

**POLITECNICO DI MILANO**

Dipartimento di Elettronica, Informazione e Bioingegneria

Tesi di Laurea Magistrale in Ingegneria Biomedica



# Virtual reality environment with low-cost haptic feedback thimble for post spinal cord injury upper-limb rehabilitation

Relatore: Prof. Alberto Redaelli

*Politecnico di Milano*

Correlatore: Prof. Álvaro Gutiérrez Martín

*Universidad Politécnica de Madrid*

Autore: Nicola Farella

Matricola: 896476

Anno Accademico 2018 – 2019



# Ringraziamenti

Termina oggi un percorso lungo e ricco di entusiasmanti esperienze. Mi sembra dunque doveroso dedicare qualche riga alle persone importanti che mi hanno sostenuto e accompagnato in questo percorso.

In primis, un ringraziamento al mio relatore Alberto Redaelli per la sua disponibilità e per avermi concesso la libertà e la possibilità di soddisfare la mia curiosità anche fuori dall'Italia. Un ringraziamento speciale al correlatore Álvaro Gutiérrez Martín per i suoi indispensabili consigli, per le conoscenze trasmesse durante tutto il percorso di tesi e per aver scommesso su di me, accogliendomi nel suo laboratorio. Un grazie a tutti i membri del Robolabo, in particolare a Blanca e Dani per avermi ascoltato e supportato, ma soprattutto per le risate e per avermi fatto sentire a casa. Muchísimas gracias a todos!

Il ringraziamento più grande va ai miei genitori, continua fonte di ispirazione e primi sostenitori di tutte le mie scelte. Voi siete il modello da seguire sempre. Mi avete educato ad essere cittadino del mondo e anche se sono spesso lontano vi sarò sempre vicino. A mio fratello Filippo, capace di farmi sorridere in ogni occasione e disposto costantemente ad ascoltare e risolvere i miei problemi. Grazie a tutti i membri della mia numerosissima famiglia. Tutti gli zii, zie, nonne e cugini sono per me punti di riferimento imprescindibili.

Gli anni del Liceo sono belli sì, ma sono gli amici che ti circondano che li rendono speciali e unici. In quegli anni ho trovato la mia seconda famiglia. Un ringraziamento di cuore va di dovere a Nico, l'amico di sempre, a Rocco, la persona più gentile che ci sia, ad Ale, il dottore che ci curerà sempre, ad Antonio, il nostro avvocato speciale, ad Ivano, l'uomo dalle mille perle e ad Edoardo, amico di mille avventure. Grazie a tutti voi per esserci sempre e supportarmi ancora a distanza di anni, anche quando non leggo i messaggi sul gruppo. E ovviamente grazie alle Labbles Anna, Titta, Isa, Mariacristina e Dona, le amiche di sempre con cui scherzare e che in un modo o nell'altro sono le ragazze, ormai donne, che reputo come sorelle.

Nella mia vita sono arrivati poi Milano e il Politecnico. Entrambi belli e difficili, alle volte quasi insopportabili. Qui ho trovato delle persone uniche e inimitabili. Con loro ho condiviso ansie, paure, felicità ed emozioni forti. In primis, devo ringraziare Andrea e Marco, per me insostituibili amici di viaggi interminabili e di serate piene di risate e nonsense. Grazie poi a Fabi, Fede michetta, Bianca e Virgi e a tutti i miei compagni in questa esperienza al Poli.

Al di fuori dell'università c'era sempre qualcuno ad aspettarmi per il weekend e per le serate più pazzes. Grazie Davide, Giovanni, Peppe, Betta, il Pilla e l'Ingegnere. E grazie via Vetere, casa di mille follie, sarai sempre un faro nella notte milanese.

Milano non sarebbe stata la stessa senza l'umile dimora di via Strambio. Grazie alle mie coinquiline speciali Silvia, Fra, Mariachiara e Roberta che si sono susseguite e che sono riuscite ad apprezzare e convivere con un pazzo come me.

Gli ultimi mesi sono stati per me rinascita e riscoperta in un posto ricco di vita: mi Madrid hermosa. Mi ha rubato il cuore e l'hai riempito di allegria, amore e di persone fantastiche. Grazie in primis a Bea e Ali per avermi regalato complicità e risate fino all'ultimo giorno, condividendo con me questa esperienza a 360 gradi. Grazie a calle Flor Baja per avermi regalato la familia madrileña. Un ringraziamento a Tommaso, Raqui, Carlotta, Ilaria, Luci e Fredo per aver riempito le mie giornate di alcool, risa e affetto vero e sincero. Os quiero mucho.

Madrid però si è superata e mi ha regalato una persona incredibilmente inaspettata e inaspettatamente incredibile. Grazie Vicky per essermi stata accanto tutti i giorni nei momenti belli e sereni, ma anche in quelli più brutti e angoscianti. A te che hai saputo sostenermi e tranquillizzarmi anche quando non riuscivo a dormire.

Grazie a tutti di cuore!

# CONTENTS

CONTENTS .....	v
List of Figures.....	viii
List of Tables .....	xii
SOMMARIO .....	xiv
ABSTRACT .....	xvi
Chapter 1: Introduction .....	1
1.1 Clinical Scenario .....	1
1.1.1 Spinal Cord Injury .....	1
1.1.2 Health, Economic and Social Impact .....	6
1.2 Current Clinical Solution .....	8
1.2.1 Traditional Rehabilitation.....	8
1.2.2 Robotic and Virtual Reality Rehabilitation .....	14
1.2.3 Haptic Feedback .....	19
1.3 State of the Art.....	22
1.3.1 Kinaesthetic Feedback Devices .....	23
1.3.2 Tactile Feedback Devices .....	27
1.4 Current Thesis: Motivation and Objectives .....	32
Chapter 2: Haptic Feedback Device .....	35
2.1 Mechanics .....	35
2.1.1 Design Evolution of The Elements.....	38
2.1.2 Motor Description .....	48
2.1.3 Mechanical Connections and Final Assembling .....	50
2.2 Electronics .....	53
2.2.1 Boards and Connections .....	53

2.3	Firmware .....	59
2.4	Evaluation of the Device.....	66
Chapter 3:	Virtual Platform.....	69
3.1	Enabling Technologies .....	69
3.1.1	Unity .....	69
3.1.2	Leap Motion Controller .....	71
3.1.3	Elements Integration.....	72
3.2	Games .....	74
3.2.1	Introduction and Patient Management.....	77
3.2.2	Common Functions .....	80
3.2.3	Maze .....	81
3.2.4	Path .....	87
3.2.5	Grab .....	92
3.2.6	Calibration Scene.....	97
Chapter 4:	Results .....	99
4.1	Final platform prototype .....	99
4.2	Clinical Validation as a Proof of Concept .....	101
4.2.1	Maze Results .....	103
4.2.2	Path Results .....	106
4.2.3	Grab Results .....	109
4.2.4	System Usability Questionnaire .....	112
Chapter 5:	Conclusions and Future Developments .....	117
5.1	Conclusions.....	117
5.2	Future Developments .....	119
BIBLIOGRAPHY	.....	122



# List of Figures

Figure 1.1 Spinal nerves classification and the body regions controlled (Singh A. , 2019).	2
Figure 1.2 Levels of injury with the associated paralysis (MichelsonMedical, 2014).	3
Figure 1.3 Robotic devices for upper limb rehabilitation: a) exoskeletons, b) end effector. (Bertomeu-Motos, et al., 2018).	15
Figure 1.4 Left image: Exoskeleton-Armeo spring (Hocoma, 2019) . Right image: InMotion Arm (Macovei & Doroftei, 2016).	16
Figure 1.5 Mechanoreceptors (Yem & Kajimoto, 2017).	20
Figure 1.6 Basic architecture for a virtual reality application incorporating visual, auditory, and haptic feedback. (Salisbury, Conti, & Barbagli, 2004).	20
Figure 1.7 Kinaesthetic devices: exoskeletons. A) CyberGrasp (CyberGloveSystems, s.d.). B) Rutgers Master II- ND (Bouzit, G., Popescu, & Boian, 2002). C) (Secciani, et al., 2019). D) PneuGlove (Connelly, et al., 2010). E) Soft Robotic Glove (Cappello, et al., 2018).	24
Figure 1.8 Kinaesthetic devices: End-Effector. Geomagic PHANToM on the left and Novint Falcon on the right. (Piggott, Wagner, & Ziat, 2016)	26
Figure 1.9 CyberTouch (Piggott, Wagner, & Ziat, 2016) and integration ith virtual reality (Dimbwadyo-Terrer, et al., 2016).	28
Figure 1.10 Thimble devices. A) Vibration motor thimble. B) Motor-driven “squeezer”. C) Shape memory alloy thimble. (Merrett, et al., 2011).	29
Figure 1.11 Tactile haptic devices. A) (Pacchierotti, Tirmizi, & Prattichizzo, 2014). B) FinGAR (Yem & Kajimoto, 2017). C) (Chinello, Malvezzi, Pacchierotti, & Prattichizzo, 2015). D) Artificial skin. (Pessina, 2019).	31
Figure 2.1 Schematization of the elements that compose the device.	37
Figure 2.2 AnetA8 (Shenzhen Anet Technology Co., Ltd.).	38
Figure 2.3 The first prototype of the base element. The side supports for the Velcro strips (red circles), the three holes (black circles), the frontal relief (yellow circle) and the two parallel walls (blue circle) to block the motor position are highlighted.	39
Figure 2.4 The second prototype of the base. The small parts in the lower corners for blocking the motor (black circles), the structure for the cables (yellow rectangles) and the tube-like structure for cables and compression strings (red circles) are highlighted.	40



Figure 2.5 The cables routing problem in the first prototype (left) and the solution on the second one (right). .....	41
Figure 2.6 The third prototype of the base. The structure that covers the motor in the upper part is visible, avoiding the inclination toward the front part (red rectangle). .....	43
Figure 2.7 Front view of the base element. ....	44
Figure 2.8 Top view of the base element. ....	44
Figure 2.9 The final version of the base. Real piece (left) and CAD model. ....	44
Figure 2.10 The first prototype of the platform. The vertical column guide is visible. ....	45
Figure 2.11 The final prototype of the platform. Real object and CAD model (left) are shown together with the measures of the element (right). The side arms (red circles) and the front arm (blue circle) are highlighted. ....	46
Figure 2.12 Final version of the reel. CAD model and real piece are shown.....	47
Figure 2.13 Dimensions of the reel. ....	48
Figure 2.14 The D-shaped connector attached to the metal gearbox (Pololu, 2019). ....	49
Figure 2.15 298:1 Micro Metal Gearmotor HPCB 12V with Extended Motor Shaft (Pololu, 2019).....	49
Figure 2.16 Magnetic Encoder mounted on the rear motor shaft (Pololu, 2019).....	49
Figure 2.17 Exploded model of the thimble. ....	50
Figure 2.18 Cable fixing. First the cables are inserted into the small holes in the third element (left), then two double copper ferrules are placed on the tips and tightened (centre). The cables are pulled into the holes and fixed with solder tin (right). ....	51
Figure 2.19 Cable configuration system. This solution allows to move the cables in the same direction. ....	52
Figure 2.20 The solution adopted to stop the cables outside the platform. ....	52
Figure 2.21 Final model of the device. ....	52
Figure 2.22 Arduino Due Board (Arduino, 2019). ....	54
Figure 2.23 Power Stage, X-NUCLEO-IHM04A1, dual brush DC motor drivers based on L6206 expansion for STM32 Core (left) and its schematic representation in the configuration that allows connections with two motors and with the power supply system (St, 2019)....	55
Figure 2.24 First welding configuration (left) and last stronger configuration (right).....	57
Figure 2.25 Final outcome of electronics and connections. ....	58
Figure 2.26 Electronic components and connections. ....	58

Figure 2.27 Essential functioning of the code. ....	60
Figure 2.28 Activation of the interrupts. ....	61
Figure 2.29 Construction of serial data. ....	62
Figure 2.30 Options for values>255. After stopping the timers (upper figure), vibration(left), releasing cable (center) or tightening cable (right) options are eligible. ....	63
Figure 2.31 PD control activation in <i>loop()</i> function. ....	63
Figure 2.32 PD control function. ....	64
Figure 2.33 Block diagram of the PD controller. ....	65
Figure 2.34 Distance between the base and the parallel platform in correspondence of the value sent. ....	67
Figure 3.1 Leap Motion controller (LeapMotion, 2019). ....	71
Figure 3.2 The Unity left-handed coordinate system superimposed on the Leap Motion device in its desktop orientation (LeapMotion, 2019). ....	73
Figure 3.3 Representation of the device and the working space in the Unity Scene View. ....	74
Figure 3.4 Path through the initial windows. ....	78
Figure 3.5 Essential information for the new patient. ....	79
Figure 3.6 The function used for the communication with the microcontroller. ....	81
Figure 3.7 Game window of Maze. Time, errors and points are continuously update. The red circle highlights the head of the squirrel controlled by the user. ....	81
Figure 3.8 State transition networks (STN) for the Maze game. ....	82
Figure 3.9 Settings window of Maze. ....	83
Figure 3.10 Results window of Maze. ....	86
Figure 3.11 Game window of Path. The white dot is on the line in the upper part. The travelled lines are obscured. ....	87
Figure 3.12 State transition networks (STN) for the Path game. ....	88
Figure 3.13 Setting window of Path. ....	89
Figure 3.14 Force feedback applied in Path depending on the distance of the white dot from the path. On the left the trajectory of the white dot from the shape moves away from the path of a distance “d”. On the right, the thimble, consisting of a base and a parallel platform (gray), tightens the finger (pink) by applying a force proportional to the distance. ....	90
Figure 3.15 Results window of Path. ....	91
Figure 3.16 Game window of Grab. ....	92

