Chapter 4 reports the tests carried out to assess the validity of the design, adding also some new solutions. For every fixed objective, a test is performed and results are obtained.

In chapter 5 the conclusions about the design and the tests are explained, evidencing which parts of the project should be modified to achieve better results and which give instead satisfying performances. Moreover, suggestions for future developments and future works are given.

Chapter 2

State of the art

2.1 Hand Gesture Recognition

The basic goal of Human Computer Interaction is to improve the interaction between users and computers by making the computer more receptive to user needs. Human Computer Interaction with a personal computer today is not just limited to keyboard and mouse interaction. Interaction between humans comes from different sensory modes like gesture, speech, facial and body expressions. Being able to interact with the system naturally is becoming ever more important in many fields of Human Computer Interaction.

Both non-vision and vision based approaches have been used to achieve hand gesture recognition. An example of a non-vision based approach is the detection of the hand with a capacitive sensor. Theoretically the literature classifies hand gestures into two types static and dynamic gestures. Static hand gestures can be defined as the gestures where the position and orientation of hand in space does not change for an amount of time. If there are any changes within the given time, the gestures are called dynamic gestures. Dynamic hand gestures include gestures like waving of hand while static hand gestures include joining the thumb and the forefinger to form the "Ok" symbol.

The literature survey conducted provides an insight into the different methods that can be adopted and implemented to achieve hand gesture recognition. It also helps in understanding the advantages and disadvantages associated with the various techniques. The literature survey is divided into two main phases i.e. the camera module and the detection module. The camera module identifies the different cameras and markers that can be used. The detection module deals with the pre-processing of image and feature extraction. The commonly used methods of capturing input from the user that has been observed are data gloves, hand belts and cameras. The approach of gesture recognition [72] and [73] uses input extraction through data gloves. A hand belt with gyroscope, accelerometer and a Bluetooth was deployed to read hand movements are used [74] [75]. The authors [76] used a creative Senz3D Camera to capture both colour and depth information and [77] used a Bumblebee2 stereo camera. A monocular camera was used by [78]. Cost efficient models like [79], [80] and [81] have implemented their systems using simple web cameras. The methods [82] [83] make use of a kinect depth RGB camera which was used to capture colour stream. As depth cameras provide additional depth information for each pixel (depth images) at frame rate along with the traditional images [84] [85]. Most technologies allow a hand region to be extracted robustly by utilizing the colour space. These do not fully solve the background problem. This background problem was resolved in [86] by using a black and white pattern of augmented reality markers (monochrome glove). While inbuilt webcams do not give depth information, they require less computing costs. Hence in our model, we used a webcam available in the laptop without the use of any additional cameras or hand markers such as gloves. A large number of methods have been utilized for pre-processing the image which includes algorithms and techniques for noise removal, edge detection, smoothening followed by different segmentation techniques for boundary extraction i.e. separating the foreground from the background. The authors [80] [87] used a morphology algorithm that performs image erosion and image dilation to eliminate noise. Gaussian filter was used to smoothen the contours after binarization [81] [88]. To perform segmentation, in [77] a depth map was calculated by matching the left and right images with the SAD (Sum of Absolute Differences) algorithm. In [77], the Theo Pavildis Algorithm which visits only the boundary pixels was used to find the contours. This method brings down the computational costs. In [80] [84] [87] the biggest contour was chosen as the contour of the hand palm after which the contour was simplified using polygonal approximation. Classification is a process in which individual items are grouped based on the similarity between the items. The approach [89] uses Euclidean distance based classifier to recognise 25 hand postures. Support Vector Machine (SVM) classifier was used in [90] and [82].

2.1.1 Leap Motion

The Leap Motion controller is a small USB peripheral device which is designed to be placed on a physical desktop, facing upward. It can also be mounted onto a virtual reality headset.

Figure 2.1: Leap Motion device [91]

Using two monochromatic IR cameras and three infrared LEDs, the device observes a roughly hemispherical area, to a distance of about 1 meter.

Figure 2.2: hemispherical area [91]

The LEDs generate pattern-less IR light and the cameras generate almost 200 frames per second of reflected data. This is then sent through a USB cable to the host computer, where it is analyzed by the Leap Motion software using "complex maths" in a way that has not been disclosed by the company, in some way synthesizing 3D position data by comparing the 2D frames generated by the two cameras. In a 2013 study, the overall average accuracy of the controller was shown to be 0.7 millimeters.

The smaller observation area and higher resolution of the device differentiates the product from the Kinect, which is more suitable for whole-body tracking in a space the size of a living room. In a demonstration to CNET, the controller was shown to perform tasks such as navigating a website, using pinch-to-zoom gestures on maps, high-precision drawing, and manipulating complex 3D data visualizations.

Figure 2.3: Leap Motion System [91]

Leap Motion initially distributed thousands of units to developers who are interested in creating applications for the device. The Leap Motion controller was first shipped in July 2013. In February 2016, Leap Motion released a major beta update to its core software. Dubbed Orion, the software is designed for hand tracking in virtual reality.[91]

2.1.2 Myo Armband

In the last years, the myoelectric control has grown significantly, and not only for prostheses applications, but also for controlling several electronic devices of daily usage, that is, for human-computer interaction. This is done for the user comfort, but mainly to allow access and manipulation of these devices also during hand-busy situations (imagine a driver which could access the navigation system without removing his hands from the steering wheel) [92],[93].

Concentrating on the EMG armbands, we start describing one recent and well known device on the market and that reflects the most common structure and capabilities of an armband: the Myo armband from Thalmic Labs [94] (Figure 2.4).

Figure 2.4: Myo Armband [2]

The Myo armband has the following characteristics:

- Is equipped with 16 electrodes (bipolar, so 8 signals) placed around the forearm of the user, sharing a common ground.
- It embeds an Inertial measurement unit (IMU) to detect arm movements.
- It comprehends a Bluetooth unit to connect with other devices.
- It has elastic material to push the electrodes to the skin and in order to adapt to the user arm size.

Five predefined gestures can be recognized: close, opening, wrist right, wrist left, and fingers-tap (Figure 2.5). However, developers can combine these preset gestures with arm motions (data from the IMU) to create new gestures.

Figure 2.5: 5 gestures recognized by Myo armband [2]

It provides satisfying performances, but also some important drawbacks (given in decreasing order of importance) are present:

1. It has a reasonable cost (around 200 $). However, this can still be expensive for the application of a low-cost myo-armband, in which the total cost of the hand is around 200$.

2. The dimensions of the armband are too big for a good comfort.

Most of the armbands already developed present similar characteristics:

- The eight bipolar electrodes are used by other devices [95],[92],[96]. This is because by placing eight bipolar electrodes around the forearm, a good displacement between them and a satisfying coverage of the underlying muscles are reached; hence it is mainly a dimensional cause.

- The inertial measurement unit is often integrated to the armband [97],[95],[96],[98], since it gives informations on arm movements and multiplies the possible definable gestures.

- The elastic material, or a Velcro, is common in all of the armbands analyzed, since a certain adaptability to the user's arm is always required.

2.2 Capacitive Sensing

In this section we outline the basic concept of capacitive sensing and review relevant research.

Although capacitive sensing has evolved since the first Theremin, we show the potential of this interaction technique which could be utilized in HCI.

The simplest capacitor consists of two metal plates put close together without touching each other. When current is placed on those plates they can store energy.

When the current is removed and the plates are connected through a circuit, the stored energy initiates a current. Thus, a capacitor works like a small accumulator. The capacity (capacitance) depends on the size of the plates and their distance. Using the effect mentioned above, one can measure and track the distance between a sensor and an object. For this, one of the two plates of a capacitor is replaced by the object to be tracked. In order to hold enough free electrons or charged molecules, the object has to have a relatively high dielectric constant. Most of these materials are electrically conductive like metal, water or the human body. When the object gets closer to the plate, the capacitance of this *capacitor* increases. One can measure the capacitance of this capacitor and from this estimate the distance between sensor plate and object. Connecting the object or person to ground can increase the availability of free electrons in it - and thus the sensitivity of the device.

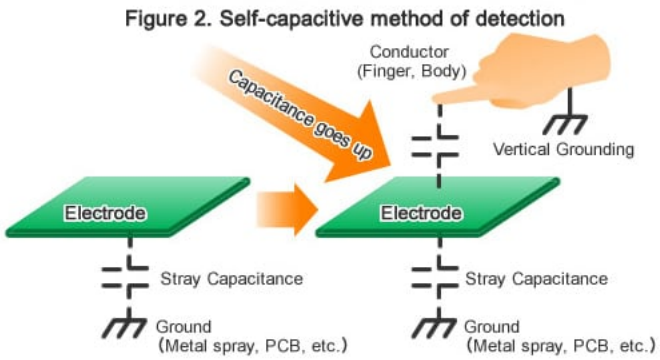


Figure 2.6: Capacitive sensing principle [1]

The most common way to measure the capacitance of a capacitor is to use a resonant circuit. Depending on the capacitor's capacitance, the resonant circuit resonates faster or slower. This technique of measuring distances between a sensor and an object is called capacitive sensing. Such sensors allow measurement of microscopic displacements in the range of micrometers. They are the industry standard for ultra-high precision measurements in many application areas.[43]

However, capacitive sensing can also be used to track objects, e.g. the human hand as electrically conductive object, in larger ranges. The feasibility of using capacitive sensing for position and gesture input to enable intuitive HCI is the main contribution of this work.

Capacitive sensing for gesture interfaces is not a totally new idea. Artists have used Theremins as input devices for video installations or light shows: One of the first capacitive sensing interfaces was a musical instrument called Theremin, which was invented in 1919. The music instrument player can adjust volume and pitch by changing the distances between his hands and two antennas. In such an arrangement, the sensing is relative and due to the feedback (the music created) the absolute position is of minor importance. Changes in the environment and in the system are automatically compensated by the artist as he or she considers the tone rather than the absolute position of the hand as the relevant parameter. Making music in the air using novel technologies, away from traditional instruments, is quite common for performance artists.

An extensive discussion of physical interfaces in arts in general is given by Bongers in [44]. This work also discusses capacitive sensing as input modality. Here, too, using capacitive sensing as relative input with a direct feedback removes many problems faced when creating more generic human computer interfaces.

The technology of capacitive sensing itself is already part of today's computers, e.g. in the touchpads of current laptops. There, input is limited to a very small range of sensor to hand. We extend the sensing range to explore the impacts on the way input to a system can be generated.

Smith et Al. [45] and Zimmerman et Al.[46] explored the potentials of electric field sensing as input modality. They e.g. developed contactless hand tracking devices using electric field sensing. This technique uses sensor plates which produce an electric field and measure its disturbance by the human hand. Electric field sensing yields higher resolution than capacitive sensing but requires significantly more hardware and processing.

Jacky Lee et al. [47] developed a 3D interface device for CAD workstations which uses capacitive sensing. This device (iSphere) only measures three different states (distant, close, pressure). The user needs to touch the iSphere for interaction. Interaction at a distance is not supported.

Different commercial integrated circuits (IC) are available that are based on capacitive sensing. These ICs are targeted at touch control applications. Ethertouch [48] is a recently developed capacitive sensing IC which provides 12 channels for sensors and was engineered for higher precision than Thracker.

Overall in human computer interacting there are very few results discussed that make use of capacitive sensing on a larger scale.The support for gesture input using several sensing plates, as introduced in this paper, is to our knowledge new.

2.2.1 Advantages of Capacitive Sensing

- Low cost sensors:

  Sensors can be built at very low cost. As only some standard ICs and a commonly available USB interface chip are used mainly for the components realizing the USB-connection. When integrating the concept of capacitive sensing into the device (e.g. appliances or a tablet PC) the cost for the hardware can in many cases be neglected.

- Small size, robustness and invisibility:

  Capacitive sensors can be built in a very small form factor. When larger distances (dozens of centimeters) have to be measured, bigger sensor plates may be necessary. However, those can be very flat. The sensor form can be adjusted to fit certain requirements. Capacitive sensing devices do not need moving parts, and can be embedded into solid cases without openings for sensors. This makes them ideally suited for areas with a high threat of vandalism and heavy duty environments. Certain devices already have metal parts that could be reused as sensors. Capacitive sensing can be embedded into devices without showing any signs of it on the outside. This eases design of visually attractive interface devices.

- Scalability, high precision and speed:

  When a greater area has to be covered, additional sensors can easily be installed. Administrative overhead is relatively small. Capacitive sensing allows for a precision in micrometer ranges. This is of course only possible in close proximity. But even in the range of up to 20 centimeters it can sense small movements of about a centimeter. As very little processing is needed on the acquired data, capacitive sensing is very fast. Unlike optical tracking the sensors do not suffer from occluded markers or changing light conditions in the environment. Changes in humidity or temperature do not influence the measured values significantly.

- Ease of use: Once calibrated, capacitive sensing devices do not need additional care. Users interacting with them do not have to carry a transponder or optical marker. Using such a device is intuitive and in most cases no explicit training is needed.

2.2.2 Limitations of Capacitive Sensing

- Sensitivity quickly decreases:

  The major problem with capacitive sensing is that its resolution is highly dependent on the distance. To measure distances of over 30 cm larger sensor plates and highly sensitive circuits have to be built. Active sensing methods can provide greater ranges. Passive capacitive sensing seems to be unfit for tracking objects in larger areas (e.g. 30 centimeters or more).

- Objects interferance:

Capacitive sensors only detect the presence of electrically conductive objects. This need not be the object to be tracked but can be another object which rather should not be tracked. Even if the interfering object does not move it dampens the signals and reduces the tracking resolution. This can only partly be compensated by shielding.

- Limited and ambiguous data:

  The only information a capacitive sensor returns is its capacitance. A certain capacitance can result from one person standing in front of the sensor or from two persons standing a little further away. Disambiguation can sometimes be achieved by using additional sensors and filtering.

2.2.3 Theremin

The theremin is an electronic musical instrument controlled without physical contact by the thereminist (performer). The name come after the Westernized name of its Soviet inventor, Léon Theremin, who patented the device in 1928.

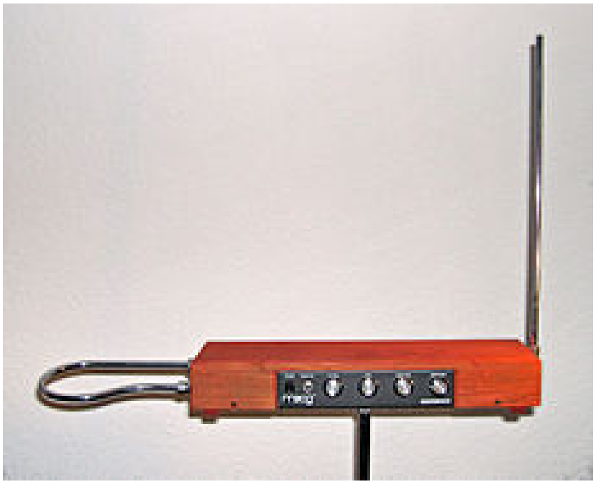


Figure 2.7: Theremin [49]

The instrument's controlling section usually consists of two metal antennas that sense the relative position of the thereminist's hands and control oscillators for frequency with one hand, and amplitude (volume) with the other. The electric signals from the theremin are amplified and sent to a loudspeaker.

The theremin was used in movie soundtracks such as Miklós Rózsa's Spellbound, The Lost Weekend, and Bernard Herrmann's The Day the Earth Stood Still. It has also been used in theme songs for television shows such as the ITV drama Midsomer Murders. This has led to its association with eerie situations.

Theremins are also used in concert music and in popular music genres such as rock.

The theremin is distinguished among musical instruments in that it is played without physical contact. The thereminist stands in front of the instrument and moves his or her hands in the proximity of two metal antennas. The distance from one antenna determines frequency (pitch), and the distance from the other controls amplitude (volume). Higher notes are played by moving the hand closer to the pitch antenna. Louder notes are played by moving the hand away from the volume antenna.

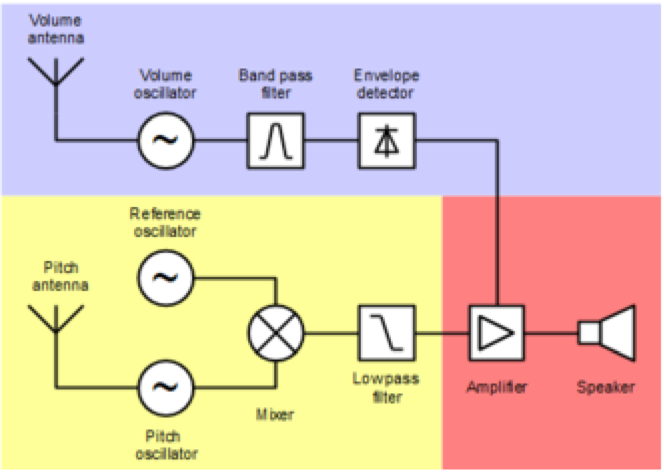


Figure 2.8: Block diagram of a theremin. Volume control in blue-grey, pitch control in yellow and audio output in red. [49]

Most frequently, the right hand controls the pitch and the left controls the volume, although some performers reverse this arrangement. Some low-cost theremins use a conventional, knob operated volume control and have only the pitch antenna. While commonly called antennas, they are not used for receiving or broadcasting radio waves, but act as plates of capacitors.

The theremin uses the heterodyne principle to generate an audio signal. The instrument's pitch circuitry includes two radio frequency oscillators set below 500 kHz to minimize radio interference. One oscillator operates at a fixed frequency. The frequency of the other oscillator is almost identical, and is controlled by the performer's distance from the pitch control antenna.

The performer's hand acts as the grounded plate (the performer's body being the connection to ground) of a variable capacitor in an L-C (inductance-capacitance) circuit, which is part of the oscillator and determines its frequency. In the simplest designs, the antenna is directly coupled to the tuned circuit of the oscillator and the 'pitch field' that is the change of note with distance, is highly nonlinear, as the capacitance change with distance is far greater near the antenna. In such systems, when the antenna is removed, the oscillator moves up in frequency.

To partly linearise the pitch field, the antenna may be wired in series with an inductor to form a series tuned circuit, resonating with the parallel combination of the antenna's intrinsic capacitance and the capacitance of the player's hand in proximity to the antenna. This series tuned circuit is then connected in parallel with the parallel tuned circuit of the variable pitch oscillator. With the antenna circuit disconnected, the oscillator is tuned to a frequency slightly higher than the stand alone resonant frequency of the antenna circuit. At that frequency, the antenna and its linearisation coil present an inductive impedance; and when connected, behaves as an inductor in parallel with the oscillator. Thus, connecting the antenna and linearising coil raises the oscillation frequency. Close to the resonant frequency of the antenna circuit, the effective inductance is small, and the effect on the oscillator is greatest; farther from it, the effective inductance is larger, and fractional change on the oscillator is reduced.

2.2.4 Thracker

The Thracker device is a circuit board containing four separate equal sensing modules (Figure 2.9) – one for each sensor plate.They share a common 5V power supply and ground from the USB port and a clock signal generated by a NE555 timer IC.

Figure 2.9: Thracker [43]

The Thracker device is about the size of a cigarette box. The sensor plates are

4 cm wide.

A NAND gate, a 300 kOhm resistor and a sensor plate in each module provide a rough resonant circuit. When a hand approaches the sensor plate, the capacitance of the sensor plate increases resulting in a lower frequency of the resonant circuit. It usually resonates at 60kHz - 120kHz depending on the distance of the hand.

This signal is fed into a 14-bit binary ripple counter. A clock signal controls whether the signal from the resonant circuit reaches the counter. This is needed to assure that the value of the counter does not change while copying it into the latch.

The raw values are filtered for obviously invalid values and averaged to remove jitter. From this data the software calculates the distance of the hand with regard to each of the sensing plates using hard-coded reference values.

An obvious usage area for Thracker-equipped screens are interactive displays in museums, exhibitions or public places. Users can interact with art, underground maps or timetables in an simple and intuitive way. Thracker is resistant to vandalism, does not require people to touch the input device and is quite cheap. Existing displays can be easily equipped with a Thracker device. While large TFT displays may dampen the sensitivity of the Thracker device, rear- or front-projection displays are ideally suited.

As Thracker is low-cost, even static paper posters could be made interactive. An interactive poster could sense if someone is standing in front of it. The user could tap special areas on the poster to hear additional voice information or a music sample.

Workers who have to wear protective gloves have difficulty interacting with touchscreens or mice. Thracker enables them to interact with a computer without exposing its input devices to hazardous environments. Similarly in a sterile operating room surgeons may not touch unsterile input devices like mice. Thracker could enable them to easily pan and zoom in x-ray images by simple hand movements. Thracker could also be integrated in vandalism-proof ticket vending machines.