Figure 3.17 State transition networks (STN) for the Grab game. ....	93
Figure 3.18 Setting window of Grab. ....	94
Figure 3.19 Haptic feedback provided for the same object. On the right the distance between fingers is shown. On the left, the device applies a force depending on the distance and on the k-value of the object (fixed in this case). $\vartheta$ represents the grabbing distance, i.e. the threshold distance that the user needs to pass to grab an object. The smaller is the distance between fingers, the greater the force applied. ....	95
Figure 3.20 Results window of Grab. ....	96
Figure 3.21 Calibration scene. ....	97
Figure 4.1 Complete final platform. ....	100
Figure 4.2 Graphs of data relative to Maze game. A healthy subject (left) and a patient (right) are compared. ....	104
Figure 4.3 Graphs of data relative to Path game. A healthy subject (left) and a patient (right) are compared. ....	107
Figure 4.4 Graphs of data relative to Grab game. A healthy subject (left) and a patient (right) are compared. Examples of object are shown in correspondence of their k-value. ....	110

# List of Tables

Table 1.1 ASIA Impairment Scale (Kirshblum, et al., 2011 ).	5
Table 1.2 Cervical nerve function (Slosar, 2019).	5
Table 1.3 Estimation of the expenses (U.S.A) (NationalSpinalCordInjuryStatisticalCenter, 2019).	7
Table 2.1 Connection between the power stage and the Arduino Due.	56
Table 2.2 Connection between the motor and the electronic system.	56
Table 2.3 Arithmetic mean and standard deviation of the distance between the base and the parallel platform in correspondence of the value sent.	66
Table 3.1 Main characteristics of each game.	77
Table 4.1 SCI patients characteristics.	101
Table 4.2 Results of Maze game. The values represent the arithmetic mean of the data in both groups: healthy subject and SCI patients.	105
Table 4.3 Errors in the Maze game. The average of the error is lower when the feedback is enables	106
Table 4.4 Results of Path game. The values represent the arithmetic mean of the data in both groups: healthy subject and SCI patients.	108
Table 4.5 Errors in the Path game. The average of the error is lower when the feedback is enables for the patients. The errors are almost the same in both cases for the healthy subjects.	109
Table 4.6 Grabbable objects with the corresponding k-values.	111
Table 4.7 Results of Grab game. The values represent the arithmetic mean of the data in both groups: healthy subject and SCI patients.	112
Table 4.8 System Usability Questionnaire for healthy subjects.	113
Table 4.9 System Usability Questionnaire for patients.	114
Table 4.10 Level of perceived effort in a scale from 0 to 10.	115



# SOMMARIO

Lo scopo di questa tesi è quello di progettare e implementare una piattaforma di realtà virtuale per la riabilitazione degli arti superiori di pazienti con lesioni cervicali del midollo spinale.

La lesione spinale o mielolesione è un danno meccanico al midollo spinale. Si tratta di una malattia neurologica che provoca cambiamenti permanenti nella forza, nella sensazione e in altre funzioni corporee al di sotto della sede della lesione. Una delle sue possibili conseguenze è la tetraplegia, che è il deterioramento o la perdita della funzione motoria e/o sensoriale di braccia, tronco, gambe e organi pelvici. In particolare, il danneggiamento dell'arto superiore limita l'autonomia del paziente nell'esecuzione delle attività quotidiane. Pertanto, la mielolesione ha un enorme impatto socioeconomico sulle persone colpite e sul sistema sanitario.

Al giorno d'oggi, i robot e la realtà virtuale sono sempre più spesso impiegati nella riabilitazione grazie alla loro efficacia. Queste tecnologie possono essere utilizzate insieme alla riabilitazione tradizionale per completarla e migliorare il processo di recupero. Combinando questi due tipi di riabilitazione, i pazienti possono aumentare la durata e il numero di sedute, accelerando così il recupero. Inoltre, i dispositivi robotici possono essere utilizzati per fornire un feedback tattile, permettendo così agli utenti di "sentire" l'interazione con l'ambiente virtuale. Il feedback tattile è fondamentale per migliorare il recupero somatosensoriale strettamente legato al recupero motorio.

Sono state create diverse soluzioni per fornire un feedback tattile in modi diversi, stimolando così parti distinte del sistema somatosensoriale.

Lo scopo principale di questa tesi è quello di progettare e realizzare un prototipo di piattaforma di realtà virtuale per la riabilitazione degli arti superiori composta da un insieme di giochi integrati con un sensore di rilevamento della mano e un dispositivo di feedback tattile. Durante lo sviluppo di questa tesi, l'autore ha progettato, elaborato e implementato gli elementi meccanici che compongono il dispositivo di feedback aptico, un ditale. Inoltre, ha configurato l'hardware da collegare al motore integrato con le parti meccaniche, implementato il firmware di controllo del dispositivo e programmato i giochi di realtà virtuale, compreso il sensore di tracciamento delle mani. Ha sviluppato tre giochi VR da integrare con il ditale.

I tre giochi in realtà virtuale sono stati sviluppati in Unity. Ogni gioco ha diversi obiettivi di riabilitazione degli arti superiori, ma tutti cercano di migliorare la coordinazione e la forza muscolare di braccio e mano. La realtà virtuale coinvolge gli utenti e permette di adattare le caratteristiche dei giochi alle capacità dei diversi pazienti. L'utente può giocare direttamente con l'uso delle mani incluse nell'ambiente virtuale grazie al sensore. Inoltre, l'utente può "sentire" l'interazione con gli oggetti virtuali grazie al dito aptico stampato in 3D in grado di fornire diversi tipi di feedback. Questo obiettivo viene raggiunto grazie al firmware che permette la comunicazione tra il computer e la scheda Arduino che controlla il dito.

Al termine della tesi è stata eseguita una prova di fattibilità sia con soggetti sani che con pazienti SCI. Questa prova di fattibilità permette di valutare la piattaforma e la sua usabilità con i pazienti. Inoltre, i suoi promettenti risultati hanno permesso di pianificare una sperimentazione clinica che si svolgerà nel 2020 presso il National Hospital of Paraplegics a Toledo (Spagna).

# ABSTRACT

The aim of this Thesis is to design and implement a virtual reality platform for upper limb rehabilitation of cervical spinal cord injury patients.

Spinal cord injury (SCI) is a mechanical damage to the spinal cord. It is a neurological disease that causes permanent changes in strength, sensation, and other body functions below the site of the injury. One of its possible consequences is tetraplegia, which is the deterioration or loss of motor and/or sensory function of the arms, trunk, legs and pelvic organs. In particular, the impairment of the upper limb limits the patient's autonomy in the execution of activities of daily living. Therefore, SCI has a tremendous socioeconomic impact on affected individuals and the health care system.

Nowadays, robots and virtual reality are more often employed in rehabilitation due to their effectiveness. These technologies can be used along with traditional rehabilitation to complement it and enhance the recovery process. By combining these two types of rehabilitation, patients can increase the duration and number of sessions, thus accelerating the recovery. Moreover, the robotic devices can be used to provide haptic feedback, thus allowing users to “feel” the interaction with the virtual environment. The haptic feedback is fundamental to improve the somatosensory recovery that is strictly related to motor recovery. Different solutions have been created to provide haptic feedback in different ways, thus stimulating distinct parts of the somatosensory system.

The main purpose of this Thesis is to design and implement a prototype of virtual reality platform for upper limb rehabilitation composed of a set of games integrated with a tracking hand sensor and a haptic feedback device. During this Thesis, the author has designed, developed and implemented the mechanical elements composing the haptic feedback device, a thimble. Moreover, he has configured the hardware to be connected to the motor integrated with the mechanical parts, implemented the firmware controlling the device and programmed the virtual reality games, including the hand tracking sensor. He has developed three VR games to be integrated with the thimble.

The three VR games have been developed in Unity. Each game has different upper limb rehabilitation objectives, but all of them attempt to improve the coordination and muscle strength of arm and hand. Virtual reality involves the users and allows to adapt the characteristics of the games to the capabilities of different patients. The user can play the



games directly with the use of hands included in the virtual environment thanks to the sensor. Moreover, the user can “feel” the interaction with the virtual objects thanks to the custom-made 3D printed haptic thimble capable of providing different types of feedback. This objective is reached thanks to the firmware that allows the communication between the computer and the Arduino board that controls the thimble.

At the end of the Thesis, a proof of concept has been performed both with healthy subjects and SCI patients. This proof of concept allows to evaluate the platform and its usability with patients. Moreover, its promising results have allowed planning a clinical trial that will take place in 2020 at National Hospital of Paraplegics in Toledo (Spain).



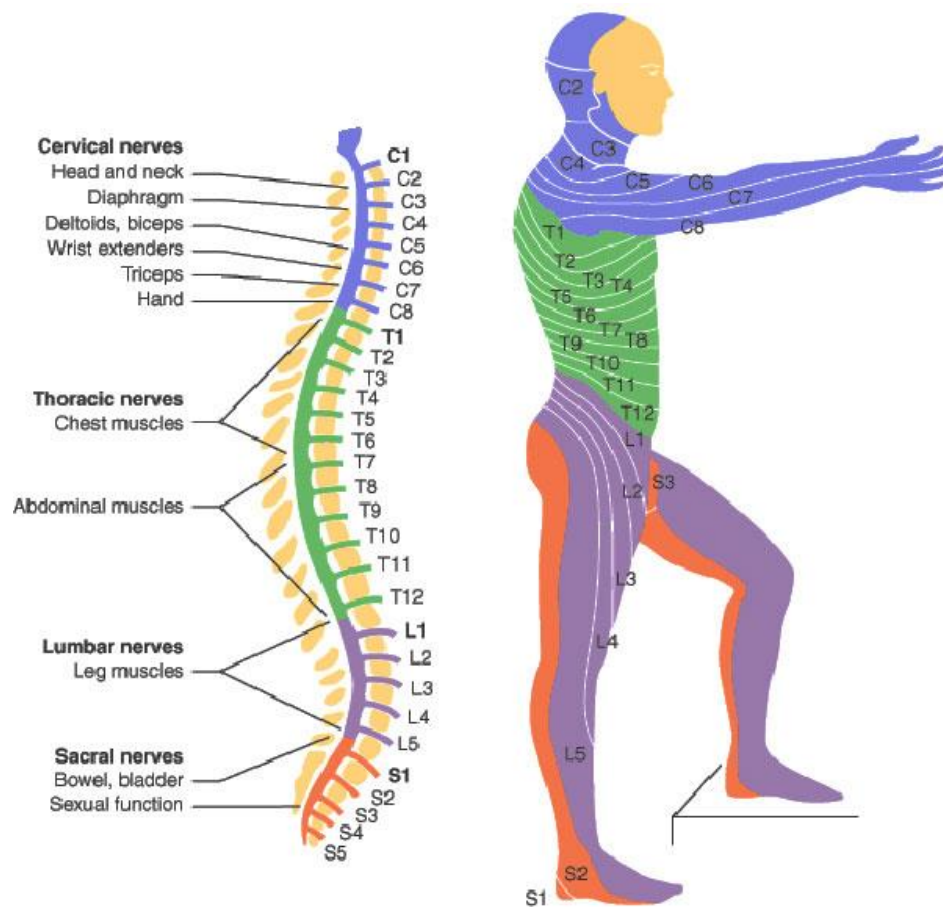
# **Chapter 1: Introduction**

## **1.1 Clinical Scenario**

### **1.1.1 Spinal Cord Injury**

The spinal cord is a fundamental component of the central nervous system (CNS) together with the brain (Sheerin, 2004). It is a long, fragile, cylindrical nerve structure that extends from the end of the brain stem at the top of the neck as a direct continuation of the medulla oblongata and continues down to the lumbar region of the vertebral column (Choi, et al., 2019). On cross section, the spinal cord shows a butterfly-shaped center of gray matter that consist of nerve cells, axons and dendrites, surrounded by a white matter structure of longitudinally running myelinated and unmyelinated axonal tracts. The ratio of gray/white matter vary between spinal cord regions together with the shape: at cervical levels the cord has an oval shape with the transverse diameter bigger than the antero-posterior (AP) diameter, while in the thoracic and lumbar levels, it appears round (Diaz & Morales, 2016).

From the gray matter two nerve roots originate, being fundamental for the transmission function of the spinal cord. The posterior nerve root (dorsal) is composed of cell bodies in posterior spinal root ganglion with afferent sensory fibers from the peripheral nervous system. These neurones carry information about pain, touch, vibration, temperature and proprioception from the body wall, tendons and joints, as well as sensory impulses from organs within the body. The anterior root (ventral) is composed of motor fibers (efferent) leaving the bodies of nerve cells in the anterior horn with the function of transmitting impulses to voluntary striated muscles as well as smooth and cardiac muscles. In addition to the main function of transmitting nerve signals from (efferent) and to (afferent) the brain, the spinal cord serves as the reflex centre for the body integrating and controlling both somatic and autonomic reflexes (Sheerin, 2004). The dorsal and ventral nerve roots come together immediately proximal to the spinal cords to form a spinal nerve. In humans there are 31 pairs of spinal nerves that control



**Figure 1.1 Spinal nerves classification and the body regions controlled (Singh A. , 2019).**

different regions ensuring different functions. The pairs are named depending on the region of the spinal cord in which are located: 8 cervical (C1-C8) located in the neck; 12 thoracic (T1-T12) in the chest; 5 lumbar (L1-L5) located in the abdomen; 5 sacral (S1-S5) located in the pelvis and 1 coccygeal pair placed in the tailbone (Choi, et al., 2019) (see Figure 1.1).

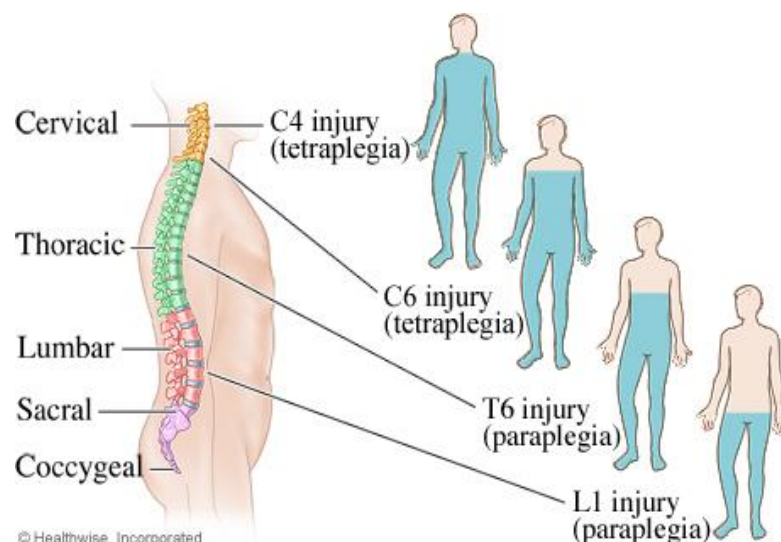
The Spinal Cord Injury (SCI) is a mechanical damage to the spinal cord. It can be traumatic when the damage results from physical traumas (car accidents, work accidents, gunshots, falls, sports injuries, etc.) or non-traumatic when it is caused by an underlying pathology (infection, tumors, musculoskeletal disease, etc.)

(Bickenbach, et al., 2013). In the first case, injury begins with a traumatic blow to the spine: displaced bone fragment, disc material, or ligament bruise or tear into spinal cord tissue. This causes an alteration of its function since there is an interruption of nervous tissue that perform the communication between the brain and the rest of the body (NINDS, 2019). Consequently, the damage results in temporary or permanent loss of movement (motor function), sensation (sensory function), and bodily systems (autonomic functions) below the site of the injury.

Owing to the difference in length between the spinal column and the spinal cord, the neurological levels do not necessarily correspond to the vertebral segments.

According to the level of the injury, there are different types of paralysis (see Figure 1.2) that may be grouped as (Kirshblum, et al., 2011 ):

- Tetraplegia: deterioration or loss of motor and/or sensory function in the cervical segments of the spinal cord that results in impairment of function in the arms as well as in the trunk, legs and pelvic organs.
- Paraplegia: deterioration or loss of motor and/or sensory function in the thoracic, lumbar or sacral (but not cervical) segments of the spinal cord resulting in impairment of function of the trunk, legs and pelvic organs (depending on the level of injury).



**Figure 1.2 Levels of injury with the associated paralysis**  
(MichelsonMedical, 2014).

Moreover, SCI can be classified according to the severity. Basically, an *incomplete injury* occurs when the damage does not completely disrupt the ability of the spinal cord to convey messages to or from the brain. People with incomplete injuries retain some sensory function and may have voluntary motor activity below the injury site. On the contrary a *complete injury* occurs when the damage impedes completely the nerve communications. Consequently, there is a total lack of sensory and motor function below the level of injury. Therefore, determining the cord segments affected by the spinal cord injury appears to be crucial in order to understand the gravity of the situation. This is done by systematically examining the dermatomes (area of the skin innervated by the sensory axons) and myotomes (muscle fibers innervated by the motor axons) (Kirshblum, et al., 2011 ). With the aim to elaborate a standard, the American Spinal Injury Association (ASIA) developed The International Standards for Neurological Classification of Spinal Cord Injury (ISNCSCI), commonly referred to as the ASIA Exam. It is a universal classification tool for Spinal Cord Injury based on both a motor and sensory examination to determine the Sensory Level and Motor Level for each side of the body (Right and Left), the single Neurological Level of Injury (NLI) and completeness of the Injury. The classification is shown in the AIS (ASIA Impairment Scale) (see Table 1).

GRADE	DEFINITION
<b>A</b>	<b>Complete.</b> No sensory or motor function is preserved in the sacral segments S4-S5.
<b>B</b>	<b>Sensory incomplete.</b> Sensory but not motor function is preserved below the neurological level and includes the sacral segments S4-S5, AND no motor function is preserved more than three levels below the motor level on either side of the body.
<b>C</b>	<b>Motor incomplete.</b> Motor function is preserved below the neurological level**, and more than half of key muscle functions below the single neurological level of injury have a muscle grade less than 3 (Grades 0–2).
<b>D</b>	<b>Motor incomplete.</b> Motor function is preserved below the neurological level**, and at least half (half or more) of key muscle functions below the NLI have a muscle grade >3.

<b>E</b>	<b>Normal.</b> If sensation and motor function as tested with the ISNCSCI are graded as normal in all segments, and the patient had prior deficits, then the AIS grade is E. Someone without a SCI does not receive an AIS grade.
----------	---

**Table 1.1 ASIA Impairment Scale (Kirshblum, et al., 2011 ).**

As already said, spinal cord injuries at the cervical level result in full or partial tetraplegia. Since this work is focused on the upper limb part, it is important to know the functions provided by each cervical spinal nerve in order to understand the particular ability that may be retained, depending on the specific location and severity of trauma. Table 1.2 illustrates both the motor and sensory function of the cervical nerves indicating the myotomes and dermatomes (Slosar, 2019).

<b>CERVICAL NERVES</b>	<b>FUNCTIONS</b>
<b>C1, C2, C3</b>	Help control the head and neck, including movements forward, backward, and to the sides. The C2 dermatome handles sensation for the upper part of the head, and the C3 dermatome covers the side of the face and back of the head (C1 does not have a dermatome).
<b>C4</b>	Helps control upward shoulder movements. C4 (along with C3 and C5) also helps power the diaphragm (the sheet of muscle that stretches to the bottom of the rib cage for breathing). The C4 dermatome covers parts of the neck, shoulders, and upper part of arms.
<b>C5</b>	Helps control the deltoids (which form the rounded contours of the shoulders) and the biceps (which allow bending of the elbow and rotation of the forearm). The C5 dermatome covers the outer part of the upper arm down to about the elbow.
<b>C6</b>	Helps control the wrist extensors (muscles that control wrist extension) and also provides some innervation to the biceps. The C6 dermatome covers the thumb side of the hand and forearm.
<b>C7</b>	Helps control the triceps (the large muscle on the back of the arm that straightens the elbow) and wrist extensor muscles. The C7 dermatome goes down the back of the arm and into the middle finger.
<b>C8, (T1)</b>	Helps control the hands, in particular the finger flexion (handgrip). The finger abduction is controlled by the T1 thoracic spinal nerve. The C8 dermatome covers the pinky side of the hand and forearm.

**Table 1.2 Cervical nerve function (Slosar, 2019).**

### **1.1.2 Health, Economic and Social Impact**

Spinal cord injury is a medically complex and life-disrupting neurological condition with tremendous socioeconomic impact on affected individuals and the health care system. Looking at the data collected by the U.S.A. National Spinal Cord Injury Statistical Center, a recent estimation showed that the annual incidence of spinal cord injury (SCI) is approximately 54 cases per one million people in the United States, or about 17,730 new SCI cases each year. The estimated number of people with SCI actually living in the United States is approximately 291,000 persons, with a range from 249,000 to 363,000 persons. Young adults are in the higher risk, in fact the average age at injury has increased from 29 years during the 1970s to 43 years recently and most of new SCI patients are male (about 78%) (NationalSpinalCordInjuryStatisticalCenter, 2019). In Italy, the study of Gruppo Italiano Studio Epidemiologico Mielolesioni shows that the incidence of spinal cord injury (SCI) is estimated to be 2500 cases per year with a prevalence of 60,000 to 70,000 cases for a population of 60 million (Baricich, et al., 2017).

The complexity of the life experience of SCI involves social, economic and health care sphere. This injury may make a person dependent on caregivers relatively to mobility, communication, self-care or domestic activities. It can lead to significant signs of depression (that has a deleterious impact on improvements and rehabilitation), accompanied and worsened by misconceptions, negative attitudes and physical barriers that in turn result in the exclusion of many people from full participation in society (WHO, 2013). Despite being a low-prevalence condition respect to others, SCI has wider implications for the health care. A patient with SCI will experience nearly every clinical setting provided: emergency services, intensive care, surgery, stabilizing medical care, and particularly rehabilitation, including return to the community, vocational rehabilitation and ongoing primary care. This results in a high weight on the system. On the other side, since it has to touch different fields, SCI care could provide evidences about the adequacy of a country's services, systems and policies and so help clinicians, health professionals, researchers and policymakers to



understand the strengths and weaknesses of their health-care system (Bickenbach, et al., 2013).

Regarding the economic aspect, the lifetime financial burden brought by SCI conditions can be massive for patients, insurance companies and hospital systems. The costs associated are greatly affected by both the patient's extent of injury and the subsequent degree of disability, regarding the acute and post-acute phases (imaging examination, prognosis and surgical intervention) as well as the chronic phase within the context of longer-term rehabilitation (Merritt, Taylor, Yelton, & Ray, 2019). The estimation (shown in Table 1.3) of the U.S.A. National Spinal Cord Injury Statistical Center of the average yearly expenses (health care costs and living expenses) and the lifetime costs directly attributable to SCI (not including any indirect costs such as losses in wages, fringe benefits, and productivity, averaged \$76,327 per year in 2018 dollars), are interested by a great variation due to education, neurological impairment, and pre-injury employment history (NationalSpinalCordInjuryStatisticalCenter, 2019).

Severity of Injury	Average Yearly Expenses (in 2018 dollars)		Estimated Lifetime Costs by Age at Injury (discounted at 2%)	
	First Year	Each Subsequen t Year	25 years old	50 years old
<b>High Tetraplegia (C1–C4) AIS ABC</b>	\$1,129,302	\$196,107	\$5,010,748	\$2,753,822
<b>Low Tetraplegia (C5–C8) AIS ABC</b>	\$816,019	\$120,303	\$3,661,165	\$2,251,944
<b>Paraplegia AIS ABC</b>	\$550,381	\$72,909	\$2,450,234	\$1,608,015
<b>Motor Functional at Any Level AIS D</b>	\$368,562	\$44,766	\$1,674,012	\$1,181,564

**Table 1.3 Estimation of the expenses (U.S.A) (NationalSpinalCordInjuryStatisticalCenter, 2019).**

## **1.2 Current Clinical Solution**

### **1.2.1 Traditional Rehabilitation**

The Spinal Cord Injury has no cure and no fully restorative therapies and, for this reason, prevention represents the best medicine (Thuret, Moon, & Gage, 2006), but it is possible to plan a rehabilitation program in order to recover some functional capabilities.

Nevertheless, treatment and rehabilitation of patients with spinal cord injury is a long, expensive and exhausting process. It needs to involve a team and a patient-centred approach due to the medical, psychosocial and economic complexity of the problem. The overall aim of rehabilitation is to enable each patient to return to a productive and satisfying life: this means different conditions for different people. Nonetheless, the setting of goals for a person with SCI is fraught with difficulties because it relies, at least in part, on physiotherapists' and patients' predictions of likely outcomes. (Harvey L. A., 2016). For these reasons, short- and long-term functional rehabilitation targets must rely on the use of the patients' ASIA scale to assess the level of impairment and to evaluate medical and social status in order to adapt the rehabilitation plan to the possibilities of the patient. In fact, factors such as injury level, age, weight, general health status, motivation and spasticity affect the recovery potential (Nas, Yazmalar, Şah, Aydın, & Öneş, 2015).

The first important step of the recovery process starts shortly after the injury: in the acute phase, medical management of people with SCI is targeted on minimising further neurological damage to the spinal cord and optimising recovery. Stability of the spine is clearly a priority and is established either conservatively with bed rest (with or without traction) or surgically (Harvey L. A., 2016). The rehabilitation in this step is focused in preventing long term complications and consists in passive exercises performed in order to resolve contractures, muscle atrophy and pain. On the other hand, in early rehabilitation another important aspect is the use of pillows, sandbags, plaster splints or even orthotics with the purpose of positioning the joints to protect the articular structure and maintain the optimal muscle tonus (Nas, Yazmalar, Şah,

Aydın, & Öneş, 2015). In addition, there are many other relevant aspects to be considered in the acute phase. They are related to maintaining blood pressure, circulation, respiration, bladder drainage, bowel care, nutrition, body temperature, and to minimising psychological distress for patients and their families (Harvey L. A., 2016). The chronic phase starts as soon as the patient is medically stable after injury. The most important goal in this step is to obtain the independent mobilization for the patient: using targeted exercises, the rehabilitation tries to improve the capability of moving upper or lower limbs. The expectations in the chronic phase or later in domestic context are ensuring the maximum independence related to the level of the patient's injury, integration of the patient to society and maintaining a lifelong treatment (Nas, Yazmalar, Şah, Aydın, & Öneş, 2015).