On a larger scale Thracker could be used to track persons in a room. On a smaller scale Thracker could be used in interactive toys, e.g. a small robot which always turn in the direction of its owner's finger.[43]

2.2.5 E-Skin

While previous attempts on capacitive sensing of human hand was mostly focused on a single cell measurement, or have used rigid electronics systems, our inspiration in this project is the human skin. Human skin is a the largest sensing organ in our body and is composed of a large number of sensors. In addition is it flexible and stretchable. An electronic skin composed of several sensing elements within a large area, can be used for sensing proximity of the hand at various points. Moreover, if such skin is flexible, it can wrap around any 3D surface for the same purpose, i.e. gesture, or event detection. This can be beneficial for a safe, interactive and rich form of human machine/ human robot interaction, provided that the machine/robot are equipped with such type of sensor over their body. Human skin is highly intuitive, making it easy to neglect the complexity of the largest sensory organ in our bodies. Our skin is the physical barrier through which we interact with our surroundings, allowing us to perceive various shapes and textures, changes in temperature, and varying degrees of contact pressure. To achieve such high sophistication in its sensing capabilities, several different types of highly specialized sense receptors are embedded within our skin. These receptors fi rst transduce information generated by physical contact into electrical signals and subsequently send it to the central nervous systems for more complex processing. The collected signals are eventually interpreted by the somatosensory cortex,[50] permitting us to successfully navigate our physical world with ease. The effort to create an artifi cial skin with human-like sensory capabilities is motivated by the possibility of such large, multi-sensory surfaces being highly applicable for autonomous artifi cial intelligence (e.g., robots), medical diagnostics, and replacement prosthetic devices capable of providing the same, if not better, level of sensory perception than the organic equivalent. Endowing robots with sensing capabilities could extend their range of applications to include highly interactive tasks, such as caring for the elderly, [51] and sensor skins applied on or in the body could provide an unprecedented level of diagnostic and monitoring capabilities. [52] An artifi cial skin with such sensory capabilities is often referred to in the literature as sensitive skin, smart skin, or electronic skin (e-skin). Although the primary function of human skin is mechanical force sensing, electronic versions can be augmented with additional capabilities. In artifi cial platforms, researchers can incorporate chemical and biological sensors onto fl exible substrates.

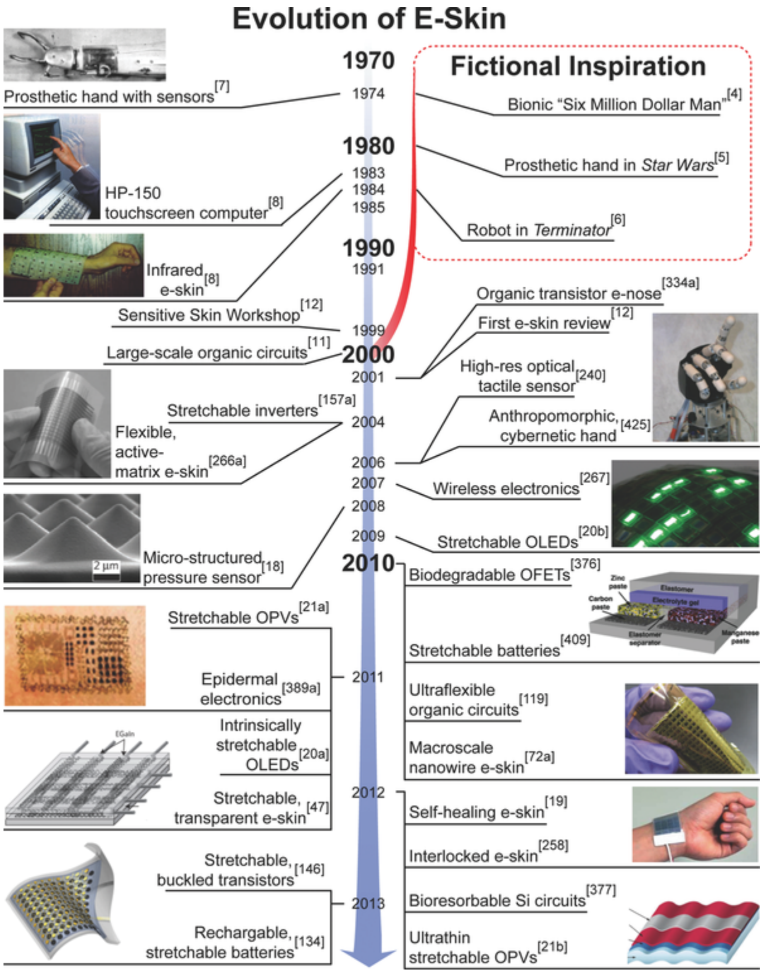


Figure 2.10: A brief chronology of the evolution of e-skin [53]

The prospect of creating artifi cial skin was in many ways inspired by science fi ction, which propelled the possibility of e-skin into the imagination of both the general public as well as the scientifi c community. One of the fi rst science fi ction books to explore the use of mechanical replacement organs was Caidin's Cyborg in 1971, on which the famed Six Million Dollar Man television series about a man with a bionic replacement arm and eye was later based (1974). [54] Shortly after, at the beginning of the 1980s, George Lucas created a vision of a future with e-skin in the famous Star Wars series. In particular, he depicted a scene showing a medical robot installing an electronic hand with full sensory perception on the main character, Luke Skywalker. [55] Shortly after, in 1984, the Terminator movie series depicted humanoid robots and even a self-healing robot. [56] These fictitious renditions of e-skin took place against a real-life backdrop of vibrant microelectronics research that began bridging science fi ction with scientifi c reality. Early technological advancements in the development of e-skin were concomitant with their science fiction inspirations. In 1974, Clippinger et al. demonstrated a prosthetic hand capable of discrete sensor feedback. [57] Nearly a decade later, Hewlett-Packard (HP) marketed a personal computer (HP-150) that was equipped with a touchscreen, allowing users to activate functions by simply touching the display. It was the first mass-marketed electronic device capitalizing on the intuitive nature of human touch. In 1985, General Electric (GE) built the first sensitive skin for a robotic arm using discrete infrared sensors placed on a flexible sheet at a resolution of 5 cm. [58] The fabricated sensitive skin was proximally aware of its surroundings, allowing the robot's arm to avert potential obstacles and effectively maneuver within its physical environment. Despite the robotic arm's lack of fingers and low resolution, it was capable of demonstrating that electronics integrated into a membrane could allow for natural human–machine interaction. For example, the robotic arm was able to 'dance' with a ballerina without any pre-programmed motions. [58] In addition to the ability of an artifificial skin to interact with its surroundings, it is equally critical that the artifificial skin mimics the mechanical properties of human skin to accommodate its various motions. Hence, to build life-like prosthetics or humanoid robots, soft, flexible, and stretchable electronics needed to be developed. In the 1990s, scientists began using flexible electronic materials to create large-area, low-cost and printable sensor sheets. Jiang et al. proposed one of the first flexible sensor sheets for tactile shear force sensing by creating silicon (Si) micro-electromechanical (MEM) islands by etching thin Si wafers and integrating them on flexible polyimide foils. [59] Much work has since been done to enhance the reliability of large sensor sheets to mechanical bending. [60] Around the same time, flexible arrays fabricated from organic semiconductors began to emerge that rivaled the performance of amorphous Si. [61] Just before the turn of the millennium, the first "Sensitive Skin Workshop" was held in Washington DC under the aegis of the National Science Foundation and the Defense Advanced Research Projects Agency, bringing together approximately sixty researchers from different sectors of academia, industry, and government. It was discovered that there was significant industrial interest in e-skins for various applications, ranging from robotics to health care. A summary of concepts outlined in the workshop was compiled by Lumelsky et al. [62] In the early 2000s, the pace of e-skin development significantly increased as a result of this workshop, and researchers began to explore different types of sensors that could be more easily integrated with microprocessors.

Significant progress in the development and advancement of e-skin has been achieved in recent years, in which particular emphasis has been placed on mimicking the mechanically compliant yet highly sensitive properties of human skin. Suo and coworkers have developed stretchable electrodes, [63] and Rogers and coworkers have transformed a typically brittle material, Si, into fl exible, high-performance electronics by using ultrathin (100 nm) films connected by stretchable interconnects. [64] Someya and coworkers have fabricated fl exible pentacene-based organic fi eld-effect transistors (OFETs) for large-area integrated pressure-sensitive sheets with active matrix readout, [65] while Bauer and coworkers have investigated novel pressure sensing methods using foam dielectrics [66] and ferroelectrets [67] integrated with FETs. Our group has investigated the use ofmicrostructured elastomeric dielectrics for highly sensitive capacitive pressure sensors [68] and has developed a composite conductive elastomer exhibiting repeatable self-healing and mechanical force sensing capabilities. [69] Other groups have developed stretchable optoelectronics, including light-emitting diodes (LEDs) [70] and organic photovoltaics (OPVs) [71] for integration with e-skin. A timeline outlining the major milestones towards the development of e-skin is depicted in Figure 2.10 .

2.3 Classification

This section describes the most used classification algorithms. For every reported classifier, advantages and drawbacks will be reported, together with a brief description of the working principles.

The classifier is defined as an algorithm which can detect between a certain number of classes the differences present in the features extracted, and so output the correct performed gesture.

Depending on the feature properties, time constraints and data dimensions present on the overall system, the optimum choice can vary significantly.

2.3.1 K-nearest neighbor

Some binary classification problems (distinction between two classes) can easily be solved sometime (depending on structure) by the usage of a simple and basic algorithm: the k-nearest neighbor [99] (Figure 2.11).

In its simplest form ($K = 1$), the algorithm finds the nearest feature point (belonging to the training set) to the gesture to be classified, and then assigns its label; for $K > 1$, the assignment depends on the majority of labels present in the K nearest neighbors (Figure 2.11). The distance evaluation can be Euclidean or not, and has to be chosen carefully, since the algorithm performance depend almost entirely on this choice.

This algorithm however presents some important limitations:

- Strong noise dependence; that is, if the data is very noisy, the algorithm will surely perform bad.

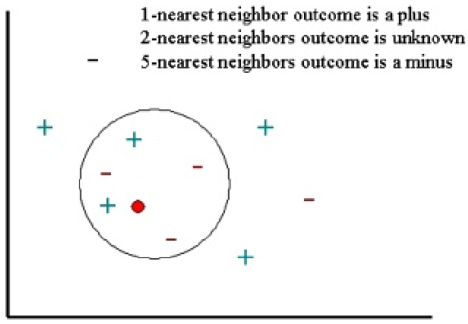


Figure 2.11: K-Nearest neighbor

- It does not perform well in terms of classification time for large training set and $K >> 1$.

2.3.2 Neural networks

The neural networks (Figure 2.12) represent surely the most used classification algorithm ([100],[101] to give some examples). This is because of their several

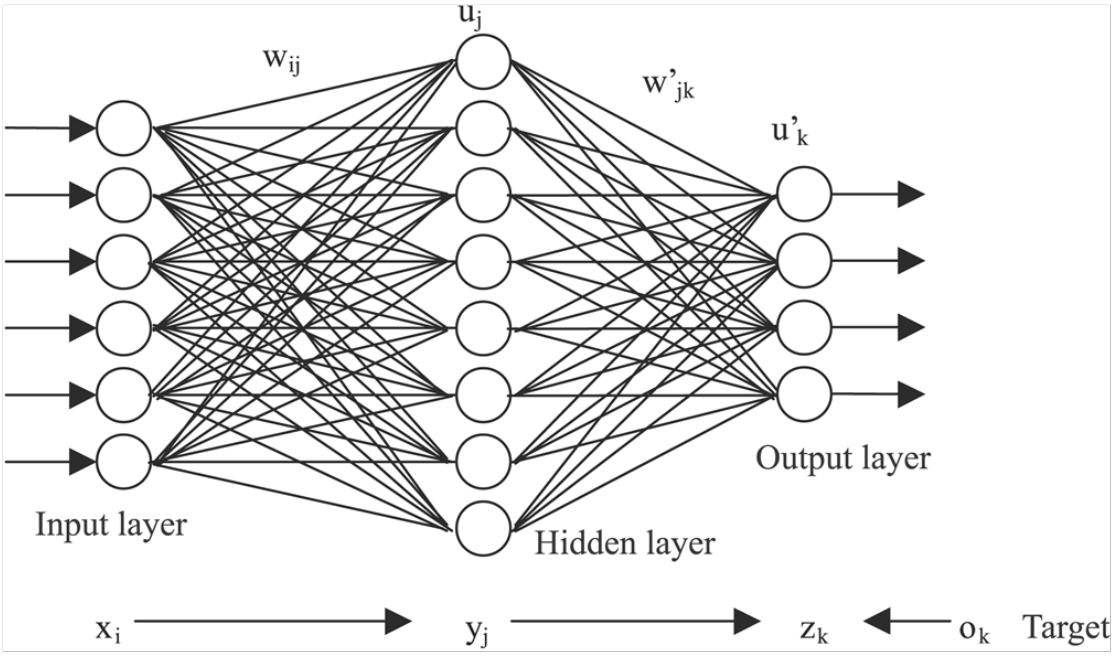


Figure 2.12: General structure of an Artificial Neural Network (ANN)

properties:

- A neural network, depending on the presence of hidden layers, can represent both linear and non-linear systems.
- The learning of the relationships between the variables is self-calibrating, that is, an ANN can create its own representation of the information it receives during learning time, only depending on the data structure. In other words [102], neural networks have the ability to detect implicitly any complex nonlinear relationships between independent and dependent variables, and represent them by automatically adjusting the connection weights in its structure.
- Neural networks can be developed using multiple different training algorithms, so the best performing (fastest) one can be selected among many.
- The neural networks meet real-time constraints, since computations may be carried in parallel.

However, due to their implicit structure, they present some drawbacks [102], which makes them not the best choice for some projects:

- Identify the modeled (by the network) relationships between the variables is hard; also, for causal inference ANN are not preferable.
- Hidden layers and nodes have to be selected accurately, otherwise it is probable to encounter the problem of over-fitting.
- They often require a big computational resource, especially when they include hidden layers.

The input nodes of an ANN are simply the features extracted, so their size is fixed; then, depending on the complexity and linearity of the modeled system, the number of hidden layers and their nodes is decided. There is no clear algorithm to define the number of neurons (nodes) of the hidden layers [100]. Only general rules can be followed,as for instance:

- As the complexity in the relationship between the input data and the desired output increases, the number of the processing elements in the hidden layer should also increase.
- The amount of training data available sets an upper bound for the number of processing elements in the hidden layer.

Once the structure has been determined, the network needs to be trained, and this is done by adjusting the weights (w_{ij}, w_{jk} in Figure 2.12) that connect the variables between the layers.

The different possible solutions are compared through a proper definition of a cost function C, for which obviously the optimal solution represents a minimum. As already said, there are many different algorithms to train an ANN, but they can be grouped, as can be done for every learning algorithm, in the following categories:

1. **Supervised learning**: N labeled training examples $\{(\mathbf{x_1}, y_1); (\mathbf{x_2}, y_2); ...; (\mathbf{x_N}, y_N)\}$ are given, where $\mathbf{X_i}$ indicates the feature vector and y_i the class. In this case, a common technique is to define as cost function the mean-square error (an error is the difference between y_i and $y_i^{\text{predicted}}$), and use the gradient descent to train the network: this technique is indicated as back-propagation algorithm.

2. **Unsupervised learning** N not labeled training examples are given; the cost function depends on the task and a priori assumptions; a commonly used algorithm is the K-means, which divides the data into k clusters such that each point in a cluster is similar to points from its own cluster than with points from some other cluster.

3. **Reinforcement learning** Data are usually unknown, but generated by a software agent. The learning is aimed to automatically determine the ideal behavior within a specific context, in order to maximize its performance.

It is worth citing a specific case of neural network: the single layer perceptron (SLP, Figure 2.13). As said, is a single layer network, with a threshold activation function, capable of separating two classes; it's capabilities are quite limited, since only linear separable data can be classified.

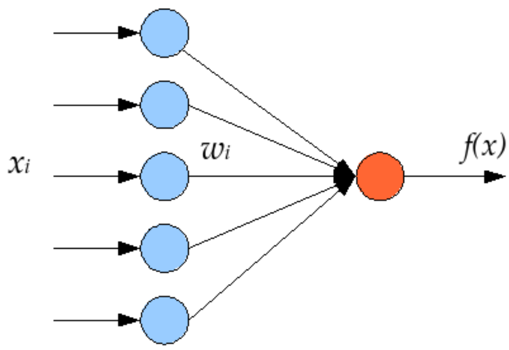


Figure 2.13: Single Layer Perceptron (SLP)

The output of the network is:

$$y = f(\sum_{i=1}^{N} w_i * x_i) \tag{2.1}$$

$$f(a) = \begin{cases} 1 & \text{if } a \geq 0 \\ -1 & \text{if } a < 0 \end{cases} \tag{2.2}$$

Basically, the working principle of the SLP is the evaluation of a hyperplane ($\in \mathbb{R}^{N+1}$) that could separate the two classes.

Equal to the SLP is also another basic form of a classifier: the linear support vector machine. This will be explored in detail further on.

2.3.3 Fuzzy logic

Another approach for classifying is using the fuzzy logic principles. This method presents important advantages [103],[104]:

- Biomedical signals are not always strictly repeatable, and may sometimes even be contradictory. One of the most useful properties of fuzzy logic systems is that contradictions in the data can be tolerated.
- Discover patterns in data which are not easily detected by other methods is possible, as can also be done with neural network.
- The experience of medical experts can be incorporated. Integrate this incomplete but valuable knowledge into the fuzzy logic system due to the system reasoning style is possible, which is similar to that of a human being. This is a significant advantage over the artificial neural network (ANN).
- It has been shown that the fuzzy logic can be a "universal approximator" in a manner similar to the ANN [105].

As described by Chan [103], the step of the classification are the followings:

1. The fuzzy systems at first "fuzzify" inputs into membership degrees of fuzzy sets. This means that a fuzzy clustering is made, where data elements can belong to more than one cluster, and associated with each element is a set of membership levels.
2. Inference by fuzzy logic through rules is made. This is the kernel of a fuzzy system: the knowledge of an expert or well-classified examples are expressed as or transferred to a set of “fuzzy production rules” of the form IF-THEN, leading to algorithms describing what action or selection should be taken based on the currently observed information.
3. The out-coming value is then "defuzzified" to take a decisions. For instance, the simplest but least useful "defuzzification" method is to choose the set with the highest membership, loosing informations. A common and useful

"defuzzification" technique is center of gravity which consists in adding in some way the results of the inference engine.

Fuzzy logic also presents a crucial disadvantage [106]:

- Develop fuzzy rules and membership functions and fuzzy outputs is tedious and can be interpreted in a number of ways making analysis difficult. Moreover, it requires lot of data and expertise to develop a fuzzy system. Also, fuzzy logic performances will be bad for small training set, making this classifier inappropriate for the current project.

2.3.4 Hidden Markov

The Hidden Markov model (Figure 2.14) is part of a bigger group, which will not be explored since it goes beyond the purpose of this Chapter: probabilistic approaches.

The cause of the interest on these classifiers, is that our signal is stochastic, hence, probabilistic approaches that are based on the probability of each class may outperform other classification approaches [107].

Concentrating on Hidden Markov models, their structure is a Markov chain topology consisting of Hidden states (vector $\mathbf{x}$) and state transition probabilities (matrix $\mathbf{A}$). Associated with each state is an observation probability density function, which accounts for the probabilistic nature of the observed data [108].

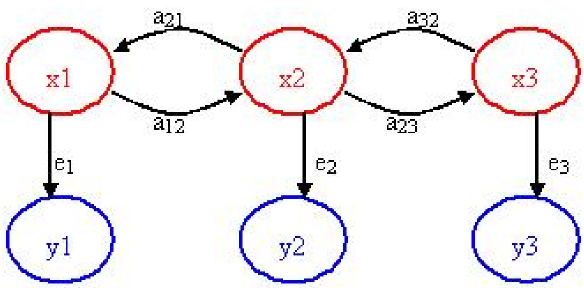


Figure 2.14: Hidden Markov model

Given the observation vector (the features extracted y_1, y_2, y_3, observables) and the previous state probabilities (fixed from experience), the HMM can determine which state has the highest probability of being the current state; the intended limb motion associated with the state with the highest probability is the classification decision for the current time index t [108].

Since initial state probabilities , called π, and the state transition matrix $\mathbf{A}$, are selected preliminarily and fixed, training is only limited to the computation of the mean vector and covariance matrix of the probability functions of state observation (e_1, e_2, e_3) The HMM can reach very high classification accuracies [108],[109], and have relatively low computational complexity, either for calibration and classification. To achieve this, A good tuning of the probabilities vectors and matrix is crucial: for instance, since the application is a prostheses control, the probability of remaining in a particular state should be set relatively high.

This probability tuning anyway allows the HMM to adapt to various systems and applications, and represent a more natural, more effective means of myoelectric control by providing high accuracy, low response time, and an intuitive control interface to the user. Moreover, the low computational overhead associated with training an HMM, also enables the possibility of adaptive classifier training while in use [107].

Chapter 3

Materials and Methods

3.1 System Description

In this section the system which is composed of the capacitive sensor and the board to send the signal is introduced.