### **Upper Limbs**

The rehabilitation process can be targeted to specific body parts, depending on the type of injury and the functional priorities of the patient. As highlighted in the previous section, spinal cord injury at a cervical level can result in partial or complete tetraplegia and the higher is the localization on the spinal cord, the worse and more pronounced is the paralysis of the arms, wrists and finger. It is estimated that cervical SCI accounts for approximately more than one third of all people living with SCI (Wyndaele & Wyndaele, 2006).

In contrast with lower limbs, upper limbs comprehend different motile joints that allow to execute several fine movements thanks to a complex neuromuscular control. This result in a considerable and crucial functionality of these elements. Consequently, upper limb and trunk motion impairment caused by SCI have a high impact on the independence and quality of life of the affected person (Dimbwadyo-Terrer, et al., 2016). This is confirmed in the study of Lo et al. (Lo, Tran, Anderson, Craig, & Middleton, 2016) where, given five high priority areas of functioning for the SCI population (arm/hand use, walking, bladder/bowel control, sexual function, and relief of pain), the arm/hand function had the highest preference among the others. The loss of upper limb function and in particular the use of the hands is one of the most

devastating losses an individual can experience because its role is critical in completing basic activities of daily living such as self-feeding, dressing, bathing, and toileting (Snoek, IJzerman, Hermens, Maxwell, & Biering-Sorensen, 2004).

The rehabilitation of the upper limbs is accompanied by the recovery of their functions. This means that each small improvement in motor control is fundamental and translates in important amelioration in the patient's independence and confidence in everyday life (Mekki, Delgado, Fry, Putrino, & Huang, 2018).

Various approaches are being developed and tested in order to maintain and restore the functionality of the upper extremity. Conventional rehabilitation includes physical therapy, massage therapy, drug prescription, orthoses, functional electrical stimulation (Maciejasz, Eschweiler, Gerlach-Hahn, Jansen-Toy, & Leonhardt, 2014) (Piggott, Wagner, & Ziat, 2016) and can be accompanied by upper extremity surgery. Eventually, wanting to organize the rehabilitation process, it is possible to divide it into three phases: acute, subacute, and reconstructive phase (Murphy & Chuinard, 1988).

In the first phase, the patient has a reduced physical capacity because of muscle weakness, loss of autonomic control, reduced activity, and subsequent changes in metabolic and vascular function. In this scenario the goal of rehabilitation is to reverse the debilitating cycle, maintaining range of motion (ROM), improving muscle strength and preventing complications in order to potentiate the functionality of the limbs and ameliorate the patient's independence in mobility and daily life activities (Haisma, et al., 2006). This can be achieved thanks to pharmacological interventions (related to hypertonia or hypotonia), and specific motor exercises. The rehabilitation strategies are then continued in the subacute phase, with some adjustment: progressive increases to the resistance of weight in physical exercise and incorporation of hand grasping and opening movement (Rice, et al., 2016).

The last phase is aimed at improving positioning and stabilization of the arms and the hands' functions, exploiting surgical reconstructive interventions or augmenting voluntary functions functional electrical stimulation (FES) (Rice, et al., 2016). An

example of the latter is given by the FREEHAND® System (Peckham, et al., 2001) that facilitates the movements in order to regain faster the function.

The rehabilitation of SCI falls within the neurorehabilitation, a complex medical process with the aim of recovering the injuries regarding nervous system and also minimizing any resulting functional alteration. The past three decades have seen a shift in the focus of neurorehabilitation from the use of compensatory approaches to allow and support the function toward an emphasis on functional neuro-recovery which aim is promoting the restoration of function through use of the affected limbs (Musselman, Shah, & Zariffa, 2018). Despite major advances in understanding the brain, neuroscience research is far from the level needed to place new neurons and create connections and so restore a lost function. However, there is another way to create functionally appropriate neural connections: plasticity shows that the brain continuously learns and remodels its neural circuitry in order to encode new experiences and enable behavioral change (Kleim & Jones, 2008).

Plasticity is the ability of neurons to rearrange their anatomical and functional connectivity in response to environmental input, thereby achieving new or modified outputs, namely behaviours. Following injury, neurons can spontaneously increase their plasticity, thereby providing the potential to create new networks as the basis for recovery and compensatory behaviour (Dunlop, 2008). The process of plasticity consists of changes in the activation pattern either of structure or function that involves alteration of the strength of old and new connections (Ding, Kastin, & Pan, 2005). At the anatomical level, spontaneous injury induced plasticity includes regenerative sprouting from damaged and intact neurons, synaptogenesis and synaptic remodelling. On the other hand, at the functional level, plasticity includes changes in neuronal excitability and inhibition, in conduction velocity and in synaptic efficacy (Dunlop, 2008). There is considerable evidence that cortical representation of body parts is continuously modulated and reorganized in response to activity, behaviour, and skill acquisition (Ding, Kastin, & Pan, 2005). Furthermore, reorganization of neuron circuit and change in activation occurs not only in the cortex, but at various hierarchical levels

in the central nervous system during the spontaneous functional recovery after the spinal cord injury (Isa & Nishimura, 2014).

Conversely, neural circuits not actively engaged in task performance for an extended period of time begin to degrade. This concept is known as “use it or lose it”, opposite to the one of “use it and improve it” : both are important for rehabilitation because show how behavioural experiences and extensive training after a damage can induce plasticity and protect neurons and networks that would otherwise be lost after the injury (Kleim & Jones, 2008).

The understanding of the potential of neuroplasticity following a SCI has contributed to the development of intensive physical interventions that aim to promote neuro-recovery through repetitive movement training (Musselman, Shah, & Zariffa, 2018). In fact, plasticity is a fundamental concept which, however, needs to be accompanied and supported by other essential concepts in neurorehabilitation: motivation of the patient, repetition of the exercise and feedback. Repetition focuses on strengthening and reteaching neural circuits that survived the injury. It helps familiarize with movements: the more repetitions, the easier the task gets. Nevertheless, the bare repetition of the same movement could be not positive, since it could easily get the patient bored. For this reason, therapies that continuously present new challenges and stimuli give better results. The psychological aspect of the rehabilitation is an element not to be underestimate: the full participation of the patient to the rehabilitation programs optimizes the motor recovery. Moreover, shaping the exercises to personal goals (of course considering its capacities and introducing difficulties) increases the motivation and constancy.

The best way to fasten the process is involving the user in the rehabilitation process making him/her aware of the performances, achievements and errors during or at the end of the session. Tactile, visual, proprioceptive and acoustic feedback enhance the patient engagement because they allow to receive responses to the action and so represent fundamental tools.

## **Hands**

Limited upper limb function is frequently more disabling with respect to the inability to walk. In particular the role of the hands appears to be fundamental in the every-day life and could reduce dependencies on the caregiver: even with few movements of the finger, a tetraplegic person could accomplish different useful task such as press a switch or interact with a keyboard to communicate or hold an object (Harvey, Dunlop, Churilov, Hsueh, & Galea, 2016). Among others, pinching and grasping movements are essential gestures in everyday life situation.

There exist different methods for making principally motor function assessments of the hand. For example, the Block and Box Test (BBT) evaluates manual dexterity proposing a test where the goal is to move as many blocks as possible from one compartment to another within one minute. Another one is the Jebsen Hand Function Test (JHFT) that require participants to turn cards, pick up physical objects, write, and simulate feeding. The completion time of each task is observed up to a 120-second time limit and then is given a score: the lower the score, the higher is the level of motor functioning (ShirleyRyanAbilityLab, 2012).

After a SCI injury, compensatory techniques, adaptations, auxiliary devices, and special training can be used to improve these capabilities. For example, one important rehabilitative approach is to enhance the effect of the active tenodesis grasp function (active wrist extension that results in passive finger flexion). Moreover, patients with SCI are taught compensatory movements and techniques that help them to interact with objects in different ways (Mangold, Keller, Curt, & Dietz, 2005). Nonetheless, the best outcome would be reobtaining the complete or partial control of the gestures. For this purpose, the rehabilitation program often relies on intensive task-specific training. It is believed that, especially for the individuals with poor grasp, superimposing functional electrical stimulation (FES) may be particularly effective. This combination of therapies provides the damaged spinal cord with excitation from the sensorimotor cortex along with intensive sensory input from the periphery, thus promoting neural plasticity and neurophysiologic and structural re-organisation of the

relevant pathways. Thereby modest improvements in hand function could have life changing consequences (Harvey, Dunlop, Churilov, Hsueh, & Galea, 2016).

### **1.2.2 Robotic and Virtual Reality Rehabilitation**

The traditional arm and hand therapies, thanks to the intense research and trials, seems to be effective and to lead to good results, obviously if perpetrated in the right way. Thanks to the plasticity, the CNS is capable of readapting itself and to recover partially the upper limb functions after a traumatic event such as SCI. Nevertheless, as stressed above, to achieve these effects a fundamental aspect is the repetition of the exercises to strengthening the neural circuit and to start the process of recovery. Intense repetitions of coordinated motor activities constitute a significant burden for the therapists assisting patients because they are time-consuming. It is necessary also to take into account the economic burden of a long rehabilitation process linked to the health care cost, the fact that the availability of therapists is limited and that the positive outcomes also depend on their performance. This set of limitations clearly represent a drawback of the traditional therapy.

In addition, it has been shown that rehabilitation of the neurological injury subjects depends upon the effort required, a principle which emphasizes that SCI individuals need to maintain a high degree of participation and involvement to facilitate motor learning (Karimi, 2013). With a traditional rehabilitation process this engagement result to be doubtful since the proposed exercises are monotonous and not attractive.

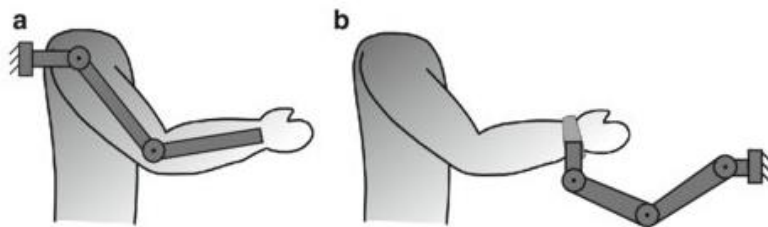
To counteract these problems, different researches have been conducted in order to find technologies useful in the context of new therapy approaches which take into account the client-dependent versatility of training goals, the limited amount of time available, the necessary prerequisites related to physiology and motor learning, the motivation of the patient and his/her level of challenge (Vanmulken, Spooren, Bongers, & Seelen, 2015). Among these technologies, there is an increasing interest in the use of robotics and virtual reality in the neurorehabilitation therapies.

Rehabilitation robotics is a branch of robotics with the aim to overcome the limits of classical rehabilitation. It comprehends the treatment of the upper extremities



following spinal cord injury. These technologies typically target either the shoulder and elbow, or the wrist and fingers, and can be categorized depending on the structure and characteristics as (Duret, Grosmaire, & Krebs, 2019):

- **Exoskeletons.** Exoskeletons are devices that align with and support the articulation of targeted joint(s). They are designed to resemble human anatomy and their structure offer the advantage of precisely recording and monitoring isolated joint movements of the arm and wrist. Exoskeletons can be further classified as grounded and ungrounded robots. The lasts can be worn like a costume and are attached only to the body of the patient allowing more naturalistic movements while the grounded provide more flexibility in the selection and design simplicity (Pehlivan, Lee, & O'Malley, 2012).
- **Robotic end-effectors.** Robotic end-effectors simply contact users at the distal part of their limb. They usually vary from having two to six degrees of freedom (DOF) but cannot apply torques to specific joints of the arm. End effector are usually used for grasping or strength tasks and can be paired with exoskeletons to increase the number of rehabilitations exercises (Piggott, Wagner, & Ziat, 2016).



**Figure 1.3 Robotic devices for upper limb rehabilitation: a) exoskeletons, b) end effector. (Bertomeu-Motos, et al., 2018).**



**Figure 1.4 Left image: Exoskeleton-Armeo spring (Hocoma, 2019) .  
Right image: InMotion Arm (Macovei & Doroftei, 2016).**

Even if these categories are described separately, they can be combined. The choice of the proper device depends on an individual's motor and sensory deficiencies (Piggott, Wagner, & Ziat, 2016).

Another classification of the devices for upper limb rehabilitation relies on the type of assistance they can provide (Maciejasz, Eschweiler, Gerlach-Hahn, Jansen-Toy, & Leonhardt, 2014):

- **Active device.** A device able to move limbs. Under such condition, this device requires active actuators which may increase the weight. It may also apply to subjects completely unable to move their limb.
- **Passive device.** A device unable to move limbs but may resist the movement when exerted in the wrong direction. This type of device may only be used for rehabilitation of subjects able to move their limbs. It is usually lighter than active device since it possesses no actuators other than brake.
- **Haptic device.** A device that interfaces with the user through the sense of touch. In most cases it provides some amount of resistive force, often also some other sensation like vibrations. It is sometimes also able to generate specific movements. However, the force it generates is usually small. Haptic devices are commonly used in rehabilitation settings with virtual environments.

- **Coaching device.** A device that neither assists nor resists movement. However, it can track the movement and provide feedback related to the performance of the subject. As haptic devices, coaching devices are also commonly used in rehabilitation settings with virtual environments.

However, despite the option chosen, using robotic models increases the number of training sessions with consistent repetitions, delivering a high volume of high-quality movement and so promoting functional recovery and enhancing adaptive plasticity (Kadivar, et al., 2011).

Moreover, automated training, increasing the duration and number of rehabilitation sessions, reduce the number of therapists required per patient thus significantly reducing the burden of care and personnel cost (Riener, 2013).

Robotics also enables the performance evaluation of patients during therapy, and objective and quantitative assessment after therapy. This is one of the features that these devices should include, together with safety, cost-efficient, comfort, varying modes to allow for training customization and to allow natural and free movement (Singh, et al., 2018).

In addition, therapy programs can be automated or semiautomated and do not require a close supervision or support from physical or occupational therapists. In this way, these could be potentially delivered as home therapy programs with the condition of smaller, more portable, and less expensive devices (Yozbatiran & Francisco, 2019).

Among the new therapies, Virtual reality (VR) based neurorehabilitation is a novel and attractive technology that allows users to interact in three dimensions with a computer-generated setting (a virtual world). VR is a computer-based, interactive, multisensory simulation environment that occurs in real time and thanks to a human-computer interface allows the users to engage with what appears to be real world objects (simulated) and events in the virtual reality (Prasad, Aikat, Labani, & Khanna, 2018). The important concept brought by VR rehabilitation are interaction and immersion. The interaction is the possibility to interact and participate in real time with the virtual world, therefore not being a mere inactive actor that only passively visualize the graphic representation. The immersion is the ability of the devices to induce an

inclusive, extensive, surrounding, and vivid illusion of reality to the senses of a human participant. It is possible to combine different technologies to set up a whole system capable of delivering mixed visual, acoustical and tactile information in real time (Tierì, Morone, Paolucci, & Iosa, 2018). A great variety of interfaces are used in order to interact to the virtual scene and they go from low level of natural interactions with the mouse, keyboard or joysticks to high level of natural interactions with full body motion tracking or haptic devices that mimics the tactile feedback while touching an object (Peñasco-Martín, et al., 2010). Compared to traditional rehabilitation, VR systems bring advantages to the recovery process. First of all, due to the possibility given by the computer, it can establish countless different fun and motivating environments and develop plenty of skills and task-based techniques that can stimulate and enhance participants interest and motivation, while maintaining a safe environment. (Prasad, Aikat, Labani, & Khanna, 2018). In addition the therapists exercises more control over the environment compared with traditional settings and this enable them not only to create a patient-specific exercise, but also to impose the difficulty of the challenge relying on the treated level of motor deficiency and to monitor and record the performance of the participants for further analysis (Masseti, et al., 2018). Regarding the economic sphere, even if the cost of these VR systems previously limited their accessibility by the general public, nowadays it is more affordable (Yeo, Chau, Chi, Ruckle, & Ta, 2019).

If the two new therapies, both robotics and VR, have a good impact on the rehabilitation of the upper limb in a SCI patient, joining them result in a very effective improvement of the recovery with respect to traditional rehabilitation. In this scenario, the robotic setup represents the interface with the virtual environment: the patient has the opportunity to accomplish the rehabilitation tasks imposed in the virtual world by moving the robotic system. Another important combination of the two allows giving haptic feedback to the patient through the robotic system while he/she is freely moving in the VR environment.

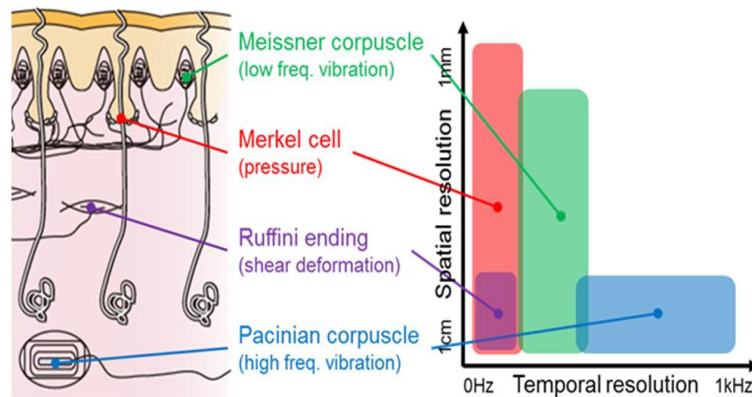
### 1.2.3 Haptic Feedback

The term haptics derives from the Greek words “haptikos” and “haptesthai” and means “relative to the sense of touch”. Haptic feedback is the delivery of somatosensory stimuli to the user via physical interfaces. Human touch sensation is a result of activities of four types of mechanoreceptors in the skin, corresponding to four tactile sensations. These can be classified respect to the location in the skin that determines the receptive field, hence superficial (small receptive field) or profound (large receptive field), but also respect to the rate of adaptation. Stimulation of fast adapting (FA) receptors is associated with the perception of vibration, whereas activation of slow adapting (SA) units is associated with perceptions of pressure, position and skin displacement. As shown in Figure 1.5, in the superficial layers there are Merkel cells for pressure and Meissner’s corpuscles for low frequency vibration while in deep layers Pacinian corpuscles for high-frequency vibration and Ruffini endings for shear deformation. If these four receptors could be selectively activated, individually or in combination, any tactile sensation could be simulated. This is similar to how the primary colors (red, green and blue) in vision are perceived and combined by the three types of cone cells in the retina (Yem & Kajimoto, 2017).

The provided information is relative to two somatosensory systems (Demain, Metcalf, Merrett, Zheng, & Cunningham, 2012):

- **Kinaesthetic/proprioceptive system.** It provides information about the position and movement of joints and the length and tension of muscles. Quick adapting Pacinian corpuscles (FAII), found in joint capsules and ligaments are maximally active during movement, whilst slow adapting Ruffini corpuscles (SAII) in individual ligaments provide information on joint position and are maximally sensitive at the end of joint range.
- **Tactile/cutaneous system.** It conveys what is called the sense of touch. The receptors of the tactile system are found in the skin over the entire body but are most prevalent in the hand. The system transmits information about various physical parameters mostly related to the surface properties of objects, but it also responds to physical parameters related to the perception of pain.

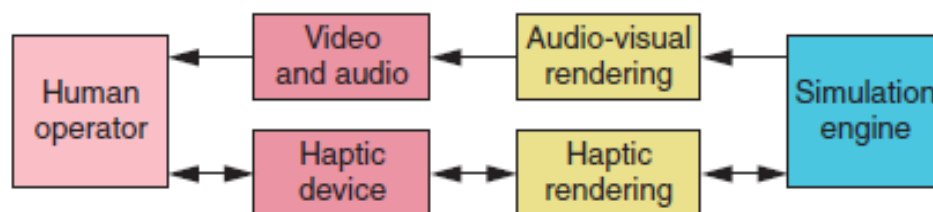
Thermal properties and pain are detected thanks to respectively thermal receptors and nociceptors. This system conveys info of pressure, position and skin displacement thanks to Merkel cells and Ruffini endings, but also sensation of vibration thanks to Meissner and Pacinian corpuscles (see Figure 1.5).



**Figure 1.5 Mechanoreceptors (Yem & Kajimoto, 2017).**

Human somatosensory system can be stimulated using a haptic feedback device in combination with a virtual environment. This allows to feel a full-immersion experience because it adds kinesthesia or tactile feedback to the visual and acoustic feedback, thus enabling the patients to interact with the virtual reality and perceive the effect of their action in real time.

The technology architecture is composed of different elements that mutually interact: the simulation engine responsible for computing the virtual environment's behaviour over time, the feedback algorithms that compute virtual environment's graphic, sound



**Figure 1.6 Basic architecture for a virtual reality application incorporating visual, auditory, and haptic feedback. (Salisbury, Conti, & Barbagli, 2004).**

and force responses and the transducers that convert these signals into a form that the operator can perceive (see Figure 1.6). Usually, the users wear or holds the haptic device that can not only provide forces but also get information from the user (e.g. position), while audio and visual feedbacks are given by loudspeakers and displays. In this architecture, it is important to understand the role and categorization of the haptic device. As already said, it represents the physical interface where energy is exchanged between the operator and the device itself. There are different ways of classifying haptic devices: first of all, their grounding location. The haptic device can be in contact with different parts of the user's body. However, it typically interfaces with hands: feedback gloves cannot reproduce the object weight or inertial forces but can give contact information, while others handheld devices give just the vibratory effects (used typically in gaming industry). Other types are exoskeletons that the user can wear or ground-based devices, much more complex and complete in providing feedback. The devices can also be differentiated by their number of degrees of freedom (DOF), that is the possible movements they can allow (Salisbury, Conti, & Barbagli, 2004). A critical classification is based on their intrinsic mechanical behaviour. There are two fundamental and dominant control methods widely used in haptic interface design: admittance and impedance control. **Admittance** control has as input the measurement of the force exerted by the human user and as output the variation of motion (position or in case velocity) of the master device. **Impedance** control has as input the measurement of the position of the haptic device imposed by the user and as output the variation the amount of force applied by the master device in response (Wen, Neculescu, & Sasiadek, 2008).

The integration of the haptic system in the virtual environment is made thanks to haptic rendering algorithms. In the virtual space, the haptic interface is represented as an avatar, a virtual object that the user can control changing its position in the virtual environment by changing the position of the physical object in the real world. Obviously, the choice of how to model the avatar depends on the intended application and the device's capabilities. The haptic rendering algorithm detects the collisions between virtual objects and the avatar thanks to the *collision-detection algorithm*,

giving information about the type, the position, the time and the extension of collision. Then thanks to the *force-response algorithm*, the haptic rendering computes the force of interaction in a collision, trying to simulate the forces that would be generated in a real case and returning typically force vectors. The last algorithm, the *control-algorithms*, receives the computed forces and governs the haptic devices trying to minimize the error between the ideal forces received and the applicable forces (Salisbury, Conti, & Barbagli, 2004). In conclusion, the opportunity of receiving haptic feedback is useful in different fields of interest including gaming, surgery and rehabilitation.

### 1.3 State of the Art

As stated in the previous section, the loss of upper limb functions and in particular those of the hand is a devastating condition that has a high impact on the quality of life of those affected. As a matter of fact, rehabilitation focusing on the distal portion of the limb leads to functional gains in the proximal limb. However, this is not possible the other way around. This suggest that upper-limb rehabilitation should necessarily focus on the functional recovery of the hand (Piggott, Wagner, & Ziat, 2016). As already highlighted, traditional rehabilitation is effective but has a set of drawbacks that impedes to perform a continuous and well controlled rehabilitation. Robotics is currently proposed as a non-invasive solution that can overcome the problem of the traditional rehabilitation and can be combined with it to improve the efficacy of training sessions and ameliorate the patient quality of life. When used in conjunction with computer based virtual reality, robotic rehabilitation seems to be effective because together they can provide the opportunity to set up a rehabilitation platform capable of ensuring frequent exercises sessions also at home. Moreover, the use of haptic devices improves somatosensory recovery. Somatosensory-motor integration is demonstrated to be fundamental and so ameliorating the somatosensory system would improve motor recovery. Consequently, different solutions have been studied in order to enhance the rehabilitation using both virtual platforms and wearable, actuated haptic devices.

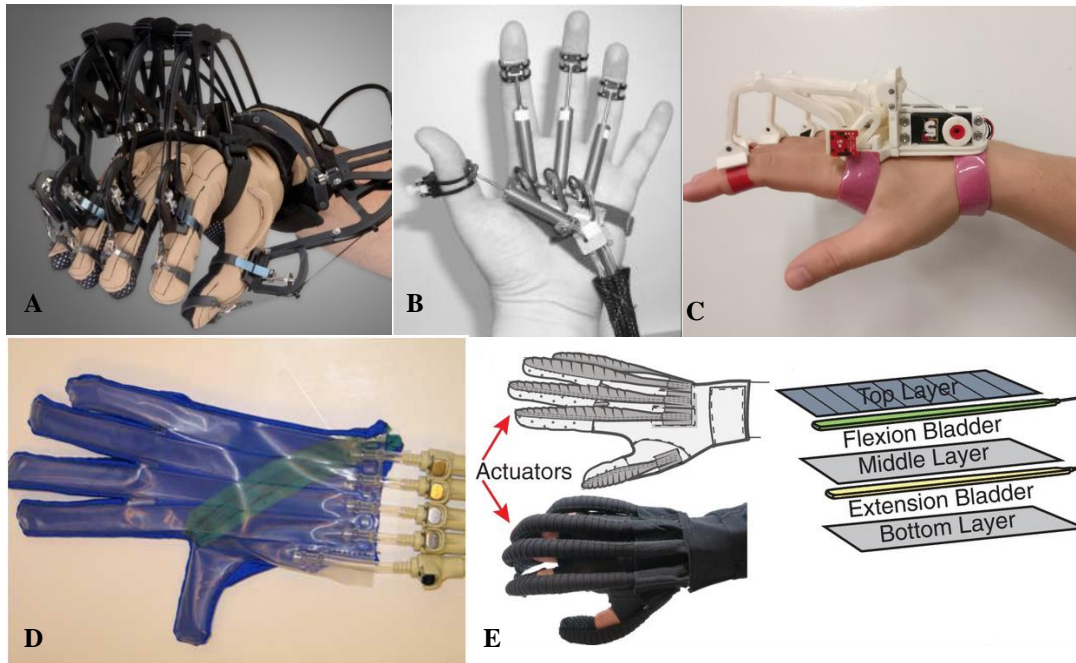


The main objective of including haptic feedback during neurorehabilitation is to allow participants to train two different parts of the somatosensory system: kinaesthetic/proprioceptive and tactile/cutaneous system. Both are used in different tasks for hand rehabilitation (grasping an object, determining grip strength, texture discrimination, identification of sticky or slippery objects, and sensations of warmth or cold) and each of them is trained by different types of haptic tools. There exist other devices and techniques used in rehabilitation which do not have as a primary aim somatosensory stimulation but nevertheless provide sensory feedback by promoting motor activity (Demain, Metcalf, Merrett, Zheng, & Cunningham, 2012).