3.1.1 Sensor

In order to match the mechanical compliance and deformability of their natural counterparts, electronic skin (E-Skin) sensor arrays and circuits are typically composed of ultrathin metal films, conductive elastomer composites, or liquid metal (LM) microfluidics [110], [111], [112], [113]. LM-based circuits are of particular interest because they can be engineered to exhibit a low elastic modulus ($\sim$ 0.1-1 MPa), high strain limit (100-1000 %), and low electrical resistance ($\sim$ 0.1-1 Ω) [114]. Another reason for their popularity is the ease with which circuits can be fabricated using elastomer casting, microcontact printing, stencil lithography, 3D printing, laser patterning, and a variety of other synthesis techniques [115]. Eutectic gallium-indium (EGaIn) is a popular liquid metal for microfluidic electronics since it has low viscosity, negligible toxicity, and can readily wet to most surfaces [116]. Early efforts with EGaIn electronics focused on low-cost microelectronics prototyping [117], stretchable wiring [118], and strain sensing [119]. Strain sensing with liquid metal was originally shown by RJ Whitney, who created a highly stretchable strain gauge using mercury-filled rubber tubing [120]. Building on this seminal work, Majidi & Wood et al. introduced pressure [121] and bend sensing [122] for applications in contact detection, wearable computing, and joint proprioception. In recent years, EGaIn-based microfluidic sensors have been extended to tactile sensing [123], electrocardiography (ECG) monitoring [124], and variety of medical applications [125], [126], [127].

Proximity detection is important for many E-Skin applications but had previously not been demonstrated with LM microfluidics. Being stretchable is a key factor toward implementation of a bio-inspired skin architecture, which can be wrapped around a robotic arm, a robotic hand or even a mobile robot and provide an excellent distributed sensing information on touch and pressure. Adding proximity sensing to a stretchable e-skin is attractive for robotics applications, with application on prevention of accidents between humans and robots, for a safer human robot interaction. In a more advanced form, such skin might be used for detection of objects shape, size, and geometry in a distributed architecture, to overcome the limitations of computer vision systems. Yao & Zhu presented wearable multifunctonal sensors for pressure and touch sensing using capacitive sensing, introducing also the proximity sensing, using Silver NanoWires[128].

Figure 3.1: Capacitive sensor

Liquid Metal was used as a conductive electrode for several reasons; To allow the sensor to be stretchable; To rapidly fabricate the sensor in an additive manner, and to exploit the deformation of the conductive layer for measuring the changes of the resistance in case of touch.In this chapter, we address this by introducing a multi-layer circuit that is capable of combined proximity and touch. The multi-modal sensor is composed of 24 X 11 capacitive electrodes arranged in two layers (Fig. 3.1). When a finger or other conductive body is above the sensor (but not in contact), the capacitance measurements will change with changes in vertical distance and horizontal motion.

Sensor Design and Fabrication Referring to Fig.3.2, the sensor contains two layer of EGaIn:(i) top capacitor electrodes, (ii) bottom capacitor electrodes.

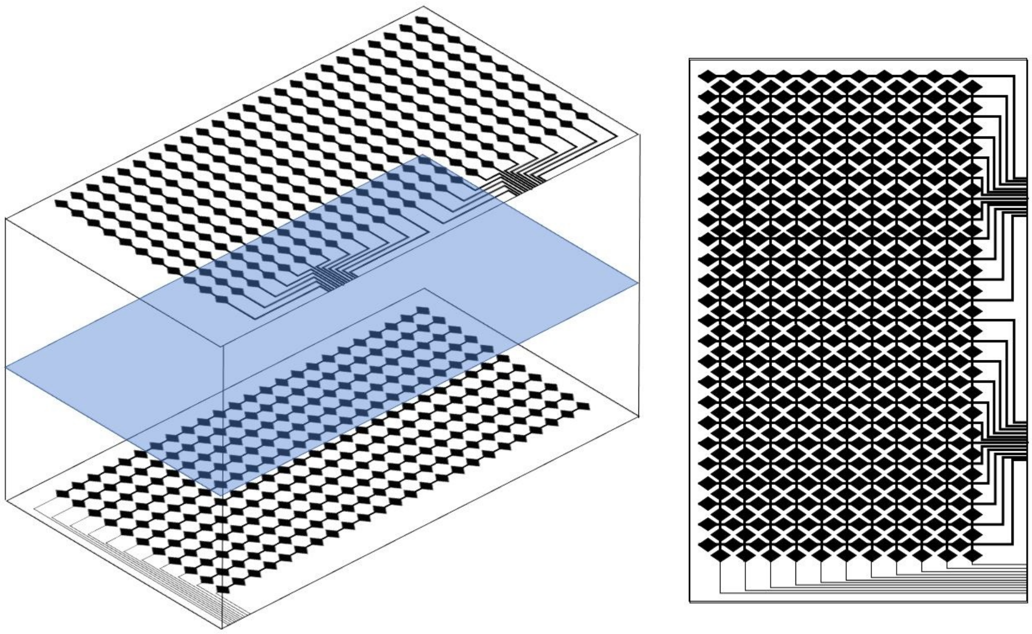


Figure 3.2: Capacitive sensor design

The capacitive sensor array is composed of two electrode layers, each with 10 columns and 23 rows of diamond-shape electrodes, respectively. The layers are aligned so that the electrode in one layer fits perfectly in the space between electrodes of the other capacitive layer.

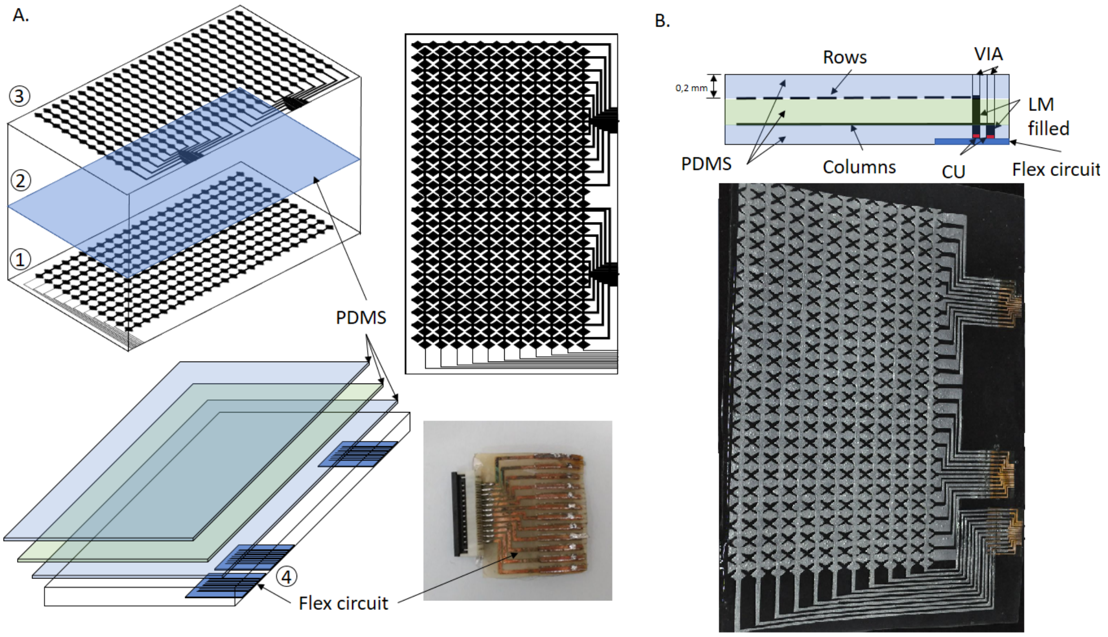


Figure 3.3: Capacitive sensor scheme design

Layers are produced sequentially by first depositing the silicone elastomer with a thin film applicator (ZUA 2000 Universal Applicator; Zehntner). After the polymer is cured (100 °C in an oven for 10 minutes), it is covered with a laser-cut stencil (VLS 3.50: Universal Laser Systems, Inc) and EGaIn is spray deposited [129]. To interface the LM connections to the measurement board, a patterned flexible circuit is used (Fig. 3.3). A flexible copper coated polyimide film was patterned using a UV laser micromachining system (Protolaser U3; LPKF). As shown in Fig. 3.4, the EGaIn layers are deposited in a way that they all interface with the leads of the flexible circuit, which is placed below the bottom-most layer of EGaIn.

Electrical interfacing is accomplished using one of two different techniques. In the first approach, the silicone layers only cover the EGaIn without covering the copper traces. In this way, after spraying EGaIn, the LM interfaces with the copper are unsealed (Fig. 3.4). Using thin layers of EcoFlex (200μ m), the liquid metal of the second layer can be sprayed over the silicone and copper leads without discontinuities, since the step from the copper to the second conductive layer is small. The process is then repeated for the remaining layers, resulting in a layup with the side view shown in Fig. 3.4. In the second approach, the flex circuit leads are coated with an anisotropic conductor that conducts only through its thickness (“z-axis”). This z-axis conductor is prepared based on the method explained in [130]. Ag-coated nickel particles (15 μ m diameter) are mixed with silicone at 30

% wt. After deposition, the sample is placed over a magnet and cured in an oven, during which the ferromagnetic particles self-align along the z-axis. Consequently, subsequently layers of EGaIn that are deposited over the leads are sealed in with the z-axis conductive elastomer. Since the elastomer is only conductive through its thickness, the EGaIn traces and Cu leads will not short within the plane.

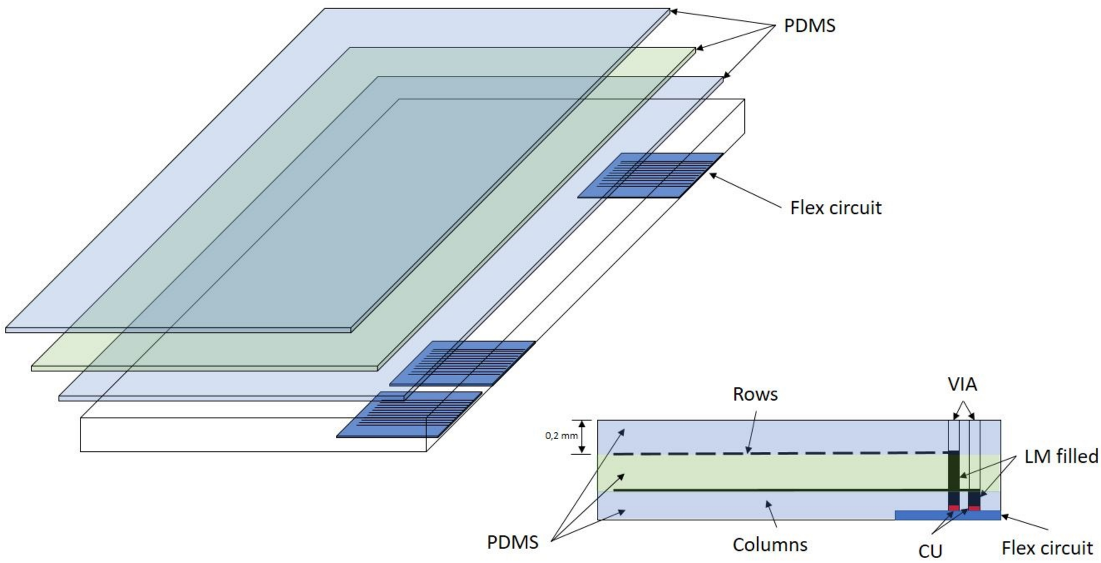


Figure 3.4: Sensor design

3.2 Gestures choice

The first decision it has to be taken is which hand physical movements match with the gestures.Theoretically the literature classifies hand gestures into two types static and dynamic gestures.

Static Gestures Static hand gestures can be defined as the gestures where the position and orientation of hand in space does not change for an amount of time. Some static hand gestures used for our work are shown in the following figures.

Figure 3.5: Static Gestures

Non-Static (Semi-Dynamic) Gestures A dynamic gesture is intended to change over a period of time. The figures below show some dynamic gesture used in this project.

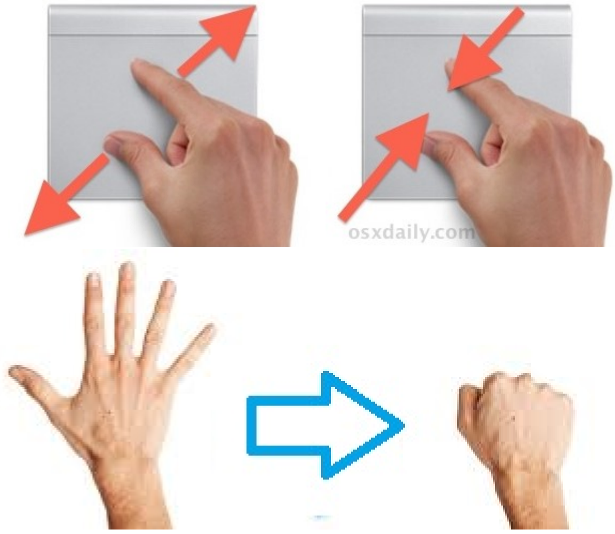


Figure 3.6: Dynamic Gestures

To understand a full message, interpretate all the static and dynamic gestures over a period of time is necessary. This complex process is called gesture recognition.

3.3 Signal Acquisition

In this section the acquisition of the signal coming out from the board will be analyzed.

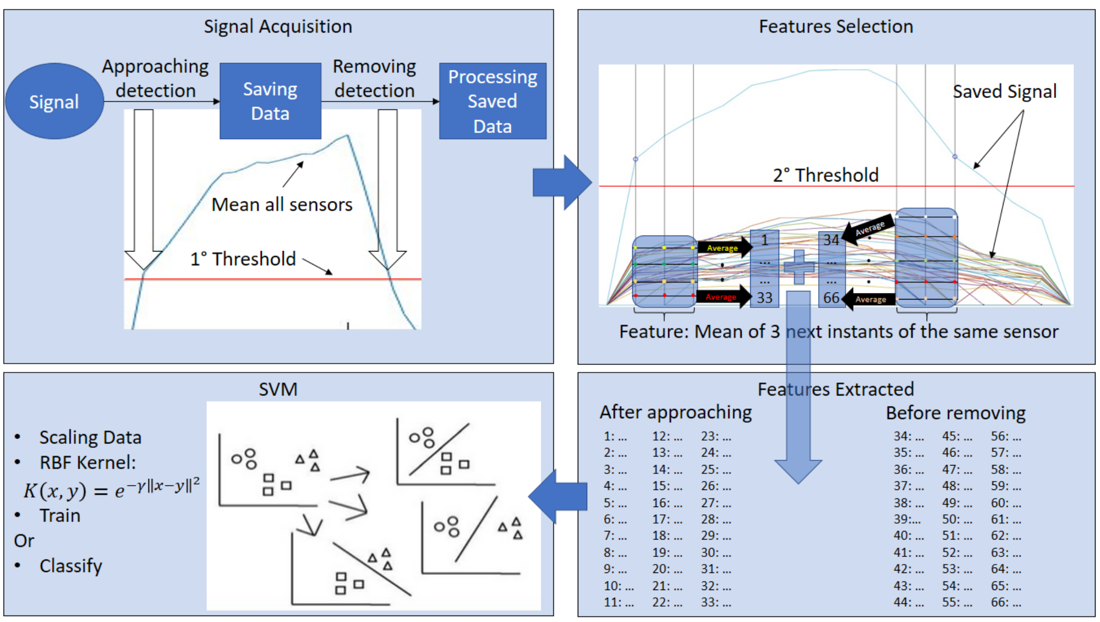


Figure 3.7: Algorithm

The application used for reading the data acquired from the sensor, PsoC Creator (board used CY8C4248AZI-L485) has a new component: the Capacitive Sensing. This component supports various widgets, such as Buttons, Matrix Buttons, Sliders, Touchpads, and Proximity Sensors. The most used for controlling the acquisition in terms of proximity or pressure is the Proximity.

The Proximity widget can detect the proximity of conductive objects. It has two different type of thresholds: one is the proximity threshold for detecting an approaching and one is the touch threshold for detecting the touching on the sensor. The data are affected by noise but the values are analysed and filtered next to the acquisition. Is possible to use different filters but the noise is not to much so we decide to analyse the real signal values.

To avoid the algorithm to classify continuously a predefined windows is needed, of a defined maximum length, where memorize the signal, otherwise the classification error would be quite high; the classification should then only start in response to an event, which represents the starting of a gesture. The technique used in this design is the threshold crossing (Figure 3.8), when the signal (continuously acquired) crosses a certain threshold, the algorithm will start saving his values until:

- The hand is removed. When the signal crosses a predefined lower threshold;

  or

- For a maximum predefined period (window). The maximum length of the window of observation is decided from experience, and is a trade-off between:

  – The average duration of the gestures, which sets a lower bound, since a complete gesture should be recorded in order to have a good classification.

  – The real-time constraints, which set an upper bound; the duration of the window indeed should not exceed a certain value. The duration upper bound is then set from experience to 1 second

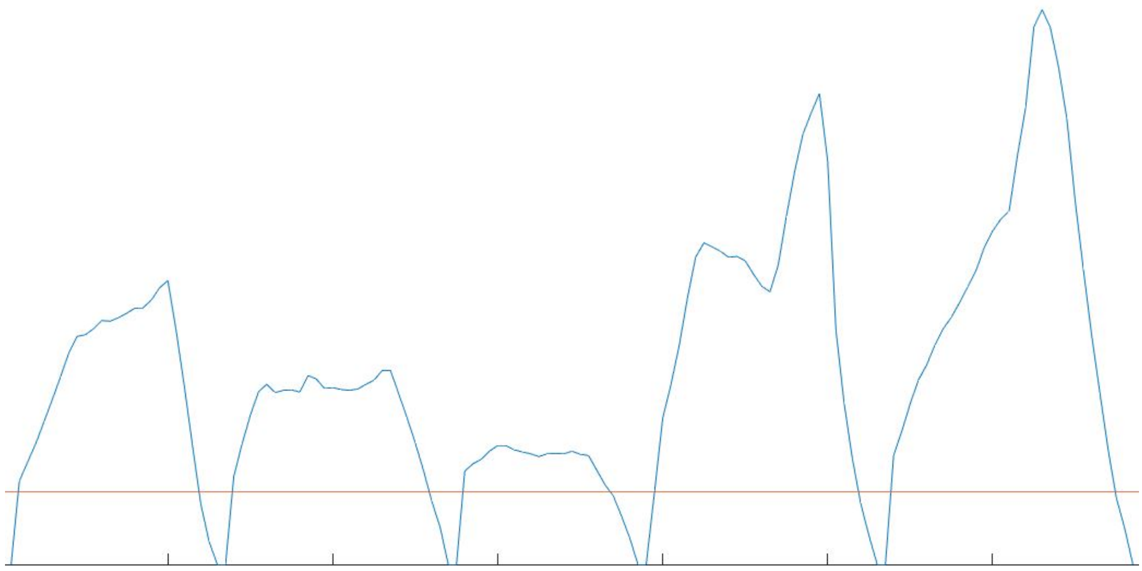


Figure 3.8: Threshold

The threshold selected for this design is the mean of all 33 sensors, this to avoid acquiring when there are peaks in just few sensors due to interferences from the surrounding environment.

The sketch of this operation is represented in Figure 3.9.

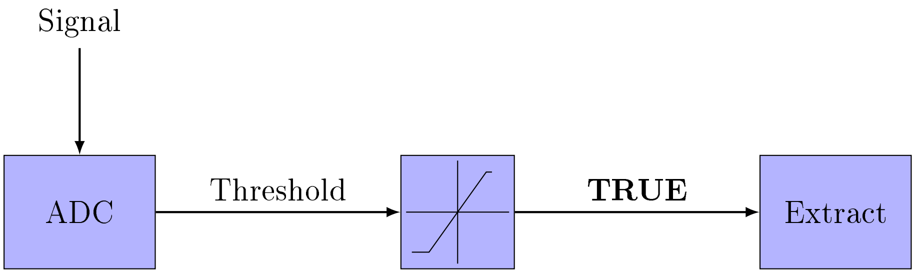


Figure 3.9: Acquisition sketch

Once the threshold has been crossed, the next N ADC output values will be recorded (with maximum N experimentally set to 100, corresponding to an overall window length of $\sim 1s$) and processed to extract proper features.

3.4 Features Selection

Choosing the right features to extract from the signal is an important step in the hand gestures classification. This section will analyze deeply the block "Extract" depicted in Figure 3.9. The block carries out the signal, and then:

- Sends the obtained features vector as an instance to train the classification algorithm, if the user is requiring a Calibration.
- Classifies the obtained features vector, if the user is using the system.

Before analyzing the choice of the features to extract, all the 33 sensors for the 5 gestures are shown in Figure 3.10, to give an idea of which characteristics separates them from each other.