### **1.3.1 Kinaesthetic Feedback Devices**

The idea behind Kinaesthetic feedback devices is to provide forces which guide, resist or interfere with movement and to deliver force feedback in response to physical properties or movements of virtual objects. To achieve this, different technologies have been designed and used to interface the user to the virtual environment. The user finger movements and action are analysed in virtual reality by the haptic rendering algorithm that in turn control mechanisms such as mechanical levers or pneumatic actuators connected to the user hand and fingers. These provide the forces generated by contacts with virtual objects. In this way, human joint capsules and ligaments are stimulated together with their corpuscles, allowing the patient to sense the object weight and size (Demain, Metcalf, Merrett, Zheng, & Cunningham, 2012). Gloves with external exoskeletons and endpoint devices are typical examples of kinaesthetic devices (Piggott, Wagner, & Ziat, 2016).

It is important to highlight that there are many examples where the rehabilitation of the upper limb part is focused on the recovery after different types of neurological diseases. All these approaches and their results can easily be transposed to SCI, since the injury results and the affected control of the limbs are the same at a motor and sensory level.



**Figure 1.7** Kinaesthetic devices: exoskeletons. A) CyberGrasp (CyberGloveSystems, s.d.). B) Rutgers Master II- ND (Bouzit, G., Popescu, & Boian, 2002). C) (Secciani, et al., 2019). D) PneuGlove (Connelly, et al., 2010). E) Soft Robotic Glove (Cappello, et al., 2018).

Viau et al. (Viau, Feldman, McFadyen, & Levin, 2005) compared the performances achieved in terms of movements done in physical and virtual environments in adults with motor deficits (hemiparesis after stroke) with respect to those in healthy individuals. In both experimental setups, they analysed the movements when reaching, grasping, transporting and releasing a ball. Temporal and spatial parameters of reaching and grasping were determined for each phase. Results showed that the both groups used similar movement strategies in both environments, but participants with hemiparesis made slower movements and more curve trajectories compared to healthy subjects. The virtual environment included interaction with a ball on a 2D computer screen with real movements translated in VR movements thanks to a fibre optic glove and an electromagnetic sensor. To enable the subject to "feel" the virtual ball, the force feedback device CyberGrasp (CyberGloveSystems, s.d.) delivered prehension force feedback in the form of extension forces to the distal phalanxes of the thumb and each finger (see Figure 1.7 A).

Similarly, Adamovich et al. (Adamovich, et al., 2005) and Merians et al. (Merians, et al., 2002) targeted VR simulations for range of motion, movement speed, fractionation, and force production in hand rehabilitation of chronic phase following stroke. In this case the haptic device used was the Rutgers Master II-ND, an exoskeleton that provides one degree of force feedback per finger, apart from the pinkie. Differently to the CyberGrasp, it does not require a separate sensing glove to collect position data to recognize and place the hand in a VR since its exoskeleton incorporates non-contact position sensors. The main difference is that instead of controlling the fingers from the upper side, the Rutgers Master II uses a direct-drive configuration and compressed air, such that each fingertip is resisted in flexion from the lower side (see Figure 1.7 B). More recently, research has focused on obtaining lighter and less intrusive devices. Connelly et al. (Connelly, et al., 2010) developed the “PneuGlove” (see Figure 1.7 D). This pneumatic glove can provide independent extension assistance to each digit while still allowing full arm movement. It was designed to take advantage of the residual control of finger flexion observed in some cases after neurological damages, and, thus, to assist only digit extension. It can be used for training grasp-and-release movements either with real objects or with virtual objects in a virtual reality environment. In this study they proved it in combination with a virtual environment where different object were presented with the aim of performing grasp-and-release tasks in a given time. The same principle is used and improved in recent years in the device developed by Cappello (Cappello, et al., 2018). It consists in a base glove with attachments points (hook and loop fasteners, straps) for modular, independent finger actuators (see Figure 1.7 E). Each fabric-based actuator is comprised of three fabric layers and two air-tight bladders placed between each fabric pocket. Finger flexion and extension is obtained by selectively pressurizing these bladders with an air pump. Nevertheless, the idea of traditional exoskeletons for hand rehabilitations has not been abandoned. Alongside commercial devices there exist some prototypes in this field. Secciani et al. (Secciani, et al., 2019) for example concentrated their work on the development of portable, wearable and highly customizable hand exoskeletons (see figure 1.7 C).

Endpoint or also called end-effector devices are grounded. Even if they are not wearable as ungrounded exoskeletons, they can have different number of DOF to afford a variety of interactions and can be integrated with other haptic devices to perform different tasks. Commonly the user interacts with the virtual environment by moving a particular grip that depends on the device chosen, receiving force feedback when an interaction occurs.

A typical example of end effector device is the PHANToM (see Figure 1.8), manufactured by SensAble (currently Geomagic) in different versions with different DOF. It uses an impedance control paradigm : thanks to the pen like stylus grip, it tracks the pitch, roll and yaw and x, y and z Cartesian coordinates of the virtual point-imposed by the user and transmit forces back to the user's fingertips as it detects collisions with virtual 3-D objects. In rehabilitation, the phantom has been used in



**Figure 1.8 Kinaesthetic devices: End-Effector. Geomagic PHANToM on the left and Novint Falcon on the right. (Piggott, Wagner, & Ziat, 2016)**

several tasks such as the 3D Bricks game, the block and ball task in TBI patients, pursuit tasks designed to increase attention and maze navigation.

Another popular device is the Novint Falcon (see Figure 1.8). It is a three-DOF haptic device originally designed for the gaming industry by Novint. Different removable grips such as a pistol or a pen holder can be mounted on its arm that can extend, retract, and fold in a workspace of  $101.6 \times 101.6 \times 101.6$  mm. As other devices, it allows to control a point in VR, to measure displacements in space and to provide not only force

feedback, but also vibration. Its characteristics make it useful for multiple tasks commonly used in neurorehabilitation research (Piggott, Wagner, & Ziat, 2016).

Andaluz et al. (Andaluz, et al., 2016) used the Novint Falcon in combination with an infrared camera mounted on an Oculus Rift to build up a virtual reality environment for patients with diseases of mobility in the upper extremities. It is an entertainment system based on videogames that trains interaction abilities based on immersion. Moreover, the system can follow trajectories that are affected by force feedback provided by the haptic device and prepare them to a posterior analysis of pertinent specialists. In another study, Yeh et al. (Yeh, 2017) developed a haptic-enhanced VR system to simulate haptic pinch tasks to assist the recovery of upper-extremity fine motor function. In this setup, two Novint Falcon devices were used in coordination to simulate the haptic perceptions of 2 fingertips. The platform provided two principal tasks: the first one was a “pinch strengthening” task, in which participants were asked to grip a virtual box by using 2 fingers (the thumb + each finger). The participants were asked to gradually increase the pinch strength until reaching the default strength threshold imposed in the simulation. The second task was a “pinch and lift” task, in which participants were required to grip a virtual box with two fingers (thumb + index and thumb + middle fingers) and lift the weighted box to a default height. In both tasks, the haptic simulation was implemented as kinaesthetic feedback: the participants perceived the reaction force of the surface and/or the weight of the box.

### **1.3.2 Tactile Feedback Devices**

The tactile feedback device represents the other type of haptic device used in rehabilitation. It differs from the kinaesthetic feedback since it stimulates the receptors linked to the sense of touch. Tactile devices are used principally to convey information relative to an object displayed in virtual reality, trying to replicate its tactile parameters including temperature, texture, shape and roughness. Several mechanisms including the application of an electric or magnetic field, the use of piezoelectric crystals, shape memory alloys or pneumatic systems, or the use of fluids which change viscosity in different environments have the potential for providing these kinds of sensations. The

main methods used to produce a feedback are the application of pressure, vibration, electrical fields or changes in thermal flow. In neurorehabilitation, the largest part of haptic device technologies has focused on providing kinaesthetic feedback while few have used tactile stimulation with greater emphasis on vibrotactile or electro-tactile stimulation (Demain, Metcalf, Merrett, Zheng, & Cunningham, 2012).

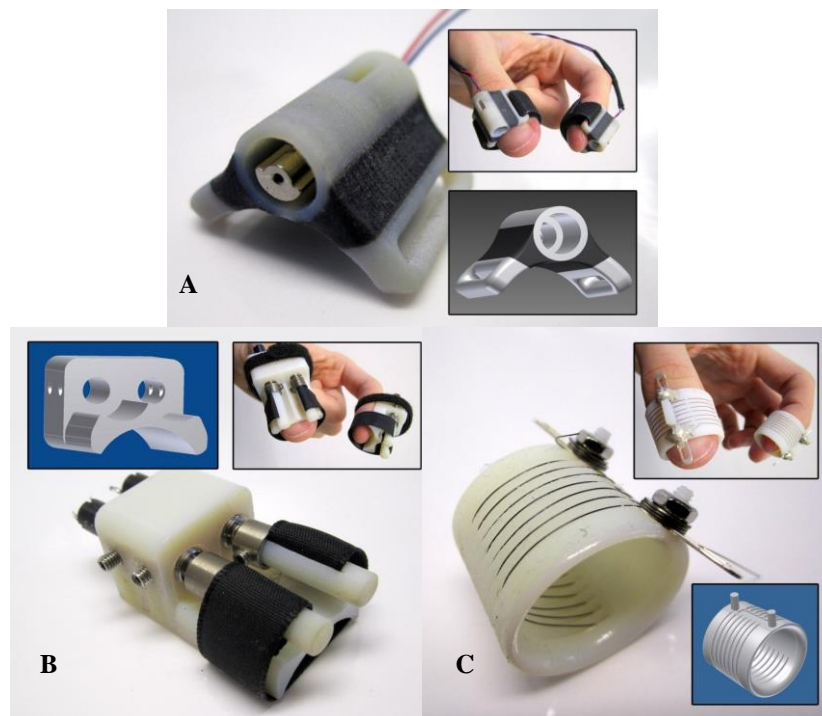
Vibro-tactile feedback is the most widely used in different application fields including mobile phones and game controllers but thanks to its ease of implementation and use it is employed in different fields with different purposes. The CyberTouch system (see Figure 1.9) is a general purpose commercial haptic glove designed and implemented for different virtual reality applications, including computer aided design and robotic surgery. This system represents a feedback option that is mounted on the CyberGlove, an instrumented glove that provides up to 22 high-accuracy joint-angle using resistive bend-sensing technology to transform hand and finger motions into real-time digital



**Figure 1.9 CyberTouch (Piggott, Wagner, & Ziat, 2016) and integration with virtual reality (Dimbwadyo-Terrer, et al., 2016).**

joint-angle data. The CyberTouch system equips the glove with a vibro-tactile stimulator on each finger and the palm in order to enable users to manually experience interactions with computer generated 3D objects. This haptic glove was used by Dimbwadyo-Terrer et al. (Dimbwadyo-Terrer, et al., 2016), combined with a virtual reality (VR) environment, for therapeutic training of reaching movements after spinal cord injury. The virtual scene consisted of one room, two shelves and a trench (see Figure 1.9) where the user is able to see his/her virtual hand in the scene, as well as virtual objects. The virtual tasks required the patient to reach and grab an object with different settings between trials in order to improve the functional reaching movement

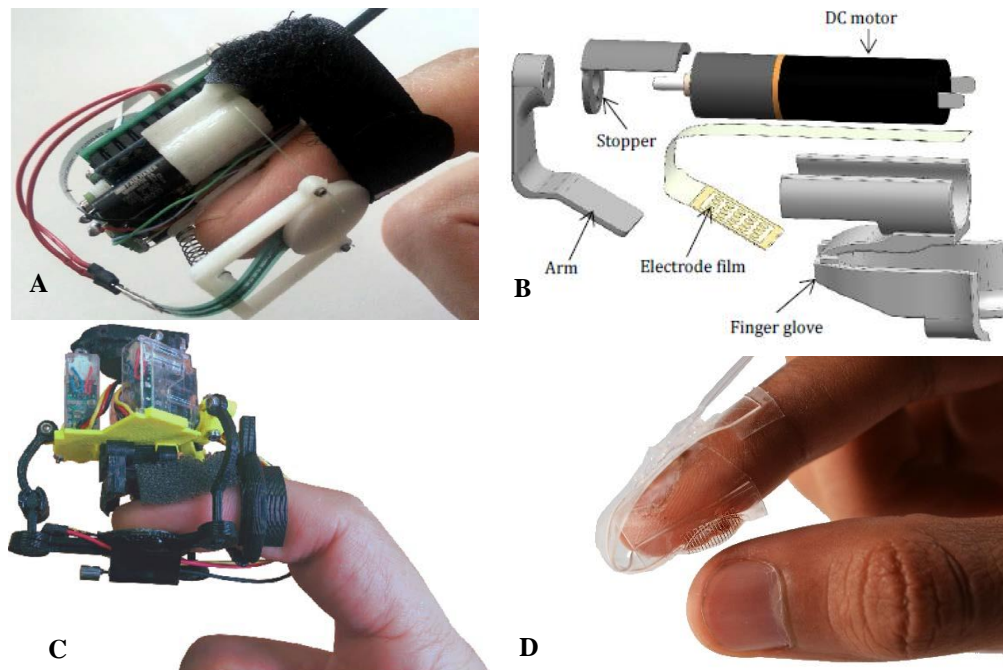
thanks to the support of the tactile feedback provided by the glove. Nonetheless, wearing a glove can sometimes be difficult and that is why researches have shifted the focus on thimbles, smaller devices easier to wear and more “realistic” in the feedback they can provide. Merret et al. (Merrett, et al., 2011) evaluated three different tactile feedback thimbles thought to be combined with virtual reality to enable users to perceive the sensation of static interaction with virtual objects during rehabilitation sessions aimed to promote recovery from neurological illnesses. In particular, they redesigned and qualitatively evaluated with trials a vibration motor thimble, a motor-driven “squeezer” and a shape memory alloy thimble (see Figure 1.10). To generate a vibration in a small and lightweight device, they opted to use a miniature bar-type vibration motor able to rotate the offset mass in a 3D printed structure designed for this specific purpose. The “squeezer” consists of two miniature motors, controlled monitoring the current passing through them, used to pull a flexible band to apply compression to the pulp of the fingertip. The last model applies the same principle, but