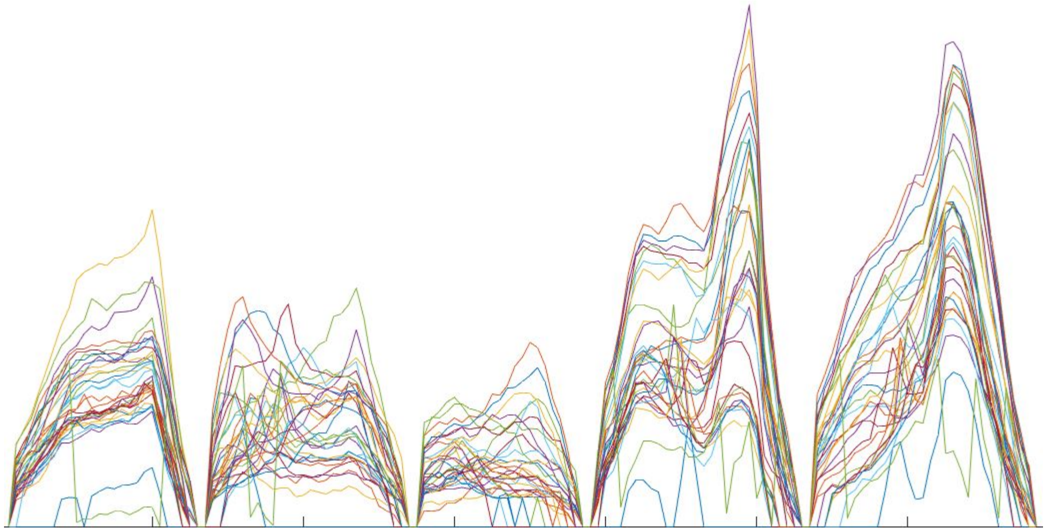


Figure 3.10: 33 sensors 5 gestures

Figure 3.11: Gestures

In order to select the right features to extract, start from the desired characteristics that these features should have is recommended:

1. The processing time to extract the features should be as low as possible, because the user, once it completes the gesture, would obviously desire a quick actuation of his commands.

2. The features serve as an input for the classifier, so should separate the gestures, static and dynamic, from each other as good as possible.

Having a good number of sensors (33) available, to identify a gesture the value of each sensor, at the same time, is sufficient. What we use in this project is to capture a snapshot of all the sensors in the desired instant. The static gestures are not changing during the time so is enough to capture only one snapshot to classify them correctly. To classify a dynamic gesture the system needs two different snapshots, one before the changing and one after. To do this is necessary to find a good way to understand where there is a change.

We can think about a gesture as a composition of 3 parts:

- Approaching to the sensor.
- Changing or standing if is static.
- Removing from the sensor.

Since that small changings can happen also in a static gesture, little movements of the hand cause changing, instead of acquire the snapshots before and after the changing is better to acquire them after the approaching and before the removing. Doing this also with the static gesture, theoretically, two "equal" snapshots are acquired, this allows to set the number of features, 2 X n°of sensors, for both categories and it is possible to classify the gestures with the same algorithm.

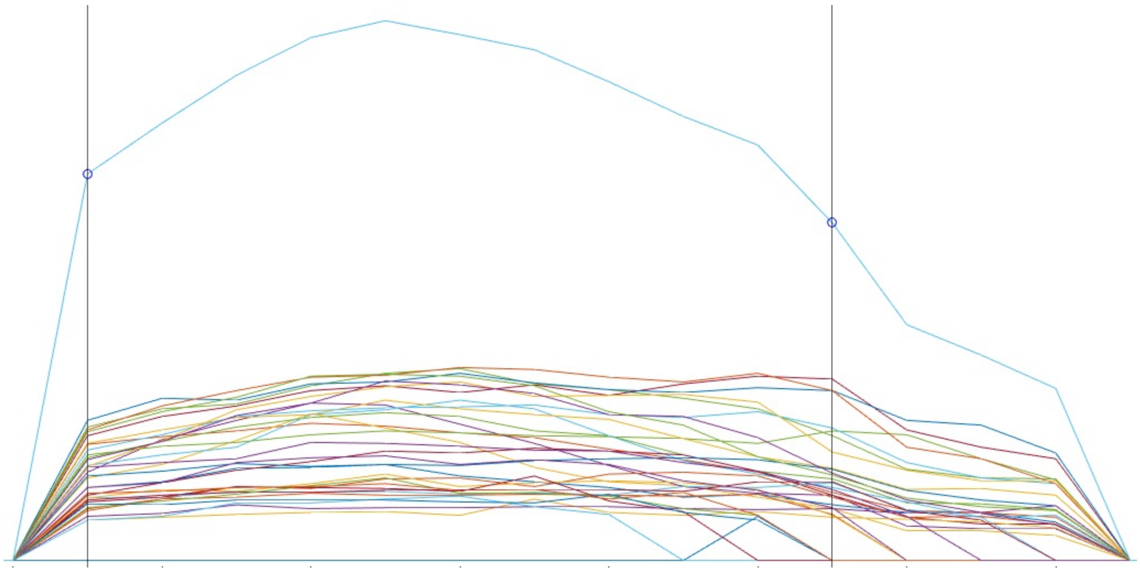


Figure 3.12: 2 Snapshots 1 instant acquiring

To achieve good results it is better to use more than one instant for each snapshot. After the first tests we decided to use the mean of 3 instants after the approaching and the same before the removing. Doing so we exclude the change part and we improve the system because each snapshot contains the informations of 3 sequential moments as shown in Figure 3.13

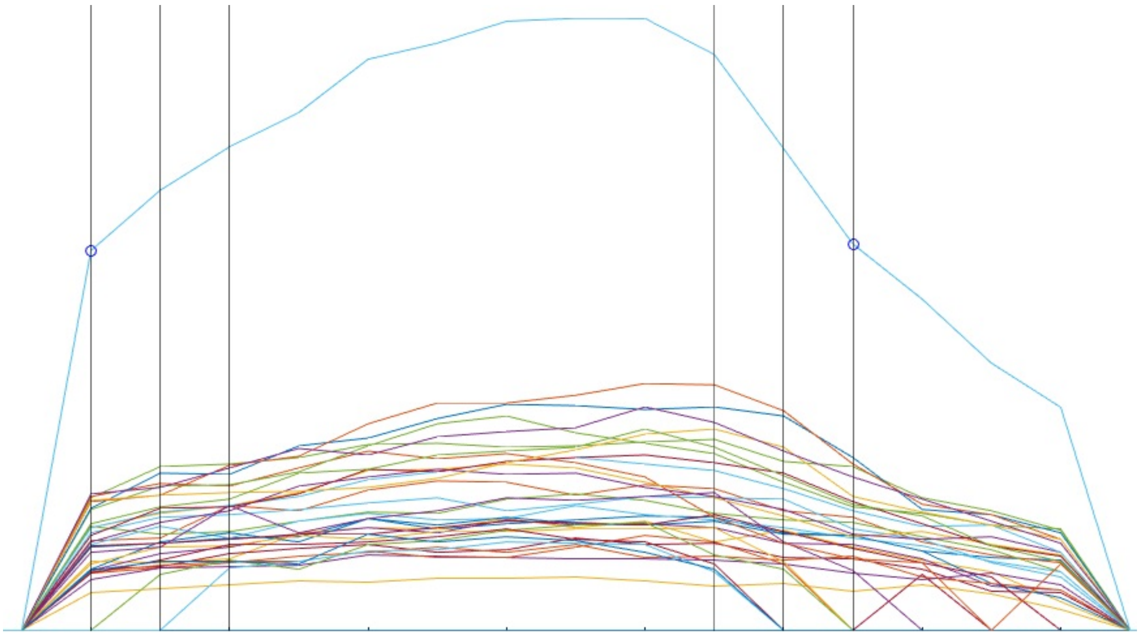


Figure 3.13: 2 Snapshots mean of 3 sequential instants

3.5 Classification Algorithm

This section describes the used classification algorithm: starting from the motivations which brought to this choice, explaining after the principles beyond the operations present inside the calibration part, and finally showing how the pure gestures classification works.

The characteristics and requirements of the system are listed below:

- The feature space is separable, probably linearly; hence, a complex classifier is not necessary.
- Due to real-time application, a very fast classification (feature extraction + classification time $t_c < 1s$) is required, even if the feature space or the training set are quite large.
- Give the possibility to decide some parameters of the classifier is preferable, mainly for future developments. These parameters can be:
  1. Training set dimensions, to feed faster the algorithm (less instances to give).
  2. Rigidity to separability. In other words, with relatively large training sets is possible that there is no reachable l.s., due to bad instances

given to feed the classifier; hence, forcing the algorithm to find a linear separator (hyperplane) could result in long time calibration and misclassification. Give the possibility to set the so-called "Soft margin" is necessary.

3. Error-tolerance and time limit for the calibration, to speed up the pure training time.

For all these requirements and peculiarities, a support vector machine classifier was chosen (Figure 3.14). Such classifier is trained with the sequential minimal optimization algorithm, developed by John Platt [131].

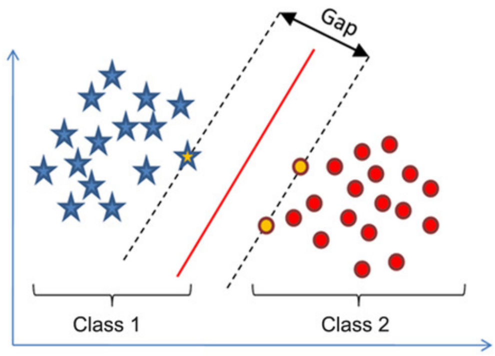


Figure 3.14: Support vector machine

In this design the technique of one vs one is used, In the one vs one reduction, one trains $K * (K-1)/2$ binary classifiers for a K multiclass problem (5 in this case); each receives the samples of a pair of classes from the original training set, and must learn to distinguish these two classes. At prediction time, a voting scheme is applied: all $K * (K-1)/2$ classifiers are applied to an unseen sample and the class that got the highest number of $+1$ predictions gets predicted by the combined classifier.

Support vector machine introduction The binary s.v.m. principle is the construction of an hyperplane which can separate as clear as possible (maximizing a distance) the two classes of instances (Figure 3.15); that is, given N instances $\{(\mathbf{x_1}, y_1); (\mathbf{x_2}, y_2); ...; (\mathbf{x_N}, y_N)\}$, where $\mathbf{x_i} \in R^d$ and $y_i \in \{1, -1\}$, the purpose of the s.v.m. algorithm is to train a separating hyperplane $f(\mathbf{x})$, which in the linear case can be written like:

$$f(\mathbf{x}) = sgn(\mathbf{w} \cdot \mathbf{x} - b) \tag{3.1}$$

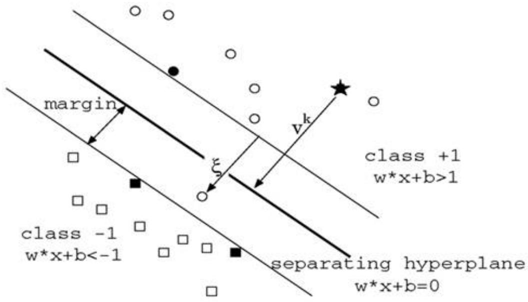


Figure 3.15: Support vector machine principle

The role of this classifier is to separate the two classes with a defined margin. Hence, the calibration of the classifier will be a modification of the weights **w** in order to have two parallel hyperplanes (H1,H2) such that:

$$H1: \ \mathbf{w} \cdot \mathbf{x} - b = \zeta \ for \ class \ 1 \tag{3.2}$$

$$H2: \ \mathbf{w} \cdot \mathbf{x} - b = -\zeta \ for \ class \ 2 \tag{3.3}$$

The two constraints can be unified:

$$y_i(\mathbf{w} \cdot \mathbf{x}^i - b) \geq \zeta \ \forall i \tag{3.4}$$

$$\zeta_i \geq 0 \quad i = 1, ..., m \tag{3.5}$$

The margin that separates the classes is the distance between the two hyperplanes (Figure 3.16), so in the case $\zeta = 1$:

$$|H_2 - H_1| = \frac{|\mathbf{w} \cdot \mathbf{x} - b|}{||\mathbf{w}||} = \frac{2}{||\mathbf{w}||} \tag{3.6}$$

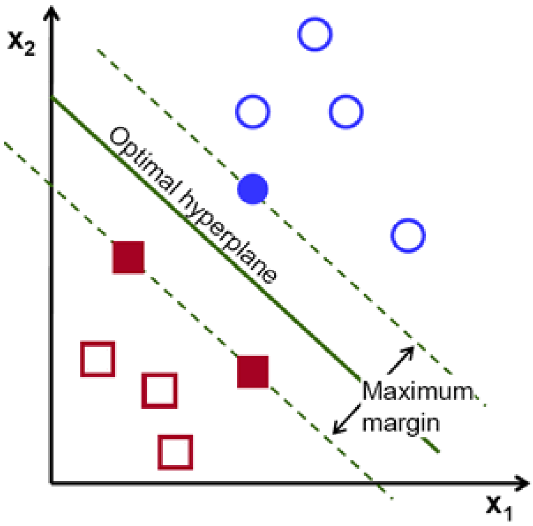


Figure 3.16: Margin of the s.v.m.

To obtain a good separation of the two classes then, $||\mathbf{w}||$ is to minimize, which is equal to minimize $\frac{1}{2}\mathbf{w}^T\mathbf{w}$. The classifier, in the linear case, can then be determined solving this quadratic programming problem:

$$\min_{\mathbf{w},b} \frac{1}{2}\mathbf{w}^T\mathbf{w} \quad with \quad y_i(\mathbf{w}\cdot\mathbf{x}-b) \geq 1 \ \forall i. \tag{3.7}$$

To solve this problem, introduce the so-called Lagrange multipliers is convenient, create the dual problem, and solve it.

Lagrange multipliers and dual problem To understand the theory of the Lagrange multipliers, following [132], start from an original minimization problem of the type is convenient:

$$\min_{\mathbf{w}} f(\mathbf{w}) \tag{3.8}$$

Subject to the primal constraints:

$$g_i(\mathbf{w}) \leq 0 \ i = 1, ..., N \tag{3.9}$$

$$h_i(\mathbf{w}) = 0 \ i = 1, ..., M \tag{3.10}$$

To solve the problem, the best way is to define the so-called generalized Lagrangian:

$$\mathcal{L}(\mathbf{w}, \alpha, \beta) = f(\mathbf{w}) + \sum_{i=1}^{N} \alpha_i g_i(\mathbf{w}) + \sum_{i=1}^{M} \beta_i h_i(\mathbf{w}) \tag{3.11}$$

The Lagrangian multipliers are α_i and β_i. If the following quantity is considered:

$$p = \max_{\alpha,\beta,\alpha_i \geq 0} \mathcal{L}(\mathbf{w}, b, \alpha, \beta) \tag{3.12}$$

it is easy to verify that, given a general $\mathbf{w}$, if it violates any of the primal constraints ($g_i(\mathbf{w}) leq 0$ or $h_i(\mathbf{w}) = 0$) the solution of the previous problem is:

$$p = \max_{\mathbf{alpha},\mathbf{beta},\alpha_i \geq 0} [f(\mathbf{w}) + \sum_{i=1}^{N} \alpha_i g_i(\mathbf{w}) + \sum_{i=1}^{M} \beta_i h_i(\mathbf{w})] = \infty \tag{3.13}$$

If then it is considered:

$$\min_{\mathbf{w},b} p = \min_{\mathbf{w}} \max_{\alpha,\beta,\alpha_i \geq 0} \mathcal{L} \tag{3.14}$$

it can be seen that is exactly the original problem (Equation 3.9) and has then the same solutions.

If then the following quantity is considered:

$$d = \max_{\mathbf{alpha},\mathbf{beta},\alpha_i \geq 0} \min_{\mathbf{w},b} \mathcal{L}(\mathbf{w}, b, \alpha, \beta) \tag{3.15}$$

it can be easily demonstrated that $d \leq p$ and under certain conditions $d = p$; that means that the above problem (Equation 3.16, called dual problem) can be solved to find the solutions of the primal one (the dual problem finds α^*, β^* and the primal finds $\mathbf{w}^*$, but it will be shown that they are dependent from each other) This is useful because often the dual problem is much easier to solve than the primal one. So all is needed is then to satisfy the following conditions, called Karush-Kuhn-Tucker (KKT) conditions, to be able to state that $d = p$:

$$\frac{\partial}{\partial w_i} \mathcal{L}(\mathbf{w}^*, \alpha^*, \beta^*) = 0 \quad i = 1, ..., L \tag{3.16}$$

$$\frac{\partial}{\partial \beta_i} \mathcal{L}(\mathbf{w}^*, \alpha^*, \beta^*) = 0 \quad i = 1, ..., M \tag{3.17}$$

$$\alpha_i^* g_i(\mathbf{w}^*) = 0 \quad i = 1, ..., k \tag{3.18}$$

$$g_i(\mathbf{w}^*) \leq 0 \quad i = 1, ..., k \tag{3.19}$$

$$\alpha_i^* \geq 0 \quad i = 1, ..., k \tag{3.20}$$

It is important to notice that the link between the KKT conditions and the solutions of the primal ($\mathbf{w}^*$) and dual (α^*, β^*) problems is bi-directional. This means that if some $\mathbf{w}^*, \alpha^*, \beta^*$ satisfy the KKT conditions, than they are also solutions of the primal and dual problem p and d.

Going back to the original s.v.m. problem, now is possible rewrite it in the following way:

$$\min_{\psi,\mathbf{w},b} \frac{1}{2}||\mathbf{w}||^2 \tag{3.21}$$

$$g_i(\mathbf{w}) = -y^i(\mathbf{w} \cdot \mathbf{x}^i + b) + 1 \leq 0 \quad i = 1, ..., N \tag{3.22}$$

The last constraint is present for each training example i.

For the third constraint of the KKT conditions (Equation 3.19) the only training examples that will have $\alpha_i > 0$ will be the ones that have functional margin exactly equal to one (i.e., the ones corresponding to constraints that hold with equality $g_i(\mathbf{w}) = 0$). These training examples are the so-called **support vectors**. This fact is useful when working with large training set, because only a few examples will have $\alpha_i > 0$, hence the classification will be anyway fast.

In other words, in the support vector machine algorithm, the value of α_i indicates the weight of the i-th training point in the classification of gestures: this can be seen in the last paragraph. Since the s.v.m. problem now is in the standard form, it can be created the Lagrangian:

$$\mathcal{L}(\mathbf{w}, b, \alpha) = \frac{1}{2}||\mathbf{w}||^2 - \sum_{i=1}^{m} \alpha_i[y^i(\mathbf{w} \cdot \mathbf{x}^i + b) - 1] \tag{3.23}$$

The β_i are not present since only inequality constraints are present.

Find the dual problem of the primal form is now convenient. Firstly then, find the minimum of the Lagrangian $\mathcal{L}(\mathbf{w}, b, \alpha)$ with respect to $\mathbf{w}$ and b is needed. To do so, the derivatives are set to zero, it is then obtained:

$$\nabla_{\mathbf{w}}\mathcal{L}(\mathbf{w}, b, \alpha) = \mathbf{w} - \sum_{i=1}^{m} \alpha_i y^i \mathbf{x}^i = 0 \tag{3.24}$$

The following relation is then obtained:

$$\mathbf{w} = \sum_{i=1}^{m} \alpha_i y^i \mathbf{x}^i \tag{3.25}$$

Setting the derivative of the Lagrangian $\mathcal{L}(\mathbf{w}, b, \alpha)$ with respect to b to zero, the following constraint is instead obtained:

$$\sum_{i=1}^{m} \alpha_i y^i = 0 \tag{3.26}$$

Combining the last two equations with the primal form of the problem, then obtain the final dual form is possible, which is:

$$\max_{\alpha} W(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} y^i y^j \alpha_i \alpha_j (\mathbf{x}^i \cdot \mathbf{x}^j) \tag{3.27}$$

$$\alpha_i \geq 0 \quad i = 1, ..., m \tag{3.28}$$

$$\sum_{i=1}^{m} \alpha_i y^i = 0 \tag{3.29}$$

The KKT conditions for d=p are also satisfied, hence to solve the dual problem and then find the solutions ($\mathbf{w}$) for the primal one using 3.26 is allowed. The algorithm to solve the problem will be discussed in the paragraph "Sequential minimal optimization". Now the non-linear s.v.m. will be discussed, together with the kernel functions.

Kernel functions and non linear s.v.m From the dual problem outlined in equation 3.28, it can be seen that the quantity to be minimized depends only on the inner product of the data: $\mathbf{x}^i \cdot \mathbf{x}^j$. Express the entire algorithm in terms of only inner products between input feature vectors is feasible, and by using this property a non-linear support vector machine algorithm can be reached, which will be very fast even in high-dimensional spaces.

As depicted in Figure 3.17, some classification problems could require a non-linear separation border, which can not be achieved by using only the simple inner product; a transformation $\phi(\mathbf{x})$ of the feature variables to another high dimensional space such that the data points will be linearly separable is needed.

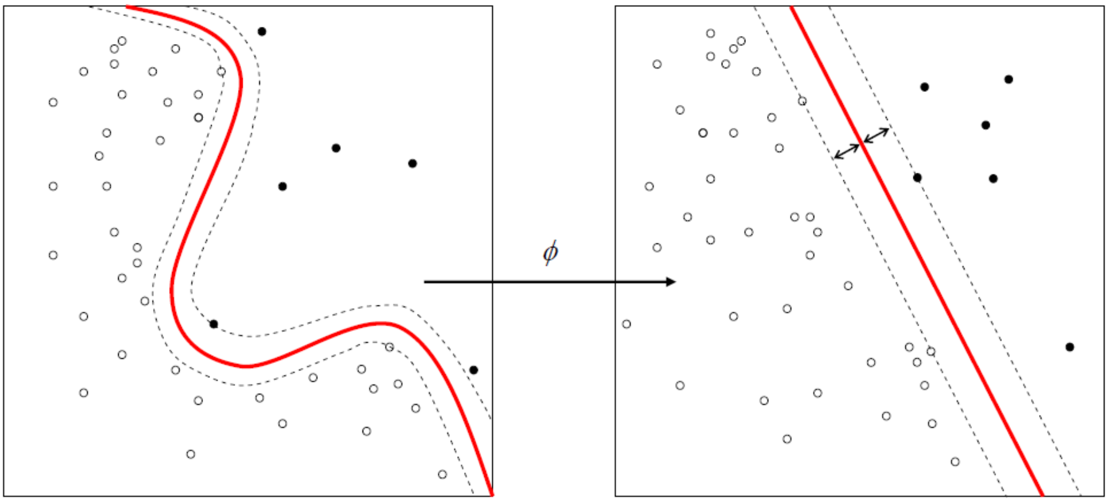


Figure 3.17: Non-linear s.v.m.

The quantity to be maximized will be then:

$$\max_{\alpha} W(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} y^i y^j \alpha_i \alpha_j (\phi(\mathbf{x}^i) \cdot \phi(\mathbf{x}^j)) \tag{3.30}$$

Since only the inner product $\phi(\mathbf{x}^i) \cdot \phi(\mathbf{x}^j)$ is present, everything can be expressed as $\phi(\mathbf{x}^i) \cdot \phi(\mathbf{x}^j) = k(\mathbf{x}_i, \mathbf{x}_j)$, where $k(\mathbf{x}_i, \mathbf{x}_j)$ is a kernel function in the input space, that represents a measure of similarity between feature points. Expand the form $\phi(\mathbf{x})$ is not necessary. In this way it is easy to face also non-linear problems with the usage of the support vector machine algorithm and kernel functions.

The function $k(\mathbf{x}_i, \mathbf{x}_j)$ expresses the closeness of the two feature points $\mathbf{x}_i$ and $\mathbf{x}_j$, in other words how similar they are.

There are many possible choices of the function $k(\mathbf{x}_i, \mathbf{x}_j)$, the only constraint is that it should reach a maximum when $i = j$, since the two points are then as close as possible (they are equal). A very common choice is using the simple inner product $\mathbf{x}^i \cdot \mathbf{x}^j$, but that can bear only linear problems (since no transformation ϕ is carried out). Another famous form, and the one that will be used in this project, is the Gaussian kernel (Figure 3.18):

$$K(\mathbf{x}_i, \mathbf{x}_j) = e^{\frac{||\mathbf{x}_i - \mathbf{x}_j||^2}{2\sigma}} \tag{3.31}$$

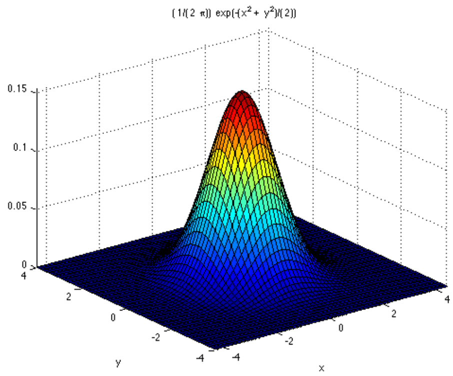


Figure 3.18: Gaussian kernel

To establish if a function can be used as a kernel function, one can use the Mercer conditions, which can be easily found on the web. Obviously the Gaussian kernel satisfies these conditions.

Imperfect separation The above support vector machine algorithm assumes that the data are perfectly separable, that is, no points exist between H_1 and H_2. With large training set this is not always the case, so a separation could not be reached, or even worse, to find an hyperplane which separates the two classes, the margin decays dramatically (Figure 3.19). To avoid that, a finite penalization is inserted for the points which cross the boundaries: C. If $C = \infty$ the original problem is obtained, since no errors are tolerated.

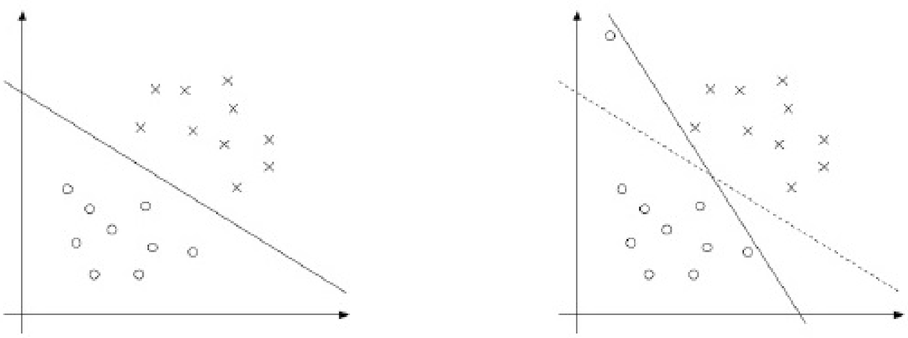


Figure 3.19: Imperfect data separation

The original primal problem becomes then:

$$\min_{\mathbf{w},b} \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{m} \zeta_i \tag{3.32}$$

$$y_i(\mathbf{w} \cdot \mathbf{x}^i - b) \geq \zeta \quad i \tag{3.33}$$

$$\zeta_i \geq 0 \quad i = 1, ..., m \tag{3.34}$$

So now have examples with functional margin less than 1 is permitted, adding because of that a cost to the weighting function equal to $C\zeta_i$. Hence, the parameter C controls how much tolerance it is wanted to the misclassified training samples.

Anyway, since it is way easier to solve the dual problem d, the effect that this new cost function has to it must be investigated. Without going in details, the only difference between the old dual problem and the new is the addition of an upper bound on the α, the new problem becoming:

$$\max_{\alpha} W(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{m} y^i y^j \alpha_i \ alpha_j (\mathbf{x}^i \cdot \mathbf{x}^j) \tag{3.35}$$

$$0 \leq \alpha_i \leq C \quad i = 1, ..., m \tag{3.36}$$

$$\sum_{i=1}^{m} \alpha_i y^i = 0 \tag{3.37}$$

This final form is going then to be used for the classification, since has the following important properties:

- Can deal with large training set and high-dimensional feature space, maintaining always good results in terms of classification speed.

- The algorithm is able to classify also non-linearly separable data sets, thanks to the usage of the Gaussian kernel.

- For large training sets, the algorithm can tolerate non perfectly separable data; and this tolerance can be tuned by changing properly the parameter C.

The next step is then find a fast algorithm to solve the maximization problem of equation 3.37. In 1998 Platt [131] found a satisfying solution to this problem: the **sequential minimal optimization**.

Sequential minimal optimization Considered the following maximization (which derives from the s.v.m.):

$$\max_{\alpha} W(\alpha) \tag{3.38}$$

To find the optimal solution, the technique is to proceed step by step, that is, to fix $m-2$ of the α_i values, and change a pair of them by putting a derivative equal to zero.

Consider the α_i in pairs is necessary because of the constraint $\sum_{i=1}^{m} \alpha_i y^i = 0$ that does not allow to change a single α, since it can be determined by the other $m-1$ values immediately.

The SMO algorithm can be synthesized with the next two steps:

- Loop until convergence or time expired [

  1. Select two multipliers α_i and α_j to update.
  2. Re-optimize $W(\alpha)$ with respect to this two values, maintaining the other multipliers fixed.

  ]

The process principle can be seen in Figure 3.20, where the maximum of a quadratic function (depending on two variables) is found varying only one parameter at a time:

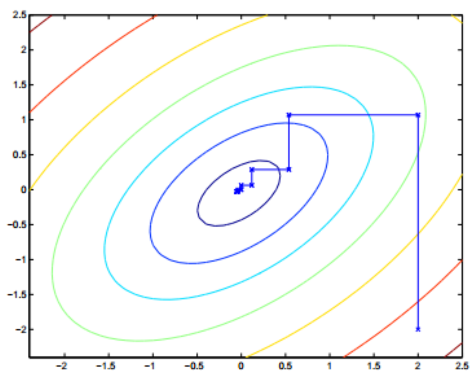


Figure 3.20: Sequential minimal optimization principle

The single iteration of the algorithm works as follows: once two multipliers are selected, called for simplicity α_1 and α_2, the following relation holds from the constraint 3.39:

$$\alpha_1 y^1 + \alpha_2 y^2 = -\sum_{i=3}^{m} \alpha_i y i \tag{3.39}$$

And since the right part of the equation is fixed:

$$\alpha_1 y^1 + \alpha_2 y^2 = \psi \tag{3.40}$$

Then rewrite one multiplier as a function of the other is possible:

$$\alpha_1 = (\psi - \alpha_2 y^2) y^1 \tag{3.41}$$

The function to be maximized becomes then:

$$W(\alpha_1, \alpha_2, ..., \alpha_m) = W((\psi - \alpha_2 y^2)y^1, \alpha_2, ..., \alpha_m) \tag{3.42}$$

Since $\alpha_3, \alpha_4, ..., \alpha_m$ are by now constants, the above function is quadratically dependent from α_2 and a maximum can be found easily.

Once an optimal value for α_2 is found, it is necessary to bound it because of the constraints coming from 3.42 and 3.38: two bounds, L and H, are then found from easy mathematical operations and depending on the values of y^1 and y^2. The α_2 is then clipped as follows:

$$\alpha_2^{new} = \begin{cases} H & \text{if } \alpha_2 \geq H \\ \alpha_2 & \text{if } L < \alpha_2 < H \\ L & \text{if } \alpha_2 \leq L \end{cases} \tag{3.43}$$

The new value of α_2 is then used in combination of equation 3.43 to find also the optimal value of α_1.

The iterations are repeated until a certain time expires, or a tolerance for every α_i is reached. The heuristic methods to find two multipliers will not be described here since they are quite straightforward.[133]

The code implementation carried out in this dissertation is inspired by the one developed in the National Taiwan University, Taipei, department of Computer Science [3]. The last step now is to take a look at how the already calibrated algorithm (that means once all the α_i are found) classifies new instances coming from the feature extraction.

3.5.1 LIBSVM

A classification task usually involves separating data into training and testing sets. Each instance in the training set contains one "target value" (i.e. the class labels) and several "attributes" (i.e. the features or observed variables). The goal of SVM is to produce a model (based on the training data) which predicts the target values of the test data given only the test data attributes.

Given a training set of instance-label pairs (x_i, y_i), i=1,...,l where $x_i \in R^n$ and y $\in \{1, -1\}^l$, the training vectors x_i are mapped into a higher (maybe infinite) dimensional space by the function ϕ . SVM finds a linear separating hyperplane with the maximal margin in this higher dimensional space. C >0 is the penalty parameter of the error term. Furthermore, $K(x_i, x_j) = \phi(x_i)^T \phi(x_j)$ is called the kernel function. Though new kernels are being proposed by researchers, beginners may find in SVM books the following four basic kernels:

- linear: $K(x_i, x_j) = {x_i}^T x_j$.
- polynomial: $K(x_i, x_j) = (\gamma {x_i}^T x_j + r)^d, \gamma > 0$.
- radial basis function (RBF): $K(x_i, x_j) = exp(-\gamma ||x_i - x_j||^2), \gamma > 0$.
- sigmoid: $K(x_i, x_j) = tanh(\gamma {x_i}^T x_j + r)$

Here, γ, r, and d are kernel parameters.

Used Procedure

- Transform data to the format of an SVM package
- Conduct simple scaling on the data
- Consider the RBF kernel $K(x, y) = e^{-\gamma \|x-y\|^2}$
- Use cross-validation to find the best parameter C and γ
- Use the best parameter C and γ to train the whole training set (The best parameter might be affected by the size of data set but in practice the one obtained from cross-validation is already suitable for the whole training set.)
- Test

We discuss this procedure in detail in the following sections.

Scaling Scaling before applying SVM is very important. [134] explains the importance of this and most of considerations also ap- ply to SVM. The main advantage of scaling is to avoid attributes in greater numeric ranges dominating those in smaller numeric ranges. Another advantage is to avoid numerical difficulties during the calculation. Because kernel values usually depend on the inner products of feature vectors, e.g. the linear kernel and the polynomial ker- nel, large attribute values might cause numerical problems. Linearly scaling each attribute to the range $[-1, +1]$ or $[0, 1]$ is recommended. Of course we have to use the same method to scale both training and testing data. For example, suppose that we scaled the first attribute of training data from $[-10, +10]$ to $[-1, +1]$. If the first attribute of testing data lies in the range $[-11, +8]$,we must scale the testing data to$[-1.1, +0.8]$.

Model Selection In general, the RBF kernel is a reasonable first choice. This kernel nonlinearly maps samples into a higher dimensional space so it, unlike the linear kernel, can handle the case when the relation between class labels and attributes is nonlinear. Furthermore, the linear kernel is a special case of RBF since the linear kernel with a penalty parameter $\tilde{C}$ has the same performance as the RBF kernel with some parameters (C,γ). In addition, the sigmoid kernel behaves like RBF for certain parameters.

The second reason is the number of hyperparameters which influences the complexity of model selection. The polynomial kernel has more hyperparameters than the RBF kernel.

Finally, the RBF kernel has fewer numerical difficulties. One key point is $0 < K_{ij} <= 1$ in contrast to polynomial kernels of which kernel values may go to infinity $(\gamma x_i^T x_j + r > 1)$ or zero $(\gamma x_i^T x_j + r < 1)$ while the degree is large.

There are some situations where the RBF kernel is not suitable. In particular, when the number of features is very large, one may just use the linear kernel, not in our case.

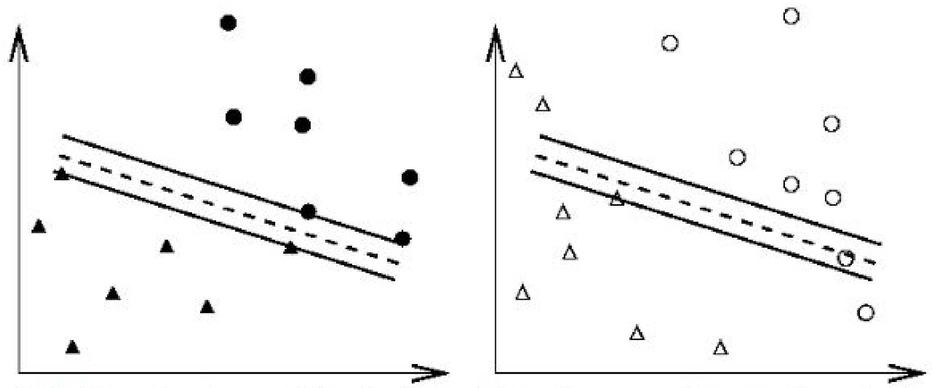
Cross-validation and Grid-search There are two parameters for an RBF kernel: C and γ . We do not know beforehand which C and γ are best for a given problem; consequently some kind of model selection (parameter search) must be done. The goal is to identify good (C,γ) so that the classifier can accurately predict unknown data (i.e. testing data). Note that it may not be useful to achieve high training accuracy (i.e. a classifier which accurately predicts training data whose class labels are indeed known). As discussed above, a common strategy is to separate the data set into two parts, of which one is considered unknown. The prediction accuracy obtained from the "unknown" set more precisely reflects the performance on classifying an independent data set. An improved version of this procedure is known as cross-validation.

In υ-fold cross-validation, we first divide the training set into υ subsets of equal size. Sequentially one subset is tested using the classifier trained on the remaining $\upsilon - 1$ subsets. Thus, each instance of the whole training set is predicted once so the cross-validation accuracy is the percentage of data which are correctly classified.

The cross-validation procedure can prevent the overfitting problem.

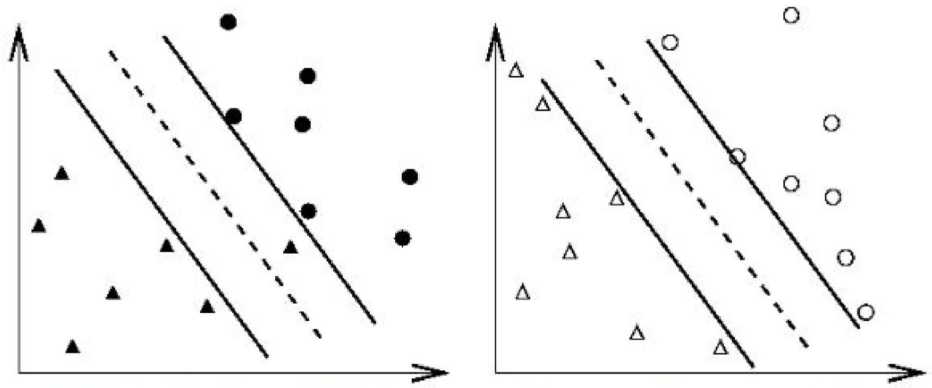
Figure 3.21 represents a binary classification problem to illustrate this issue. Filled circles and triangles are the training data while hollow circles and triangles are the testing data. The testing accuracy of the classifier in Figures 1a and 1b is not good since it overfits the training data. If we think of the training and testing data in Figure 1a and 1b as the training and validation sets in cross-validation, the accuracy is not good. On the other hand, the classifier in 1c and 1d does not overfit the training data and gives better cross-validation as well as testing accuracy.

We used a "grid-search" on C and γ using cross-validation. Various pairs of (C,γ) values are tried and the one with the best cross-validation accuracy is picked. We found that trying exponentially growing sequences of C and γ is a practical method to identify good parameters (for example, C= $2^{-5}, 2^{-3}, ..., 2^{15}, \gamma = 2^{-15}, 2^{-13}, ..., 2^{3}$).



(a) Training data and an overfitting classifier

(b) Applying an overfitting classifier on testing data



(c) Training data and a better classifier

(d) Applying a better classifier on testing data

Figure 3.21: An overfitting classifier and a better classifier [3]

The grid-search is straightforward but seems naive. In fact, there are several advanced methods which can save computational cost by, for example, approximating the cross-validation rate. However, there are two motivations why we prefer the simple grid-search approach.

One is that, psychologically, we may not feel safe to use methods which avoid doing an exhaustive parameter search by approximations or heuristics. The other reason is that the computational time required to find good parameters by grid-search is not much more than that by advanced methods since there are only two parameters. Furthermore, the grid-search can be easily parallelized because each (C,γ) is independent. Many of advanced methods are iterative processes, e.g. walking along a path, which can be hard to parallelize.

Since doing a complete grid-search may still be time-consuming, we used a coarse grid first. After identifying a “better” region on the grid, a finer grid search on that region can be conducted. [3]

3.6 Lock/Unlock detection

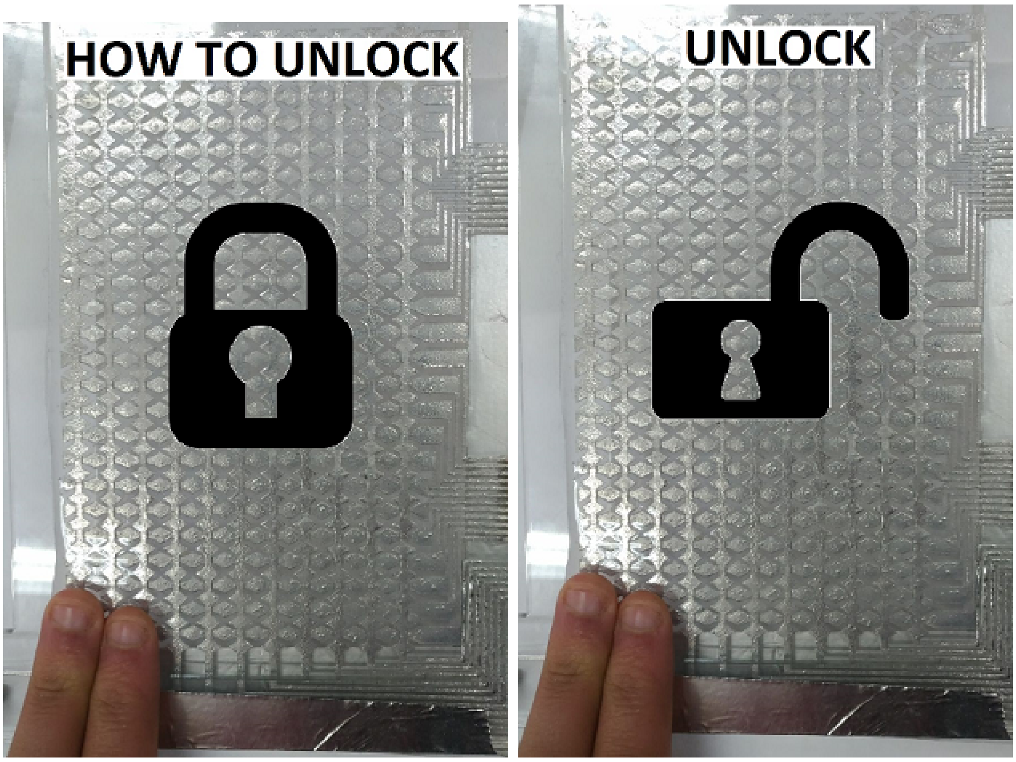


Figure 3.22: Lock/Unlock

An appreciated property of the real time application is for sure the possibility to "Lock" the classification of the gestures, in order to allow the user to move freely. To achieve this, two possible expedients could be adopted:

- To ask the user to do it manually, by pressing a physical button. This technique is the most reliable, and assures a sure locking of the gesture detection.
- To add a 6th gesture to the previous ones, that when classified, it locks/unlocks the detection of the other 5 hand movements. This is for sure a more comfortable solution, since can be a "secret" gesture that lock the system, for this purpose touching the sensor is preferible to avoid misclassification with the other gesture, use this solution is not that difficult because the differences between touching the sensor and just approaching are huge.