**Figure 1.10 Thimble devices. A) Vibration motor thimble. B) Motor-driven “squeezer”. C) Shape memory alloy thimble. (Merrett, et al., 2011).**

the compression is obtained by controlling the capability of shape memory alloys to shorten in length when heated. Comparisons between the devices showed the SMA to be more convincing in delivering sensations comparable to grasping objects (static exploration), whilst the vibration device provided information about object surface properties (dynamic exploration). The squeezer device was found to be more cumbersome than the other devices, and reduced mobility through splinting. In more recent years more complex technologies have been developed with the aim of providing tactile sensations while interacting with virtual world not only focusing on rehabilitation. The FinGAR (Finger Glove for Augmented Reality) is a tactile thimble device (see Figure 1.11 B) developed by Yem and Kajimoto (Yem & Kajimoto, 2017) that combines electrical and mechanical stimulation to selectively stimulate skin sensory mechanoreceptors. A DC motor provides high-frequency vibration and shear deformation to the whole finger and an array of electrodes provide pressure and low-frequency vibration with high spatial resolution. Their results indicated that skin deformation and cathodic stimulation affect macro roughness and hardness, whereas high-frequency vibration and anodic stimulation affect friction and fine roughness. Pacchierotti et al. (Pacchierotti, Tirmizi, & Prattichizzo, 2014) presented a wearable 3-DOF fingertip device for interaction with virtual and remote environments. It consists of two platforms connected by three wires (see Figure 1.11 A) controlled by three small electrical motors, equipped with position encoders. The length of the wires is imposed in order to move and re-angle the mobile platform toward the fingertip. One force sensor is placed in contact with the fingertip at the centre of the platform for closed-loop control. Based on the same principles, the 3RRS wearable fingertip device (Chinello, Malvezzi, Pacchierotti, & Prattichizzo, 2015) is composed of two parallel platforms that are instead connected by three articulated legs, actuated by three motors in order to make and break contact with the skin and move or re-angle the mobile platform toward the fingertip (see Figure 1.11 C). The latest invention in this field was developed by EPFL scientists and consists in a soft, flexible artificial skin made of





**Figure 1.11 Tactile haptic devices. A) (Pacchierotti, Tirmizi, & Prattichizzo, 2014). B) FinGAR (Yem & Kajimoto, 2017). C) (Chinello, Malvezzi, Pacchierotti, & Prattichizzo, 2015). D) Artificial skin. (Pessina, 2019).**

silicone and electrodes (see Figure 1.11 D). The system allows the artificial skin to adapt itself to the user and provide haptic feedback in the form of pressure and vibration. Strain sensors continuously measure the skin's deformation so that the haptic feedback can be adjusted in real time thanks to soft pneumatic actuators that form a membrane layer which can be inflated by pumping air into it (Pessina, 2019). All these devices can be used for the upper limb rehabilitation of spinal cord injury patients if used alongside computer-displayed virtual reality environments. In order to connect them, the system needs to have technologies capable of getting real-time information about the hands position. Kim et al (Kim, Jang, Lee, Lee, & Lee, 2016) integrated IMU sensors on one of these devices and on the hand to track its position in 3-dimensional space. In a different work, Meli et al. (Meli, Scheggi, Pacchierotti, & Prattichizzo, 2015) used a single DoF tactile device in combination with the Leap Motion device (Ultraleap, 2019) in order to allow users to receive a cutaneous feedback while touching objects in VR.

## 1.4 Current Thesis: Motivation and Objectives

The main purpose of this work is to design and implement a low-cost virtual reality platform for upper limb rehabilitation of patients affected by cervical spinal cord injury. The platform takes advantage of an immersive VR system to enhance involvement and engagement of patients, also thanks to the possibility of providing different types of feedback during motor exercises. In particular, the system consists of three different virtual rehabilitation games that allow the patient to interact using hands and to receive a haptic feedback on the index thanks to a motorized haptic thimble device. The idea is to overcome the issues that arise when the patient is asked to wear a haptic feedback glove, as donning and doffing becomes difficult. It is important to highlight that even if the feedback is provided on one finger, the virtual games are implemented with the purpose of training both the hand functions and the arm movements. An important feature relies on the flexibility of virtual scenarios, that allow adapting the motor exercises to the needs of the patient. The work includes the design and implementation of a low-cost prototype of the haptic device capable of providing pressure and vibration sensation to the fingertip.

The platform will be validated as a proof of concept at the National Hospital for Paraplegics (Toledo, Spain) with cervical SCI patients. This platform could be a complementary rehabilitation method for in-home application, used alongside the conventional therapy to lead to better functional outcomes.

For this purpose, the work has been divided into different main parts as follows:

- Chapter 2 defines the design and implementation of the feedback device. It illustrates the mechanic part (design evolution of the elements, motor description, mechanical connections and final assembling), the electronic part (Microcontroller, motor encoder and connections) and the firmware implemented.
- In Chapter 3, the design and implementation of the virtual platform are illustrated. Leap Motion sensors and Unity platform are first introduced and described. The last part shows and explain the main elements and functions of the software developed.

- Chapter 4 shows the results. First, the whole rehabilitation platform is described. Next, the results of the data analysis and questionnaires following the trials with healthy and patients are shown.
- Chapter 5 explains the conclusions and future developments of the work.



## Chapter 2: Haptic Feedback Device

This chapter is aimed to illustrate the motivations that led to the design of the haptic feedback thimble and the process followed to implement it. The design and implementation of the mechanical parts, the connections between them and the motor used will be first explained in detail, trying to give an idea of the difficulties and obstacles encountered. Next, the electronics used and the connections with the motor are shown. The end of the chapter explains the firmware implemented to achieve the desired functions and shows an evaluation of the device.

First it is important to understand the reason why the project involves such a device. The problem encountered by the therapists relies on the fact that for cervical SCI patients wearing a tactile feedback glove (such as the CyberGlove illustrated before) sometimes brings a slight discomfort. Moreover, very usually, this discomfort results in a considerable pain, especially when donning or doffing the device. Therefore, the idea is to take advantage of the same feedback principles but to restrict them from the whole hand to the fingertips. For this reason, this work is aimed to develop a working prototype of a thimble that will be controlled and used in rehabilitation to provide two types of haptic feedback on the fingertip: vibration and pressure.

Moreover, the central objective of the work is to reduce the cost of realization and the complexity of the thimble, to be able to build a device that could be acquired, together with the software, as a low-cost platform. In this way, the whole system would be useful, effective and affordable for the application in home-rehabilitation sessions.

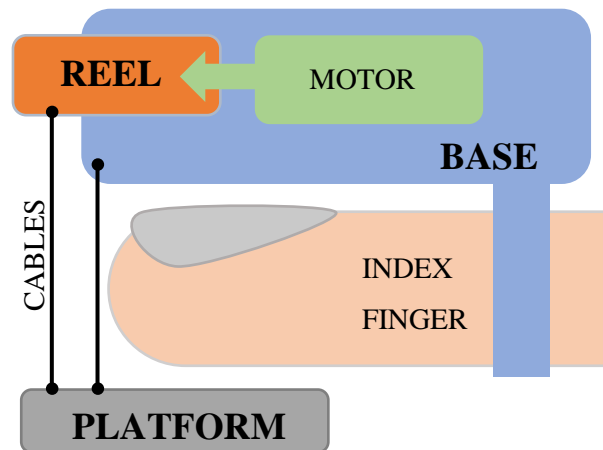
### 2.1 Mechanics

With the purpose of achieving these objectives, it is necessary to establish which characteristics of the device are more important and need to be prioritized, but also to assume that others may be compromised. The main functionality that the device has to accomplish, is to provide a haptic feedback in form of vibration and pressure to the fingertip without interfering with the task imposed in virtual reality. This means that

the design must be minimally invasive not only for the patient, but for the whole system itself. Consequently, the thimble-like device designed for the rehabilitation platform needs to satisfy some important requirements listed below:

- The weight of the thimble must be as light as possible. This because it must be located on the finger. Additional weight would increase the inertia that in turn would affect the hand movements, resulting in erroneous gestures.
- The model must be small in size. A large object in fact would interfere with the hand gestures recognition algorithm of the camera sensor, thus influencing also the feedback provided by the device itself.
- The thimble must be adjustable to different-sized fingers, thus avoiding having a set of different thimble devices for each patient. Connecting and disconnecting every time the device from the platform would in fact result in deterioration of the mechanical and electrical connections and consequently lead to poor performances of the system.
- The connection of the device to the user's finger must not interfere with the haptic perception provided. However, it must be properly firm avoiding the dislodging of the thimble during the rehabilitation sessions. Ergonomic considerations are essential because they ensure a correct sensation of the feedback and minimal influences in the performance.
- The materials used for the prototype must be low-cost, in accordance with the requirements of the thesis. Nonetheless they must be solid enough to manage repetitive mechanical stresses in different points.

Several prototypes have been created while detecting different possible improvements. In the next sections, some of the versions are shown in order to highlight the most important improvements made. Nonetheless, the main concepts have been maintained in all versions. The device is composed of three interconnected parts: the base, the reel and the platform (see Figure 2.1).

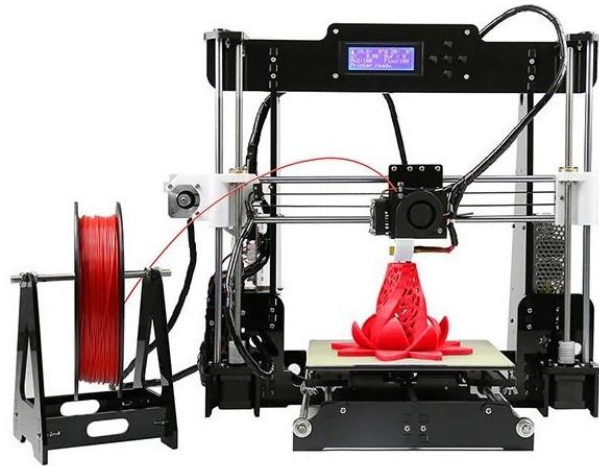


**Figure 2.1 Schematization of the elements that compose the device.**

The main piece, the base, is placed on the back of the fingertip and has the functions of securing the whole structure on the finger and holding the motor in position. The second part is a parallel platform that is pulled through cables to the first part, so it is the one that produces the feedback touching user's fingertip. The third one is directly connected to the motor connector and allows to pull or release the cables when it rotates. Three compression strings are placed vertically between the parallel platforms to maintain the alignment between the first and the second elements and to help the releasing movement of the cable.

Every single piece was first designed using Autodesk Inventor (Autodesk, 2019), a computer-aided design application for three-dimensional mechanical design, simulation, visualization, and documentation. This software facilitates the conception of the prototype by allowing the visualization of the element in different views and by allowing to sketch and quote the pieces in all their parts.

Once designed, every single component was then saved in "stl" format and imported in Ultimaker Cura (Ultimaker, 2019), the management software of 3D printers which allows to tune a series of settings for the printing. Since the pieces for the prototype are significantly small, a perfect tuning of the settings (percentage of internal piece filling, speed and temperature of the extruder or the imposition of external supports.) was necessary to obtain a good quality of printings. The 3D printer used was the



**Figure 2.2 AnetA8 (Shenzhen Anet Technology Co., Ltd.).**

AnetA8 (Shenzhen Anet Technology Co.) (see Fig. 2.2). This is an entry-level, low-cost 3D printer with a 220 x 220 x 240mm<sup>3</sup> printing volume. The printer is modifiable thanks to different 3D printable enhancements available. The Anet A8 comes with a heating bed which is essential for the attaching of the first layer in the printing process. From the first layer, in fact, depends the outcome of the following ones and so of the entire piece.

The use of the 3D printer is fundamental for this work since it allows to evaluate, print different ideas and to solve mistakes at a low production cost. Furthermore, it accelerates the production and modification of the prototype due to the possibility of printing with precision even very little complex parts. Even if some drawbacks of this technology must be considered, such as limited materials that can be used for printing and not enough precision, they are not relevant problems in this first prototyping phase. This solution allows to print the shape designed in the CAD software using solid plastics, hence providing low-cost, custom-made pieces strong enough to resist to mechanical stresses.