Since the purpose of this project is to create a hand gestures classifier, the second solution is preferable.

Chapter 4

Tests and Results

In this chapter tests will be carried out to assess the algorithm performances in different applications and configurations:

1. **Pure algorithm performances**: A classification performance evaluation, using a training set of 7 instances for each gesture. Moreover, the calibration of the algorithm is done before every usage, and so session independence is not assessed.

   This test is carried out to verify the algorithm functioning in an ideal situation, verifying if there is a limit in the classification performances for the SVM.

2. **Session independence**: To assess the possibility of not recalibrating the classifier at every utilization, speeding up the usage of the system. The algorithm is trained once, the model saved and the board is turned off; after a certain period, the system is used again and the board is turned on; gestures are then performed, without requesting a new calibration. A comparison is made between performances immediately after the calibration (Test 1) and after a certain period.

Every test is repeated for 3 different subjects, one expert, one intermediate, someone who knows something about the system and what we did, and one basic user who do not know anything about the usage of the system.

The acquisition threshold (Section 3.4) which activates the features extraction is set to 40.

The maximum observation window length is fixed to 100 iterations (ADC conversions), which translates in $\sim 1s$, and during this window, no more than 3300 values are taken(100values∗33sensors) and 66 features are extracted.

4.1 Rules for a good utilization

In this section some suggestions about how to perform the 5 gestures are given, in order to help the user in the training and reach good results in terms of classification accuracy.

1. With this sensor all the gestures must be performed with the left hand to avoid interferences with the connections on the right of the sensor.With a shield this is not anymore necessary.

2. Every time that a gesture is performed is important the approaching and the removing part. The hand must approach the sensor enough to overpassing the threshold, perform the gesture (if is dynamic) and removing the hand to stop the acquisition.

3. The static gesture ("OK") should be performed clearly, spreading the fingers well, the gesture measuring should end removing the hand with the gesture still performed. The "OK" can be seen in Figure 4.1:

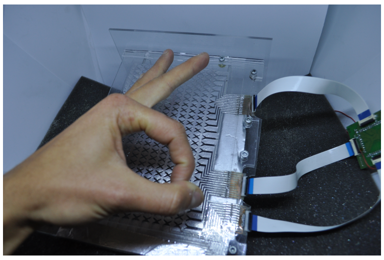


Figure 4.1: OK gesture

This is done to differentiate clearly the "OK" gesture from the others that could be similar in some sessions.

4. The "Zoom in" and "Zoom out" gestures should not be performed pretty fast, also, the position of hand should be further than the fingers performing the gesture (Figure 4.2). This helps to separate accurately the "Zoom" from the "OK" and "Close-Open".

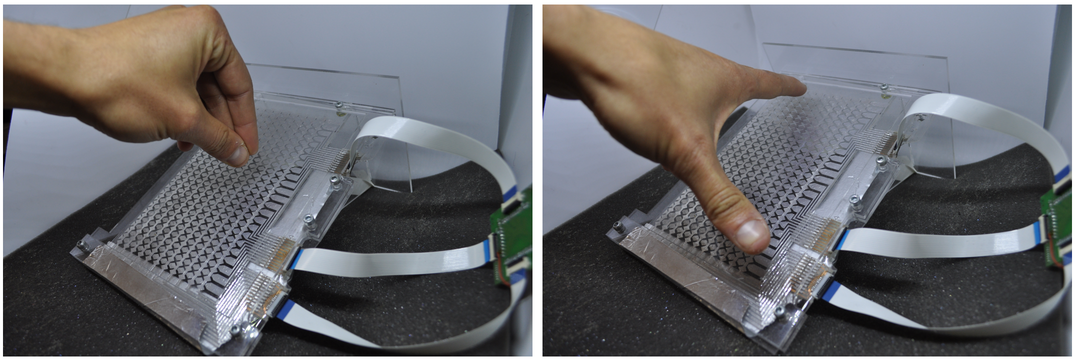


Figure 4.2: Zoom gesture

5. The "Open-Close" and "Close-Open" gestures should not be performed pretty fast, in order to activate the gesture measurement with the initial peak the hand should approach open in the "Open-Close" and close in the "Close-Open".

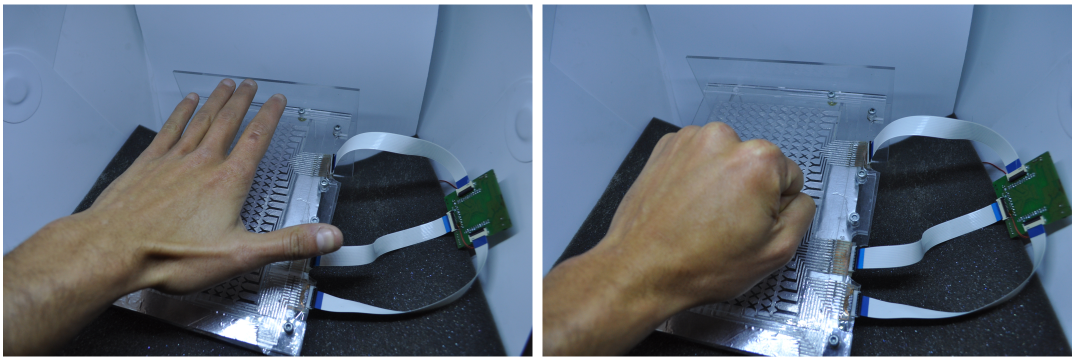


Figure 4.3: Open-Close gesture

6. To perform the "Lock" gesture, we use a simple "combination", to press the bottom left corner as shown in Figure 4.4. This is asked to differentiate greatly this gesture from the random signal caused by other body movements, and so to allow the user to move freely after the system was locked or to avoid unwanted uses by people.

Figure 4.4: Lock gesture

In addition to these requirements, we suggest to think about a immaginary plan, close to the sensor, where perform the gestures as on a touchpad.

In the tests there will be differentiated the results for an expert user and the others not expert. Nevertheless, even for the beginner, some time was left to train (less than half an hour) and get used to the gesture performing.

4.2 Pure algorithm performances

7 instances of each gesture are given for the training of the algorithm.

After the calibration, 20 instances of each gesture are performed, changing type every 2 of the same kind, in order to not let the user to adapt to the movement and simulate a more real application. For every subject, a confusion matrix is built, which indicates the classification outputs for each of the 5 classes.

	OK	ZI	ZO	CO	OC
OK	20	0	0	0	0
ZI	1	18	0	0	1
ZO	0	0	20	0	0
CO	1	0	0	19	0
OC	1	0	0	0	19

	OK	ZI	ZO	CO	OC
OK	100%	0	0	0	0
ZI	5%	90%	0	0	5%
ZO	0	0	100%	0	0
CO	5%	0	0	95%	0
OC	5%	0	0	0	95%

Table 4.1: Test 1, Confusion matrix: User 1 (expert)

	OK	ZI	ZO	CO	OC
OK	20	0	0	0	0
ZI	0	20	0	0	0
ZO	0	3	16	0	1
CO	1	0	0	19	0
OC	0	0	0	1	19

	OK	ZI	ZO	CO	OC
OK	100%	0	0	0	0
ZI	0	100%	0	0	0
ZO	0	15%	80%	0	5%
CO	5%	0	0	95%	0
OC	0	0	0	5%	95%

Table 4.2: Test 1, Confusion matrix: User 2 (intermediate)

	OK	ZI	ZO	CO	OC
OK	16	0	0	2	2
ZI	0	17	0	1	2
ZO	0	1	19	0	0
CO	0	0	0	19	1
OC	2	1	0	0	17

	OK	ZI	ZO	CO	OC
OK	80%	0	0	10%	10%
ZI	0	85%	0	5%	10%
ZO	0	5%	95%	0	0
CO	0	0	0	95%	5%
OC	10%	5%	0	0	85%

Table 4.3: Test 1, Confusion matrix: User 3 (beginner)

Results It can be stated from tables that the algorithm classifies well the 5 gestures in an ideal configuration, and so no upper limits on the performances should be taken in consideration.

4.3 Session independence

7 instances of each gesture are given for the training of the algorithm, since real-time constraints are present and hence the training of the algorithm ought to be fast (with 5 gestures it takes $\sim 30sec$ + the time to perform the training set).

As before, after the calibration, 20 instances of each gesture are performed, changing type every 2 of the same kind, in order to not let the user to adapt to the movement and simulate a prostheses application. For every subject, a confusion matrix is built.

The algorithm is trained in one session, and tested in other 2 sessions, spaced between each other more than 1 day, turning off/on the system.

	OK	**ZI**	**ZO**	**CO**	**OC**
OK	20	0	0	0	0
ZI	1	19	0	0	0
ZO	0	1	19	0	0
CO	0	0	0	20	0
OC	1	1	0	0	18

	OK	**ZI**	**ZO**	**CO**	**OC**
OK	100%	0	0	0	0
ZI	5%	95%	0	0	0
ZO	0	5%	95%	0	0
CO	0	0	0	100%	0
OC	5%	5%	0	0	90%

Table 4.4: Test 2, Confusion matrix: User 1 (expert), session 1

	OK	**ZI**	**ZO**	**CO**	**OC**
OK	17	2	0	0	1
ZI	0	20	0	0	0
ZO	1	0	18	1	0
CO	0	0	1	19	0
OC	0	0	0	0	20

	OK	**ZI**	**ZO**	**CO**	**OC**
OK	85%	10%	0	0	5%
ZI	0	100%	0	0	0
ZO	5%	0	90%	5%	0
CO	0	0	5%	95%	0
OC	0	0	0	0	100%

Table 4.5: Test 2, Confusion matrix: User 1 (expert), session 2

	OK	**ZI**	**ZO**	**CO**	**OC**
OK	18	2	0	0	0
ZI	0	18	0	0	2
ZO	0	2	18	0	0
CO	1	0	1	18	0
OC	0	0	0	0	20

	OK	**ZI**	**ZO**	**CO**	**OC**
OK	90%	10%	0	0	0
ZI	0	90%	0	0	10%
ZO	0	10%	90%	0	0
CO	5%	0	5%	90%	0
OC	0	0	0	0	100%

Table 4.6: Test 2, Confusion matrix: User 1 (expert), session 3

	OK	ZI	ZO	CO	OC
OK	20	0	0	0	0
ZI	0	20	0	0	0
ZO	0	3	16	0	1
CO	1	0	0	19	0
OC	0	0	0	1	19

	OK	ZI	ZO	CO	OC
OK	100%	0	0	0	0
ZI	0	100%	0	0	0
ZO	0	15%	80%	0	5%
CO	5%	0	0	95%	0
OC	0	0	0	5%	95%

Table 4.7: Test 2, Confusion matrix: User 2 (intermediate), session 1

	OK	ZI	ZO	CO	OC
OK	16	0	0	4	0
ZI	0	20	0	0	0
ZO	0	1	19	0	0
CO	0	0	0	20	0
OC	0	0	0	0	20

	OK	ZI	ZO	CO	OC
OK	80%	0	0	20%	0
ZI	0	100%	0	0	0
ZO	0	5%	95%	0	0
CO	0	0	0	100%	0
OC	0	0	0	0	100%

Table 4.8: Test 2, Confusion matrix: User 2 (intermediate), session 2

	OK	ZI	ZO	CO	OC
OK	19	0	0	0	1
ZI	0	19	0	0	1
ZO	2	1	17	0	0
CO	0	0	0	20	0
OC	0	0	0	0	20

	OK	ZI	ZO	CO	OC
OK	95%	0	0	0	5%
ZI	0	95%	0	0	5%
ZO	10%	5%	85%	0	0
CO	0	0	0	100%	0
OC	0	0	0	0	100%

Table 4.9: Test 2, Confusion matrix: User 2 (intermediate), session 3

	OK	ZI	ZO	CO	OC
OK	16	0	0	2	2
ZI	0	17	0	1	2
ZO	0	1	19	0	0
CO	0	0	0	19	1
OC	2	1	0	0	17

	OK	ZI	ZO	CO	OC
OK	80%	0	0	10%	10%
ZI	0	85%	0	5%	10%
ZO	0	5%	95%	0	0
CO	0	0	0	95%	5%
OC	10%	5%	0	0	85%

Table 4.10: Test 3, Confusion matrix: User 3 (beginner), session 1

	OK	ZI	ZO	CO	OC
OK	19	0	0	0	1
ZI	0	17	2	1	0
ZO	0	1	19	0	0
CO	0	0	0	20	0
OC	2	0	0	1	17

	OK	ZI	ZO	CO	OC
OK	95%	0	0	0	5%
ZI	0	85%	10%	5%	0
ZO	0	5%	95%	0	0
CO	0	0	0	100%	0
OC	10%	0	0	5%	85%

Table 4.11: Test 3, Confusion matrix: User 3 (beginner), session 2

	OK	ZI	ZO	CO	OC
OK	18	0	0	0	2
ZI	0	20	0	0	0
ZO	0	1	18	1	0
CO	0	0	0	20	0
OC	4	0	0	0	16

	OK	ZI	ZO	CO	OC
OK	90%	0	0	0	10%
ZI	0	100%	0	0	0
ZO	0	5%	90%	5%	0
CO	0	0	0	100%	0
OC	20%	0	0	0	80%

Table 4.12: Test 3, Confusion matrix: User 3 (beginner), session 3

Results Comparing these results with the ones from the previous test, it can be said that the session independence is a property of this classification system. The results indeed do not change significantly, even if the algorithm was trained a day before the utilization. The system could then be used without training the SVM at every usage; when the classification does not work well anyway, a recalibration is suggested.

Chapter 5

Conclusions and future developments

In this chapter conclusions about the developed work are evaluated. We assessed if the classification system gives satisfying results in terms of gestures detection and session independence. After that, some possible future developments are initially given, which could increase the capabilities, reliability, utility and comfort of the system as a general purpose application.

5.1 Conclusions

To differentiate sufficiently the 5 needed gestures ("OK", "Zoom-in", "Zoom-out", "Close-Open" and "Open-Close") between each other, a high-dimensional feature space was adopted, where every gesture was described by 66 subsequently mean values; the used classifier was then the support vector machine.

The main question that was asked in this specific design was indeed if the system could achieve sufficient results in terms of classification reliability and real-time constraints; from the tests carried out in the previous chapter we can say that this goal was reached.

The 5 gestures were in fact classified correctly with high percentage (overall $>=$ 94%) in the Test 1 either for User 1 (expert) and User 2 (intermediate); for User 3 (beginner) the percentage is $\sim$ 88% due to the inexperience of beginner user (results are better with session independence).

The real-time constraints were as well respected since the above satisfying results were obtained with a small training set and feature space dimensions, which brought to a reasonable processing time for calibration ($<$1 min) and a fast system response to a gesture performing ($\sim 1s$, comprehensive of the gesture length, calculated with a Timer).

In addition to that, the classifier (Support vector machine) did not need to be calibrated at every usage, since Test 2 demonstrated the session independence of the design.

The System locking instead assured a total disturbances rejection, indeed 0% of the other limb movements were classified as a "Lock" gesture, maintaining so the system locked. Moreover, the addition of the 6th gesture (called "Lock") did not affect significantly the normal performances of the classification, since the others 5 gestures are never classified wrongly as "Lock" since the "Lock" gesture must be performed touching the sensor.

If we add too many gestures the classify is not able to recognize so well every gestures, this is a limitation for this system.

5.2 Future developments

Although reached results are satisfying, some important improvements can be carried out on the classification system:

- The dimension of the sensor could be bigger. This to allow to use both hands at same time or just covered a larger surface, with a bigger sensor the introduction of the centre of mass is necessary, since the numbers of values increas the system should acquire just around the hand, so acquiring just a predifinied number of sensors close to the centre of mass.
- All the connections should be shielded to avoid interferences with the arm or other objects or body parts.
- Since the differences between touch and approach are noticeable, theoretically the system is able to recorgnize infinite gestures combination of touch and approach.
- With some improvement one could think of using in different ways such as to translate sign language, playing videogames or fix a bigger sensor over a wall to recognize human-body gestures.

Appendix A

Feature extraction code

```

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.IO;



namespace CaptureBaseProgram
{
public partial class Form1 : Form
{
public const int samegesture = 250;
public const int window = 100;
public const int gesti = 5;
const int p = 3;
const int pred = 2;
int[,] closetrain = new int[samegesture, sensor * pred];

float[] closemean = new float[sensor];
public const int SIZE_X = 23;
public const int SIZE_Y = 10;
public const int sensor = 33;
public const int n_to_sum = 1;

public SerialPortClass Serial = new SerialPortClass();
```

```
int window_acq = 0;
public byte[] packet = new byte[500];
float[] data = new float[sensor];
int[] scan = new int[sensor];
float[,] close = new float[window+1, sensor];
float[] snap = new float[2 * sensor];
int[]  cnt_sample_vec = new int[window+1];
int j_min_left = 0, j_min_right = 0;
int cnt_sample = 0;
int flag = 0, i = 1, data_ready = 0;
int acquire = 0;
float mean = 0;
float sum = 0;
int gestures = 0, train = 5;
int prova = 0,  numgest = 0, trained = 0;
int data_acquaried = 0;

float[] CenterMass = new float[2], CenterMassacq = new float[2];
float[] sum_sensor_deltas = new float[sensor];
int locked = 0;
string str = "";
string line = "";
private float[] array1 = { 0f, 0.9f, 1.8f, 2.7f, 3.6f, 4.5f, 5.4f, 6.3f, 7.2f, 8.1f };
private float[] array2 = { 0f, 0.7f, 1.4f, 2.1f, 2.8f, 3.5f, 4.2f, 4.9f, 5.6f, 6.3f, 7.0f, 7.7f, 8.4f, 9.1f, 9.8f, 10.5f, 11.2f, 11.9f, 12.6f, 13.3f, 14f, 14.7f, 15.4f };
public const int X_Dim = 10, Y_Dim = 23;
int[] i_gest_snap = new int [2];

public Form1()
{
InitializeComponent();
}

private void Form1_Load(object sender, EventArgs e)
{
// pictureBox1.Image = new Bitmap(SIZE_X, SIZE_Y);
try
{
Serial.Open(serialPort1);
}
catch
{
string msg = "Serial port " + serialPort1.PortName + " fail to open\r\n Please check settings/dongle";
MessageBox.Show(msg);
}

```

```
}

private void button1_Click(object sender, EventArgs e)
{
serialPort1.PortName = textBox1.Text;
try
{
Serial.Open(serialPort1);
string msg = "Serial port " + serialPort1.PortName + " sucess";
MessageBox.Show(msg);
}
catch
{
string msg = "Serial port " + serialPort1.PortName + " fail to open↩
    \r\n Please check settings/dongle";
MessageBox.Show(msg);
}
}

private void timer1_Tick(object sender, EventArgs e)
{
if (mean > 40 && acquire == 0)
{
timer2.Enabled = true;
}
else if (mean < 40 && acquire == 1)
{
timer2.Enabled = false;
for (int k = 0; k < sensor; k++)
{
close[i, k] = 0;
}
cnt_sample_vec[i] = cnt_sample + 1;
window_acq = i + 1;
i = 1;
acquire = 2;
}

if (acquire == 2)
{
writetxt();
index_snap();
}

if(data_acquaried==1)
{
str = "";
line = "";
for (int k = 0; k < sensor; k++)
```

```
{
line = line + (k + 1).ToString() + ":" + ((close[i_gest_snap[0], k] + close[i_gest_snap[0] + 1, k] + close[i_gest_snap[0] + 2, k]) / 3).ToString() + " ";
}
for (int k = 0; k < sensor; k++)
{
line = line + (sensor + k + 1).ToString() + ":" + ((close[i_gest_snap[1], k] + close[i_gest_snap[1] - 1, k] + close[i_gest_snap[1] - 2, k]) / 3).ToString() + " ";
}
str = str + (gestures).ToString() + " " + line + "\r\n";
line = "";
}

if (train == 0 && trained == 0 && data_acquaried == 1)
{
textBox39.Text = "ACQUIRING TRAIN...";
if (numgest < samegesture)
{
data_acquaried = 0;
saveFileDialog3.FileName = "C:\\Users\\Matteo\\Dropbox\\Matteo\\AAmeccatronica\\TESI-SVM\\CaptureBaseProgram\\record\\CaptureBaseProgram\\bin\\Debug\\snap_train.txt";
FileStream rec_snap = new FileStream(saveFileDialog3.FileName, FileMode.Append);
StreamWriter sw_rec = new StreamWriter(rec_snap);
sw_rec.Write(str);
sw_rec.Close();
rec_snap.Close();
numgest++;
textBox40.Text = numgest.ToString();
if (numgest == samegesture)
{
if (gestures == gesti)
{
gestures = 1;
}
else
{
gestures++;
}
pictureBox1.Refresh();
}
}
}
else if (train == 1 && data_acquaried == 1)
{
textBox39.Text = "ACQUIRING TEST...";
```

```
if (numgest < samegesture)
{
data_acquaried = 0;
saveFileDialog3.FileName = "C:\\Users\\Matteo\\Dropbox\\Matteo\\AAAmeccatronica\\TESI-SVM\\CaptureBaseProgram\\record\\CaptureBaseProgram\\bin\\Debug\\snap_test.txt";
FileStream rec_snap = new FileStream(saveFileDialog3.FileName, FileMode.Append);
StreamWriter sw_rec = new StreamWriter(rec_snap);
sw_rec.Write(str);
sw_rec.Close();
rec_snap.Close();
numgest++;
textBox40.Text = numgest.ToString();
if (numgest == samegesture)
{
if (gestures == gesti)
{
gestures = 1;
}
else
{
gestures++;
}
pictureBox1.Refresh();
}
}
}
else if (train == 2 && trained == 0)
{
textBox39.Text = "TRAINING...";
data_acquaried = 0;
var process = System.Diagnostics.Process.Start("CMD.exe", "/C easy.py snap_train.txt");
trained = 1;
}

else if (locked == 0 && trained == 1 && train == 3 && data_acquaried == 1)
{
textBox39.Text = "WAITING GESTURE...";
data_acquaried = 0;
saveFileDialog3.FileName = "C:\\Users\\Matteo\\Dropbox\\Matteo\\AAAmeccatronica\\TESI-SVM\\CaptureBaseProgram\\record\\CaptureBaseProgram\\bin\\Debug\\snap_gesture.txt";
FileStream rec_snap = new FileStream(saveFileDialog3.FileName, FileMode.Create);
StreamWriter sw_rec = new StreamWriter(rec_snap);
sw_rec.Write(str);
```

```
sw_rec.Close();
rec_snap.Close();
var process = System.Diagnostics.Process.Start("CMD.exe", "/C easyprova.py snap_train.txt snap_gesture.txt");
process.WaitForExit();
using (TextReader reader = File.OpenText("snap_gesture.txt.txt"))
{
int x = int.Parse(reader.ReadLine());
print_image(x);
if (x == 0)
{
locked = 1;
pictureBox1.Image = Image.FromFile(@"C:\Users\Matteo\Dropbox\Matteo\AAAmeccatronica\TESI-SVM\CaptureBaseProgram\record\CaptureBaseProgram\bin\Debug\immagini\lock.png");
}
}
}
else if (train == 4)
{
if (trained == 1)
{
var process = System.Diagnostics.Process.Start("CMD.exe", "/K easyprova.py snap_train.txt snap_test.txt");
train = 3;
}
else if (trained == 0)
{
var process = System.Diagnostics.Process.Start("CMD.exe", "/K easy.py snap_train.txt snap_test.txt");
trained = 1;
train = 3;
}
}
else if (locked == 1 && data_acquaried == 1)
{
textBox39.Text = "SYSTEM LOCKED...";
data_acquaried = 0;
saveFileDialog3.FileName = "C:\\Users\\Matteo\\Dropbox\\Matteo\\AAAmeccatronica\\TESI-SVM\\CaptureBaseProgram\\record\\CaptureBaseProgram\\bin\\Debug\\snap_gesture.txt";
FileStream rec_snap = new FileStream(saveFileDialog3.FileName, FileMode.Create);
StreamWriter sw_rec = new StreamWriter(rec_snap);
sw_rec.Write(str);
sw_rec.Close();
rec_snap.Close();
var process = System.Diagnostics.Process.Start("CMD.exe", "/C easyprova.py snap_train.txt snap_gesture.txt");
```

```
process.WaitForExit();
using (TextReader reader = File.OpenText("snap_gesture.txt.txt"))
{
int x = int.Parse(reader.ReadLine());
if (x == 0)
{
locked = 0;
pictureBox1.Image = Image.FromFile(@"C:\Users\Matteo\Dropbox\Matteo\AAAmeccatronica\TESI-SVM\CaptureBaseProgram\record\CaptureBaseProgram\bin\Debug\immagini\unlock.png");
}
}
}
if (train < 2 && trained == 0)
{
guessgesture(gestures);
}
CalculateCenterOfMass();
pictureBox1.Refresh();
pictureBox2.Refresh();
writeonscreen(data);
textBox62.Text = cnt_sample.ToString();
}

private void timer2_Tick(object sender, EventArgs e)
{
if (data_ready == 1 && i < window)
{
acquire = 1;
data_ready = 0;
if (i == 1)
{
for (int k = 0; k < sensor; k++)
{
close[0, k] = 0;
cnt_sample_vec[0] = cnt_sample - 1;
}
}
for (int k = 0; k < sensor; k++)
{
close[i, k] = data[k];
cnt_sample_vec[i] = cnt_sample;
}
i++;
}
else if (i == window)
{
for (int k = 0; k < sensor; k++)
{
```

```
close[i, k] = 0;
}
cnt_sample_vec[i] = cnt_sample + 1;
window_acq = i + 1;
i = 1;
acquire = 2;
timer2.Enabled = false;
}
}

private void serialPort1_DataReceived(object sender, System.IO.Ports.SerialDataReceivedEventArgs e)
{
packet = Serial.ReadFromSerial();
if (packet != null)
{
int box;
cnt_sample = packet[1];
cnt_sample = cnt_sample << 8;
cnt_sample = cnt_sample + packet[0];
for (int i = 0; i < sensor; i++)
{
box = packet[2 * i +1+2];
box = box << 8;
box = box + packet[2 * i  + 2];

if (flag < 5)
{
scan[i] = box;
if (i == sensor - 1)
{
flag++;
}
}
else
{
if (box - scan[i] < 0)
{
data[i] = box - scan[i];
}
else
{
data[i] = box - scan[i];
}

if (i == 0)
{
sum = 0;
}
```

```
sum += data[i];
mean = sum / sensor;
}
}
Serial.datapoints.Clear();
Serial.state_machine = 0;
data_ready=1;
}
}



private void index_snap()
{

float[,] meanvector = new float[window_acq / n_to_sum, sensor];
float[,] delta = new float[window_acq / n_to_sum, sensor];
for (int h = 0; h < sensor; h++)
{
int count = 0, j = 0;

while (j < window_acq / n_to_sum)
{
if (count < window_acq)
{
for (int n = 0; n < n_to_sum; n++)
{
meanvector[j, h] += close[count, h];
count++;
}
}
meanvector[j, h] = meanvector[j, h] / n_to_sum;
j++;
}

for (int l = 1; l < window_acq / n_to_sum; l++)
{
delta[l, h] = Math.Abs(meanvector[l, h] - meanvector[l - 1, h]);
}
}
float[] sum_deltas = new float[window_acq / n_to_sum];
for (int j = 0; j < window_acq / n_to_sum; j++)
{
for (int h = 0; h < sensor; h++)
{
sum_deltas[j] += delta[j, h];

}
}
```

```
float[] sum_sensor = new float[window_acq / n_to_sum];
for (int j = 0; j < window_acq / n_to_sum; j++)
{
for (int h = 0; h < sensor; h++)
{
sum_sensor[j] += meanvector[j, h];

}
}
int first = 0;
for (int j = 0; j < window_acq / n_to_sum; j++)
{
if (sum_sensor[j] > 2000 && first==0)
{
j_min_left = j;
first = 1;
}
if(sum_sensor[j]<2000 && first==1)
{
j_min_right = j - 1;
first = 0;
}
}




i_gest_snap[0] = j_min_left * n_to_sum;
i_gest_snap[1] = j_min_right * n_to_sum;



data_acquaried = 1;
acquire = 0;

}

private void writetxt()
{
string str = "";
string line = "";

for (int m = 0; m < window_acq; m++)
{
for (int k = 0; k < sensor; k++)
{

line = line + close[m, k].ToString() + " ";

```

```
}
str = str + cnt_sample_vec[m].ToString() + " " + line + "\r\n";
line = "";
}

saveFileDialog3.FileName = "C:\\Users\\Matteo\\Dropbox\\Matteo\\AAmeccatronica\\TESI-SVM\\CaptureBaseProgram\\record\\CaptureBaseProgram\\bin\\Debug\\closeexcell.txt";
FileStream rec_file = new FileStream(saveFileDialog3.FileName, FileMode.Append);

StreamWriter sw_record = new StreamWriter(rec_file);


sw_record.Write(str);

sw_record.Close();
rec_file.Close();
}

private void print_image(int x)
{


if (x == 1 && locked == 0)
{
textBox42.Text = "WELL DONE";
pictureBox1.Image = Image.FromFile(@"C:\Users\Matteo\Dropbox\Matteo\AAmeccatronica\TESI-SVM\CaptureBaseProgram\record\CaptureBaseProgram\bin\Debug\immagini\welldone.jpg");
}
else if (x == 2 && locked == 0)
{
textBox42.Text = "ZOOM IN";
pictureBox1.Image = Image.FromFile(@"C:\Users\Matteo\Dropbox\Matteo\AAmeccatronica\TESI-SVM\CaptureBaseProgram\record\CaptureBaseProgram\bin\Debug\immagini\zoom-in.jpg");
}
else if (x == 3 && locked == 0)
{
textBox42.Text = "ZOOM OUT";
pictureBox1.Image = Image.FromFile(@"C:\Users\Matteo\Dropbox\Matteo\AAmeccatronica\TESI-SVM\CaptureBaseProgram\record\CaptureBaseProgram\bin\Debug\immagini\zoom-out.jpg");
}
else if (x == 8 && locked == 0)
{
textBox42.Text = "SWIPE LEFT -> RIGHT";
```

```
pictureBox1.Image = Image.FromFile(@"C:\Users\Matteo\Dropbox\Matteo\AAAmeccatronica\TESI-SVM\CaptureBaseProgram\record\CaptureBaseProgram\bin\Debug\immagini\swipe-right.jpg");
}
else if (x == 7 && locked == 0)
{
textBox42.Text = "SWIPE RIGHT -> LEFT";
pictureBox1.Image = Image.FromFile(@"C:\Users\Matteo\Dropbox\Matteo\AAAmeccatronica\TESI-SVM\CaptureBaseProgram\record\CaptureBaseProgram\bin\Debug\immagini\swipe2.jpg");
}
else if (x == 5 && locked == 0)
{
textBox42.Text = "CLOSE -> OPEN";
pictureBox1.Image = Image.FromFile(@"C:\Users\Matteo\Dropbox\Matteo\AAAmeccatronica\TESI-SVM\CaptureBaseProgram\record\CaptureBaseProgram\bin\Debug\immagini\close-open.jpg");
}
else if (x == 4 && locked == 0)
{
textBox42.Text = "OPEN -> CLOSE";
pictureBox1.Image = Image.FromFile(@"C:\Users\Matteo\Dropbox\Matteo\AAAmeccatronica\TESI-SVM\CaptureBaseProgram\record\CaptureBaseProgram\bin\Debug\immagini\open-close.jpg");
}

else if (x == 0)
{
textBox42.Text = "SYSTEM LOCK";

pictureBox1.Image = Image.FromFile(@"C:\Users\Matteo\Dropbox\Matteo\AAAmeccatronica\TESI-SVM\CaptureBaseProgram\record\CaptureBaseProgram\bin\Debug\immagini\howlock.jpeg");

}

pictureBox1.Refresh();
}

}
}
```

Bibliography

[1] , "Capacitive touch sensor system (second-generation capacitive touch technology)." `https://www.renesas.com/en-eu/solutions/key-technology/human-interface/touch-sensor-system2.html`.

[2] , "Myo armband." `https://www.myo.com/`.

[3] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[4] L. S. A. D. Bagdanov, A. Del Bimbo and L. Usai, "Real-time hand status recognition from rgb-d imagery," *Proceedings of the 21st International Conference on Pattern Recognition (ICPR '12)*, 2012.

[5] A. A.-H. M. Elmezain and B. Michaelis, "A robust method for hand gesture segmentation and recognition using forward spotting scheme in conditional random fields," *Proceedings of the 20th International Conference on Pattern Recognition (ICPR '10)*, 2010.

[6] S. Y. C. C.-S. Lee and S. W. Park, "Articulated hand configuration and rotation estimation using extended torus manifold embedding," *Proceedings of the 21st International Conference on Pattern Recognition (ICPR '12)*, 2012.

[7] S. S. V. G. M. R. Malgireddy, J. J. Corso and D. Mandalapu, "A framework for hand gesture recognition and spotting using sub-gesture modeling," *Proceedings of the 20th International Conference on Pattern Recognition (ICPR '10)*, 2010.

[8] a. P. Suryanarayan, A. Subramanian, "Dynamic hand pose recognition using depth data," *Proceedings of the 20th International Conference on Pattern Recognition (ICPR '10)*, 2010.

[9] J. K. S. K. S. Park, S. Yu and S. Lee, "3d hand tracking usingkalman filter in depth space," *Eurasip Journal on Advances in Signal Processing*, 2012.

[10] a. S. J. L. Raheja, A.Chaudhary, "Tracking of fingertips and centers of palm using kinect," *Proceedings of the 2nd International Conference on Computational Intelligence, Modelling and Simulation (CIMSim'11)*, 2011.

[11] X. W. S. X. Y. Wang, C. Yang and H. Li, "Kinect based dynamic hand gesture recognition algorithm research," *Proceedings of the 4th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC '12)*, 2012.

[12] M. Panwar, ""hand gesture recognition based on shape parameters," *Proceedings of the International Conference on Computing, Communication and Applications (ICCCA '12)*, 2012.

[13] K.-K. T. Z. Y. Meng, J.-S. Pan and W. Zheng, "Dominant points based hand finger counting for recognition under skin color extraction in hand gesture control system," *Proceedings of the 6th International Conference on Genetic and Evolutionary Computing (ICGEC '12)*, 2012.

[14] I. A. S. R. Harshitha and S. Srivasthava, "Hci using hand gesture recognition for digital sand model," *Proceedings of the 2nd IEEE International Conference on Image Information Processing (ICIIP '13)*, 2013.

[15] S. S. R. Yang and B. Loeding, "Handling movement epenthesis and hand segmentation ambiguities in continuous sign language recognition using nested dynamic programming," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2010.

[16] T. S. H. H. Z. Zafrulla, H. Brashear and P. Presti, "American sign language recognition with the kinect," *Proceedings of the 13th ACM International Conference on Multimodal Interfaces (ICMI '11)*, 2011.

[17] M. d. D.Uebersax, J. Gall and L. VanGool, "Realtime sign language letter and word recognition from depth the scientificworld journal 9 data," *Proceedings of the IEEE International Conference on Computer VisionWorkshops (ICCV '11)*, 2011.

[18] N. Pugeault and R. Bowden, "Spelling it out: real-time asl fingerspelling recognition," *Proceedings of the IEEE International Conference on Computer VisionWorkshops (ICCV '11)*, 2011.

[19] P. B. D. Wickeroth and U. Lang, “Markerless gesture based interaction for design review scenarios,” *Proceedings of the 2nd International Conference on the Applications of Digital Information andWeb Technologies (ICADIWT ’09)*, 2009.

[20] V. F. andD. Prattichizzo, “Usingkinect for hand tracking and rendering in wearable haptics,” *Proceedings of the IEEEWorld Haptics Conference (WHC ’11)*, 2011.

[21] H. P. J. Choi and J.-I. Park, “Hand shape recognition using distance transform and shape decomposition,” *Proceedings of the 18th IEEE International Conference on Image Processing (ICIP ’11)*, 2011.

[22] T.-D. Tan and Z.-M. Guo, “Research of hand positioning and gesture recognition based on binocular vision,” *Proceedings of the IEEE International Symposium on Virtual Reality Innovations (ISVRI ’11)*, 2011.

[23] J. S. D. Droeschel and S. Behnke, “Learning to interpret pointing gestures with a time-of-flight camera,” *Proceedings of the 6th ACM/IEEE International Conference onHuman-Robot Interaction (HRI ’11)*, 2011.

[24] S. C. K. Hu and L. Yin, “Hand pointing estimation for human computer interaction based on two orthogonal-views,” *Proceedings of the 20th International Conference on Pattern Recognition (ICPR ’10)*, 2010.

[25] N. B. K. S. J. Reza Azad, Babak Azad, “Real-time human-computer interaction based on face and hand gesture recognition,” *International Journal in Foundations of Computer Science and Technology (IJFCST)*, 2014.

[26] W. T. Freeman and M. Roth, “Orientation histograms for hand gesture recognition,” *International workshop on automatic face and gesture recognition*, 1995.

[27] T. Starner and A. Pentland, “Real-time american sign language recognition from video using hidden markov models,” *Motion-Based Recognition*, 1997.

[28] I. L. L. Bretzner and T. Lindeberg, “Hand gesture recognition using multi-scale colour features, hierarchical models and particle filtering,” *Automatic Face and Gesture Recognition, 2002. Proceedings. Fifth IEEE International Conference on. IEEE*, 2002.

[29] N. H. Dardas and N. D. Georganas, “Real-time hand gesture detection and recognition using bag-of-features and support vector machine techniques,” *IEEE Transactions on Instrumentation and Measurement*, 2011.

[30] Y. Wu and T. S, “Huangvision-based gesture recognition: A review,” *International Gesture Workshop*, 1999.

[31] S. Mitra and T. Acharya, “Gesture recognition: A survey,” *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 2007.

[32] T. Y. A. Shimada and R.-I. Taniguchi, “Hand gesture based tv control system—towards both user—and machinefriendly gesture applications,” *Proceedings of the 19th Korea- Japan JointWorkshop on Frontiers of Computer Vision (FCV '13)*, 2013.

[33] Y. E. K. C. Keskin, F. Kirac¸ and L. Akarun, “Real time hand pose estimation using depth sensors,” *Proceedings of the IEEE International Conference on Computer VisionWorkshops (ICCV '11)*, 2011.

[34] X. W. S. X. Y. Wang, C. Yang and H. Li, “Kinect based dynamic hand gesture recognition algorithm research,” *Proceedings of the 4th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC '12)*, 2012.

[35] M. Panwar, “Hand gesture recognition based on shape parameters,” *Proceedings of the International Conference on Computing, Communication and Applications (ICCCA '12)*, 2012.

[36] N. Pugeault and R. Bowden, “Spelling it out: real-time asl fingerspelling recognition,” *Proceedings of the IEEE International Conference on Computer VisionWorkshops (ICCV '11)*, 2011.

[37] S. C. K. Hu and L. Yin, “Hand pointing estimation for human computer interaction based on two orthogonal-views,” *Proceedings of the 20th International Conference on Pattern Recognition (ICPR '10)*, 2010.

[38] V. F. andD. Prattichizzo, “Usingkinect for hand tracking and rendering in wearable haptics,” *Proceedings of the IEEEWorld Haptics Conference (WHC '11)*, 2011.

[39] J. K. S. K. S. Park, S. Yu and S. Lee, “3d hand tracking usingkalman filter in depth space,” *Eurasip Journal on Advances in Signal Processing*, 2012.

[40] T.-D. Tan and Z.-M. Guo, “Research of hand positioning and gesture recognition based on binocular vision,” *Proceedings of the IEEE International Symposium on Virtual Reality Innovations (ISVRI '11)*, 2011.

[41] P. B. D. Wickeroth and U. Lang, "Markerless gesture based interaction for design review scenarios," *Proceedings of the 2nd International Conference on the Applications of Digital Information andWeb Technologies (ICADIWT '09)*, 2009.

[42] S. Y. C. C.-S. Lee and S. W. Park, "Articulated hand configuration and rotation estimation using extended torus manifold embedding," *Proceedings of the 21st International Conference on Pattern Recognition (ICPR '12)*, 2012.

[43] R. Wimmer, P. Holleis, M. Kranz, and A. Schmidt, "Thracker - using capacitive sensing for gesture recognition," in *26th IEEE International Conference on Distributed Computing Systems Workshops (ICDCSW'06)*, pp. 64–64, July 2006.

[44] B. Bongers, "Physical interfaces in the electronic arts," *Trends in Gestural Control of Music*, 2000.

[45] C. D. J. P. N. G. J. Smith, T. White and D. Allport, "Electric field sensing for graphical interfaces," *IEEE Comput. Graph. Appl.*, 1998.

[46] J. A. P. D. A. N. G. T. G. Zimmerman, J. R. Smith, "Applying electric field sensing to humancomputer interfaces.," *CHI '95: Proceedings of the SIGCHI conference on Human factors in computing systems*, 1995.

[47] C.-H. Lee and T. S. Y. Hu., "isphere: a proximity-based 3d input interface," *CAAD Futures*, 2005.

[48] E. Inc, "Ethertouch," visited January 2018.

[49] , "Theremin." `https://en.wikipedia.org/wiki/Theremin`.

[50] P. J. A. F. L. Rice, "The senses: A comprehensive reference," *Academic Press* ,, 2007.

[51] J. W. J. Tegin, "Ind. rob.," *Ind. Rob.*, 2005.

[52] J. M. A. L. A. D. H. K. H. W. Z. B. G. Schwartz, B. C.-K. Tee, "Nat. commun.," *Nat. Commun.*, 2013.

[53] M. L. Hammock, A. Chortos, B. C.-K. Tee, J. B.-H. Tok, and Z. Bao, "25th anniversary article: The evolution of electronic skin (e-skin): A brief history, design considerations, and recent progress," *Advanced Materials*, vol. 25, no. 42, pp. 5997–6038, 2013.

[54] H. Bennet, "The six million dollar man," *American Broadcasting Company*, 1974.

[55] I. Kershner, "Star wars episode v: The empire strikes back," *20th Century Fox*, 1980.

[56] J. Cameron, "The terminator," *Orion Pictures*, 1984.

[57] B. R. T. F. W. Clippinger, R. Avery, "Bull. prosthetics res.," *Bull. Prosthetics Res.*, 1974.

[58] S. W. V. J. Lumelsky, M. S. Shur, "Ieee sensors j.," *IEEE Sensors J.*, 2001.

[59] K. W. T. T. G. B. L. C. M. H. F. K. Jiang, Y. C. Tai, "Proc. ieee tenth annu. int. workshop on micro. electro. mech. syst.," *Proc. IEEE Tenth Annu. Int. Workshop on Micro. Electro. Mech. Syst.*, 1997.

[60] S. W. I.-C. Cheng, "Flexible electronics: Materials and applications," *Springer Science+Business Media*, 2009.

[61] Y. Y. L. R. W. F. Z. B. A. L. R. S. H. E. K. W. L. B. Crone, A. Dodabalapur, "Nature," *Nature*, 2000.

[62] S. W. V. J. Lumelsky, M. S. Shur, "Ieee sensors j.," *IEEE Sensors J.*, 2001.

[63] S. W. T. L. Z. G. S. S. P. Lacour, J. Jones, "Proc. ieee," *Proc. IEEE*, 2005.

[64] J. S. Y. H. J. A. R. D.-H. Kim, J. Xiao, "Adv. mater.," *Adv. Mater.*, 2010.

[65] U. Z. H. K. S. B. K. T. M. T. T. S. T. S. T. Sekitani, T. Yokota, "Science," *Science*, 2009.

[66] J. M. M. D. I. G. M. K. C. K. R. S. S. B. C. Metzger, E. Fleisch, "Appl. phys. lett.," *Appl. Phys. Lett.*, 2008.

[67] C. K. R. S. S. B. S. P. L. S. W. I. Graz, M. Kaltenbrunner, "Appl. phys. lett.," *Appl. Phys. Lett.*, 2006.

[68] R. M. S. C. V. H. H. C. S. B. B. V. O. M. A. N. S. C. R. Z. B. S. C. B. Mannsfeld, B. C.-K. Tee, "Nat. mater.," *Nat. Mater.*, 2010.

[69] R. A. Z. B. B. C.-K. Tee, C. Wang, "Nat. nanotech.," *Nat. Nanotech.*, 2012.

[70] J. X. B. H. K. S.-I. P. B. P. R. G. J. Y. M. L. Z. L. V. M. D. G. K. A.-P. L. R. G. N. D. L. K. F. G. O. Y. H. Z. K. J. A. R. R.-H. Kim, D.-H. Kim, "Nat. mater.," *Nat. Mater.*, 2010.

[71] E. D. G. T. S. T. S. N. S. S. S. B. M. Kaltenbrunner, M. S. White, "Nat. commun.," *Nat. Commun.*, 2012.

[72] V. P. Granit Luzhnica, Elizabeth Lex, "A sliding window approach to natural hand gesture recognition using a custom data glove," *3D User Interfaces (3DUI) IEEE Symposium*, 2016.

[73] T.-S. K. Ji-Hwan Kim, Nguyen Duc Thang, "3-d hand motion tracking and gesture recognition using a data glove," *Industrial Electronics IEEE Symposium*, 2009.

[74] W. H. Hung CH, Bai YW, "Home outlet and led array lamp controlled by a smartphone with a hand gesture recognition," *Consumer Electronics (ICCE); 2016 IEEE International Conference*, 2016.

[75] W. H. Hung CH, Bai YW, "Home appliance control by a hand gesture recognition belt in led array lamp case," *Consumer Electronics (GCCE); 2015 IEEE 4th Global Conference*, 2015.

[76] J. Y. G. T. H. Q. Y. B. She Y, Wang Q, "A real-time hand gesture recognition approach based on motion features of feature points," *Computational Science and Engineering (CSE); 2014 IEEE 17th International Conference*, 2014.

[77] H. K. Lee DH, "A hand gesture recognition system based on difference image entropy," *Advanced Information Management and Service (IMS), 2010 6th International Conference*, 2010.

[78] K. T. N. I. Dulayatrakul J, Prasertsakul P, "Robust implementation of hand gesture recognition for remote human-machine interaction.," *Information Technology and Electrical Engineering (ICITEE); 2015 7th International Conference*, 2015.

[79] Z. K. Tsai TH, Huang CC, "Embedded virtual mouse system by using hand gesture recognition," *Consumer Electronics-Taiwan (ICCE-TW); 2015 IEEE International Conference*, 2015.

[80] S. K. Hussain I, Talukdar AK, "Hand gesture recognition system with real-time palm tracking," *India Conference (INDICON);2014 Annual IEEE*, 2014.

[81] L. X. T. Huong TN, Huu TV, "Static hand gesture recognition for vietnamese sign language (vsl) using principle components analysis," *Communications, Management and Telecommunications (ComManTel); 2015 International Conference*, 2015.

[82] C. Y. L. G. W. X. Chen Y, Luo B, “A real-time dynamic hand gesture recognition system using kinect sensor,” *Robotics and Biomimetics (ROBIO); 2015 IEEE International Conference*, 2015.

[83] Z. L. C. Wang and S. C. Chan, “Superpixel-based hand gesture recognition with kinect depth camera,” *IEEE Transactions on Multimedia 2015*, 2015.

[84] L. C. Chen WL, Wu CH, “Depth-based hand gesture recognition using hand movements and defects,” *Next-Generation Electronics (ISNE); 2015 International Symposium*, 2015.

[85] H. C. Wong WS, Hsu SC, “Virtual touchpad: Hand gesture recognition for smartphone with depth camera,” *Consumer Electronics- Taiwan (ICCE-TW);2015 IEEE International Conference*, 2015.

[86] K. S. Ishiyama H, “Monochrome glove: A robust real-time hand gesture recognition method by using a fabric glove with design of structured markers,” *Virtual Reality (VR); 2016 IEEE*, 2016.

[87] V. V. Suriya R, “A survey on hand gesture recognition for simple mouse control,” *nformation Communication and Embedded Systems (ICICES); 2014 International Conference*, 2014.

[88] M. S. Chanda K, Ahmed W, “A new hand gesture recognition scheme for similarity measurement in a vision based barehanded approach.,” *Image Information Processing (ICIIP); 2015 Third International Conference*, 2015.

[89] L. E. P. V. Luzhnica G, Simon J, “sliding window approach to natural hand gesture recognition using a custom data glove,” *3D User Interfaces (3DUI); 2016 IEEE Symposium*, 2016.

[90] C. Y. W. X. Chen Y, Ding Z, “Rapid recognition of dynamic hand gestures using leap motion.,” *Information and Automation; 2015 IEEE International Conference*, 2015.

[91] , “Leap motion.” `https://en.wikipedia.org/wiki/Leap_Motion`.

[92] Saponas, T. Scott and Tan, Desney S and Morris, Dan and Balakrishnan, Ravin, “Demonstrating the feasibility of using forearm electromyography for muscle-computer interfaces,” *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 515–524, (2008).

[93] Guerreiro, Tiago Joao Vieira and Jorge, Joaquim Armando Pires, “Emg as a daily wearable interface,” *GRAPP*, (2006).

[94] Thalmic Labs, "Myo gesture control armband." `https://www.myo.com/`.

[95] Michael T. Wolf, Christopher Assad, Adrian Stoica,Kisung You, Henna Jethani, Matthew T. Vernacchia, Joshua Fromm, Yumi Iwashita, "Decoding static and dynamic arm and hand gestures from the jpl biosleeve," *Aerospace Conference, 2013 IEEE*, pp. 1–9, (2013).

[96] Georgi, Marcus and Amma, Christoph and Schultz, Tanja, "Recognizing hand and finger gestures with imu based motion and emg based muscle activity sensing," (2015).

[97] "Biovolt v1.0." `http://infusionsystems.com/catalog/product_info.php/products_id/198`.

[98] Christopher Allum, Shehzeen S. Hussain, Jeffrey A. Maloney, "E-μ armband," pp. 1–8.

[99] Al-Faiz, Mohammed Z and Ali, Abduladhem and Miry, Abbas H and others, "A k-nearest neighbor based algorithm for human arm movements recognition using emg signals," in *Energy, Power and Control (EPC-IQ), 2010 1st International Conference on*, pp. 159–167, IEEE, (2010).

[100] Soares, Alcimar and Andrade, Adriano and Lamounier, Edgard and Carrijo, Renato, "The development of a virtual myoelectric prosthesis controlled by an emg pattern recognition system based on neural networks," *Journal of Intelligent Information Systems*, (2003).

[101] Zhang, Xu and Chen, Xiang and Wang, Wen-hui and Yang, Ji-hai and Lantz, Vuokko and Wang, Kong-qiao, "Hand gesture recognition and virtual game control based on 3d accelerometer and emg sensors," in *Proceedings of the 14th international conference on Intelligent user interfaces*, pp. 401–406, ACM, (2009).

[102] Tu, Jack V, "Advantages and disadvantages of using artificial neural networks versus logistic regression for predicting medical outcomes," *Journal of clinical epidemiology*, pp. 1225–1231, (1996).

[103] Chan, Francis HY and Yang, Yong-Sheng and Lam, FK and Zhang, Yuan-Ting and Parker, Philip and others, "Fuzzy emg classification for prosthesis control," *Rehabilitation Engineering, IEEE Transactions on*, pp. 305–311, (2000).

[104] Ajiboye, Abidemi Bolu and others, "A heuristic fuzzy logic approach to emg pattern recognition for multifunctional prosthesis control," *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, pp. 280–291, (2005).

[105] Li-Xin Wang, "Adaptive fuzzy systems and control: design and stability analysis," pp. 29–31, (1994).

[106] Godil, Saniya Siraj and Shamim, Muhammad Shahzad and Enam, Syed Ather and Qidwai, Uvais, "Fuzzy logic: A "simple" solution for complexities in neurosciences?," *Surgical neurology international*, (2011).

[107] Mohammadreza Asghari Oskoei, Huosheng Hu, "Myoelectric control systems - a survey," *Elsevier*, pp. 276–292, (2007).

[108] Chan, Adrian DC and Englehart, Kevin B, "Continuous myoelectric control for powered prostheses using hidden markov models," *Biomedical Engineering, IEEE Transactions on*, pp. 121–124, (2005).

[109] Zhang, Xu and Chen, Xiang and Wang, Wen-hui and Yang, Ji-hai and Lantz, Vuokko and Wang, Kong-qiao, "Hand gesture recognition and virtual game control based on 3d accelerometer and emg sensors," in *Proceedings of the 14th international conference on Intelligent user interfaces*, pp. 401–406, ACM, (2009).

[110] M. L. Hammock, A. Chortos, B. C.-K. Tee, J. B.-H. Tok, and Z. Bao, "25th anniversary article: the evolution of electronic skin (e-skin): a brief history, design considerations, and recent progress," *Advanced materials*, vol. 25, no. 42, pp. 5997–6038, 2013.

[111] J. A. Rogers, T. Someya, and Y. Huang, "Materials and mechanics for stretchable electronics," *science*, vol. 327, no. 5973, pp. 1603–1607, 2010.

[112] D. J. Lipomi, M. Vosgueritchian, B. C. Tee, S. L. Hellstrom, J. A. Lee, C. H. Fox, and Z. Bao, "Skin-like pressure and strain sensors based on transparent elastic films of carbon nanotubes," *Nature nanotechnology*, vol. 6, no. 12, p. 788, 2011.

[113] H. Jinno, K. Kuribara, M. Kaltenbrunner, N. Matsuhisa, T. Someya, T. Yokota, and T. Sekitani, "Printable elastic conductors with a high conductivity for electronic textile applications," *Nature communications*, vol. 6, p. 7461, 2015.

[114] M. D. Dickey, "Emerging applications of liquid metals featuring surface oxides," *ACS applied materials & interfaces*, vol. 6, no. 21, pp. 18369–18379, 2014.

[115] I. D. Joshipura, H. R. Ayers, C. Majidi, and M. D. Dickey, "Methods to pattern liquid metals," *Journal of Materials Chemistry C*, vol. 3, no. 16, pp. 3834–3841, 2015.

[116] M. D. Dickey, R. C. Chiechi, R. J. Larsen, E. A. Weiss, D. A. Weitz, and G. M. Whitesides, “Eutectic gallium-indium (egain): a liquid metal alloy for the formation of stable structures in microchannels at room temperature,” *Advanced Functional Materials*, vol. 18, no. 7, pp. 1097–1104, 2008.

[117] A. C. Siegel, D. A. Bruzewicz, D. B. Weibel, and G. M. Whitesides, “Microsolidics: fabrication of three-dimensional metallic microstructures in poly (dimethylsiloxane),” *Advanced Materials*, vol. 19, no. 5, pp. 727–733, 2007.

[118] A. C. Siegel, D. A. Bruzewicz, D. B. Weibel, and G. M. Whitesides, “Microsolidics: fabrication of three-dimensional metallic microstructures in poly (dimethylsiloxane),” *Advanced Materials*, vol. 19, no. 5, pp. 727–733, 2007.

[119] S. Cheng and Z. Wu, “A microfluidic, reversibly stretchable, large-area wireless strain sensor,” *Advanced Functional Materials*, vol. 21, no. 12, pp. 2282–2290, 2011.

[120] R. Whitney, “The measurement of volume changes in human limbs,” *The Journal of physiology*, vol. 121, no. 1, pp. 1–27, 1953.

[121] Y.-L. Park, C. Majidi, R. Kramer, P. Bérard, and R. J. Wood, “Hyperelastic pressure sensing with a liquid-embedded elastomer,” *Journal of Micromechanics and Microengineering*, vol. 20, no. 12, p. 125029, 2010.

[122] C. Majidi, R. Kramer, and R. Wood, “A non-differential elastomer curvature sensor for softer-than-skin electronics,” *Smart Materials and Structures*, vol. 20, no. 10, p. 105017, 2011.

[123] R. D. P. Wong, J. D. Posner, and V. J. Santos, “Flexible microfluidic normal force sensor skin for tactile feedback,” *Sensors and Actuators A: Physical*, vol. 179, pp. 62–69, 2012.

[124] Y. Yu, J. Zhang, and J. Liu, “Biomedical implementation of liquid metal ink as drawable ecg electrode and skin circuit,” *PLoS One*, vol. 8, no. 3, p. e58771, 2013.

[125] P. J. Codd, A. Veaceslav, A. H. Gosline, and P. E. Dupont, “Novel pressure-sensing skin for detecting impending tissue damage during neuroendoscopy,” *Journal of Neurosurgery: Pediatrics*, vol. 13, no. 1, pp. 114–121, 2014.

[126] F. L. Hammond, R. K. Kramer, Q. Wan, R. D. Howe, and R. J. Wood, “Soft tactile sensor arrays for micromanipulation,” in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pp. 25–32, IEEE, 2012.

[127] V. Arabagi, O. Felfoul, A. H. Gosline, R. J. Wood, and P. E. Dupont, "Biocompatible pressure sensing skins for minimally invasive surgical instruments," *IEEE sensors journal*, vol. 16, no. 5, pp. 1294–1303, 2016.

[128] S. Yao and Y. Zhu, "Wearable multifunctional sensors using printed stretchable conductors made of silver nanowires," *Nanoscale*, vol. 6, no. 4, pp. 2345–2352, 2014.

[129] Q. Zhang, Y. Gao, and J. Liu, "Atomized spraying of liquid metal droplets on desired substrate surfaces as a generalized way for ubiquitous printed electronics," *Applied Physics A*, vol. 116, no. 3, pp. 1091–1097, 2014.

[130] T. Lu, J. Wissman, and C. Majidi, "Soft anisotropic conductors as electric vias for ga-based liquid metal circuits," *ACS applied materials & interfaces*, vol. 7, no. 48, pp. 26923–26929, 2015.

[131] John C. Platt, "Sequential minimal optimization: A fast algorithm for training support vector machines," pp. 1–21, (1998).

[132] Andrew Ng, "Support vector machine." `http://cs229.stanford.edu/notes/cs229-notes3.pdf`.

[133] C. benussi, "Emg control of a soft prosthetic hand."

[134] W. S. Sarle., "Part 2 of neural network faq," *Periodic posting to the Usenet newsgroup comp.ai.neural-nets*, 1997. `ftp://ftp.sas.com/pub/neural/FAQ.html`.

Ringraziamenti

Dopo 2 anni e sette lunghi e intensi mesi, finalmente il giorno è arrivato: scrivere queste frasi di ringraziamento è il tocco finale del mio percorso di studi. È stato un periodo di profondo apprendimento, non solo a livello scientifico, ma soprattutto a livello personale, un percorso che mi ha fatto crescere e maturare anche se a molti di voi non sembrerà così. Scrivere questa tesi e l'esperienza che ho affrontato in questi 7 mesi hanno avuto un forte impatto sulla mia personalità. Vorrei spendere due parole di ringraziamento nei confronti di tutte le persone che mi hanno sostenuto, aiutato o anche solo quelle che hanno deciso di percorrere e condividere un pezzetto della loro vita con me durante questo lungo periodo.

Prima di tutto vorrei ringraziare i miei relatori, i professori M. Chiaberge e M. Tavakoli per avermi dato questa preziosa possibilità, per i consigli e per avermi fornito tutti gli strumenti di cui avevo bisogno per intraprendere la strada giusta e portare a compimento la mia tesi.

Un ringraziamento particolare va a tutta la mia famiglia, iniziando da mamma e papà che mi hanno sempre sopportato, sostenuto nelle mie decisioni e soprattutto economicamente facendo molti sacrifici, grazie a Marta, mia sorella, che alla fine sono 23 anni che mi sopporta; grazie ai miei nonni, in fondo i geni partono da lì, grazie a tutti gli zii e i cugini, perché anche loro mi hanno sempre appoggiato anche se molte volte avranno pensato e penseranno che sono pazzo.

Per ultimi ma assolutamente non meno importanti i miei amici, qualcuno mi ha tanto preso in giro perché ho sempre sostenuto di averne 3 o 4, beh ora che sono qua a dover pensare a tutti mi rendo conto che non è così e ne sono felice. Vorrei non dover decidere un ordine in cui elencarvi e proverò a farlo in ordine cronologico, partendo dagli amici di sempre Simo, Giamma e Fra ci conosciamo da una vita e anche se abbiamo preso strade diverse e abbiamo obiettivi differenti vi manderò sempre mille messaggi per raccontarvi tutte le cose divertenti che mi capiteranno e a Torino troverò sempre il tempo per una serata a ridere sul niente o su di me!

Maggi e Ico un'amicizia iniziata per caso, che spero di preservare anche se ci troviamo sempre in posti diversi del mondo, i migliori compagni di avventura!

Ruben e Alberto, amici del mio Erasmus in casa, i quali mi hanno fatto capire che sarei dovuto partire per farne uno vero perché è un'esperienza da vivere, i migliori compagni di serata.

I compagni di pallavolo e beach volley, che dopo ogni tipo di giornata, dalla più triste alla più bella, loro erano pronti a farti sudare e divertire facendoti dimenticare di tutto quello che era successo, i miei compagni di università che hanno reso le lezioni più interessanti per quello che ricordo quando non dormivo, gli amici del mare che mi hanno accolto nella loro compagnia con i quali ho condiviso molte giornate e nottate in spiaggia.

At the end, you, those friends that you would never have thought could be your friends, those that you seem to know for a lifetime and instead you know them just for few months, intense months, months of Erasmus! Collegues, travellers mates, laughs, evenings, nights, feelings, emotions, adventures, a period of my life that has changed me, for better or worse I don't know, but I am different and it is your merit/fault: Irene, Marco, Alex, Marc, Andrea, Leo, Malika, Ste, Mimmo, Rafa, Roman, Ale, Jose, Cate.

Grazie, Thanks, Obrigado, Gracias.