### **2.1.1 Design Evolution of The Elements**

All the elements composing the thimble underwent to different improvements because little adjustments were needed to overcome the problems encountered. Only some of



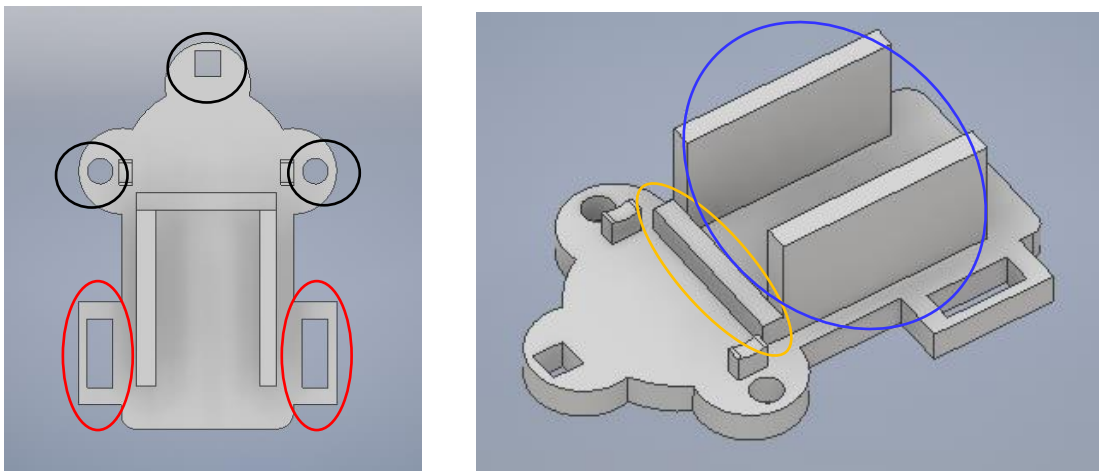
the versions will be shown hereafter with the aim to highlight the most important changes.

### ***The base.***

The first element is the biggest one and the one that had undergone the most changes. It is the base, the piece that is located at the end of the finger. The first of its two functions is securing the whole thimble structure on the finger of the user, trying not to interfere with his/her feedback perception. The second one is to hold the motor in position, avoiding additional movements that would interfere with the proper functioning of the whole device.

It is essential to underline that the design of this part has been realized in order to incorporate a low-cost engine (already present in the laboratory) that will be shown in Section 2.1.2. As a consequence, all element sizes have been adapted to the engine itself in terms of measurements and shapes.

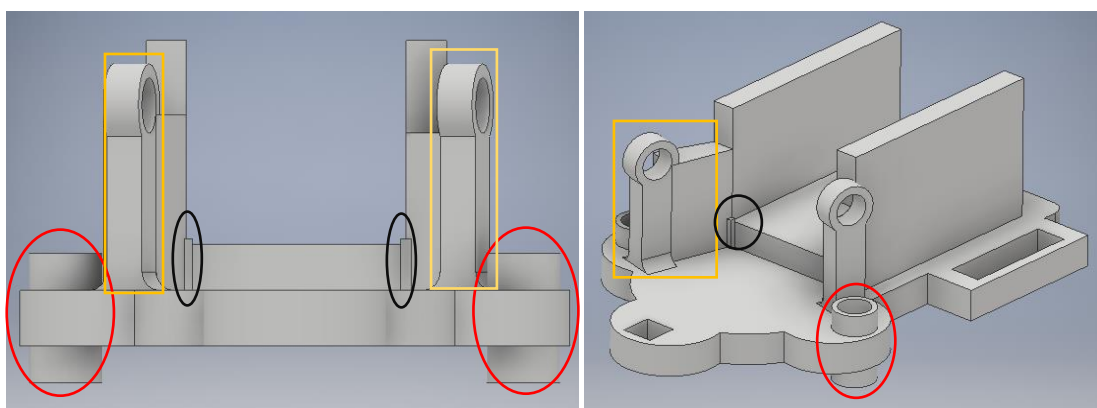
The very first prototype of this element (see Figure 2.3) consisted in a flat base to be placed on the first two phalanxes of the index finger. To accomplish the first function, two side supports with holes were designed to allow the introduction of two Velcro strips. This configuration has been maintained in all the following versions, because it has been very effective: the Velcro strips allow to anchor the device to the second



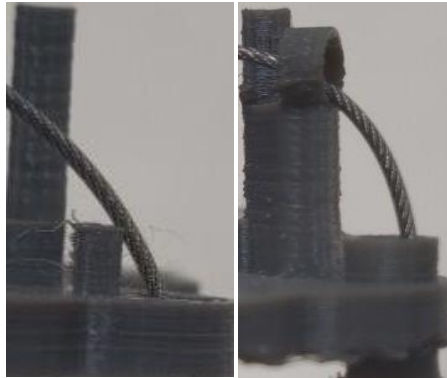
**Figure 2.3** The first prototype of the base element. The side supports for the Velcro strips (red circles), the three holes (black circles), the frontal relief (yellow circle) and the two parallel walls (blue circle) to block the motor position are highlighted.

phalanx of the index, without touching the first phalanx where the user perceives the tactile feedback. Moreover, this solution allows the thimble to be adjustable to different-sized fingers, since it is possible to close the strips at different levels. In this way, it meets also one of the requirements listed for a thimble-like device. Regarding the stability of the motor, this first prototype is able to close it between two parallel 11,5 mm high walls placed at a distance of 12 mm and avoid movements towards the front thanks to a 2.5mm high relief. The three holes visible in the image represent the only way for the cables to pass: since only one motor is used, the two 3mm diameter lateral holes let pass the cables responsible for the movement of the platform, while the third, placed in the front, allows the movement of a cable used as a guide (check Figure 2.3). Little attention in this prototype has been dedicated to the cables: two small pieces placed at the internal side of the holes try to guide the movement of the cables. The problem encountered with this prototype is related to the rotation of the connector of the gearmotor and the reel element of the device that is hooked on it and that therefore affects the pulling or releasing of the cables.

The solution to this problem comes together with significant adjustments after different trials in the device that can be identified as the “second prototype” (check Figure 2.4). In this version, the problem of the rotation of the gearmotor and the reel element is solved by raising the base where the engine is placed. In addition, the previous relief used to block the forward movements of the engine is replaced by two



**Figure 2.4 The second prototype of the base. The small parts in the lower corners for blocking the motor (black circles), the structure for the cables (yellow rectangles) and the tube-like structure for cables and compression strings (red circles) are highlighted.**

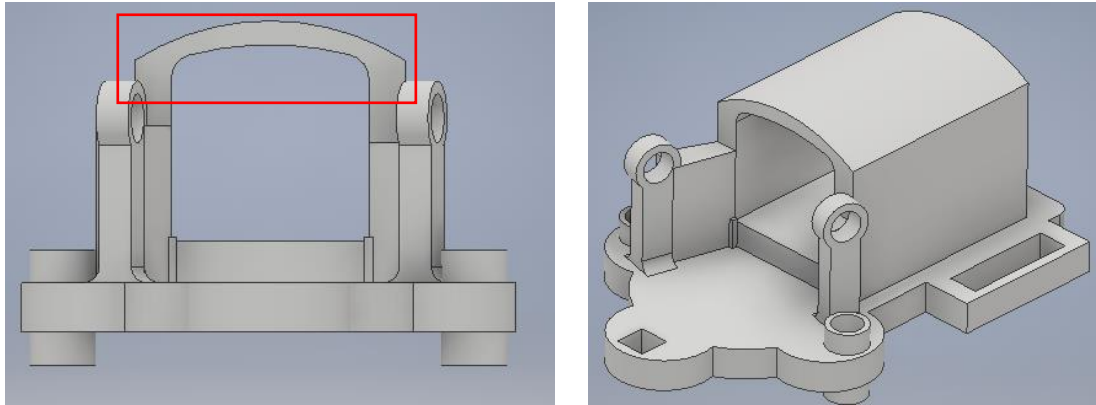


**Figure 2.5 The cables routing problem in the first prototype (left) and the solution on the second one (right).**

small parts located in the lower corners of the base. These parts avoid blocking the rotation of the connector from the bottom. In this version it is important to note that special attention was paid to the cables routing. In fact, it was noticed that in the previous prototype the cables were pulled or released diagonally. They were free to tilt in many uncontrolled ways, escaping from the guides and generating frictions with the base holes, thus preventing the correct operation of the motor. These frictions caused a non-fluid movement of the base as consequence. Two structures have been added to the side of the holes. Basically, they are two columns with two circle structures of 3mm diameter positioned perpendicular to the base (see Figure 2.5). The positioning of these structures is designed to align each circle to one cable, trying to interfere minimally with its movement. In previous versions of the prototype, these elements appeared too fragile at the base also because of their considerable height and small thickness. For this reason, wall-like supports have been added. They go from the parallel walls to the columns with the aim of reinforcing the structures (check Figure 2.4). This system has proved to be more efficient: the cables are guided and contained by the lateral holes and descend vertically to the holes on the base. To better guide them, two 2 mm long vertical tube-like structures have been superimposed to the holes on the base (see Figure 2.5). Another issue concerning the first prototype has been detected only after the assembly of the different elements. The springs located between the two parallel planes tend to move during the normal employment of the device, thus

causing the cables to bend and leading to unusual positioning of the parallel plate. This problem is solved by placing another pair of 2 mm long tube-like structure in the lower part of the piece just below the holes on the base where the cable connected to the motor pass (check Figure 2.4). The springs can wrap these elements in the terminal part in order to maintain a rigid vertical position. Since the third hole serves only as a guide and has no active parts, it is not needed to adopt the same solution (the spring will be attached to the parallel floating platform). Nevertheless, printing these elements implies substantial changes in the printing settings. In particular, it is necessary to impose the creation of a removable support base that would stand below the first layer of the whole piece. This allows to print the entire piece with the lower structures while maintaining a horizontal position. However, since the removable support is a plastic mesh, the lower surface of the element which rests on it is no longer smooth but very rough. The consequent need to sand off this part brings two advantages: the lower part is no longer so smooth as to be slippery, so it enhances grip. Since it is sanded off manually, it is possible to give the lower part an arc-like shape that can adapt to the upper part of the finger, improving ergonomics.

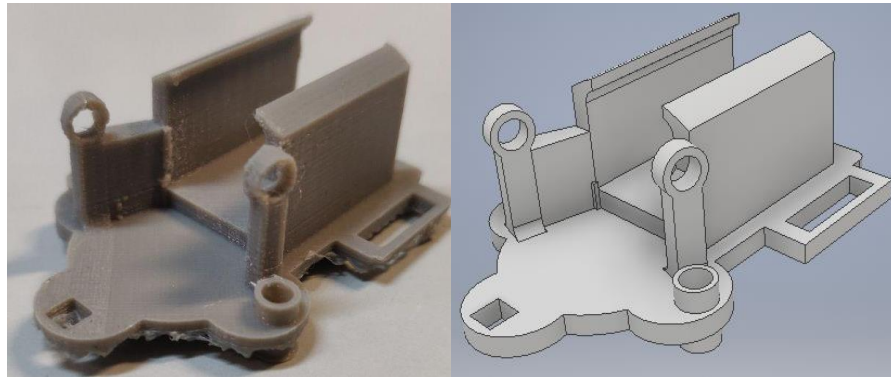
A fundamental problem that emerged during the testing of the device draws a clear line between the second and the third prototype of the base, designed precisely to solve this limitation. Although the motor is well embedded between the side walls and the elements in the lower corners block its movements forward, it has been noticed that it is not constrained in the vertical movements. In fact, when the motor is activated to pull the cables and raise the parallel platform to tighten the finger, it undergoes an inclination towards the front where the connector of the gearbox is connected to the cables through the reel. This causes damage to the plastic elements and implies abnormal behaviours of the device, that, in the long term, damage the same engine. In the third prototype the issue is fixed by adding a layer structure in the upper part to connect the two side walls and lock the motor in the four side surfaces (see Figure 2.6). The arc structure is thought to avoid some printing issues. A straight surface connecting the two side walls and closing a rectangle shape would ensure a complete



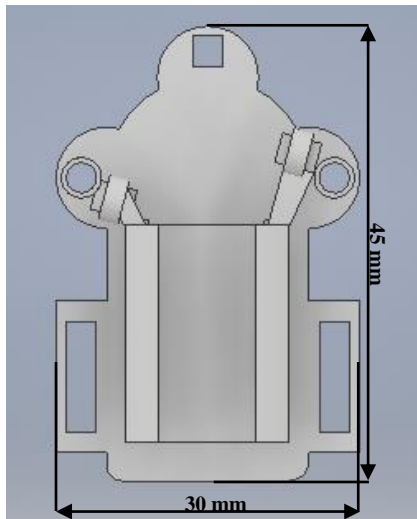
**Figure 2.6 The third prototype of the base. The structure that covers the motor in the upper part is visible, avoiding the inclination toward the front part (red rectangle).**

constraint of the motor, but it would need an underlying removable support. This support would be placed in the empty space where the motor is inserted, thus causing printing imperfections and roughness impeding the motor to fit in properly. For this reason, imposing a very low printing speed, the 3D printer is able to print an arc structure by just adding layers more and more central until the structure is closed.

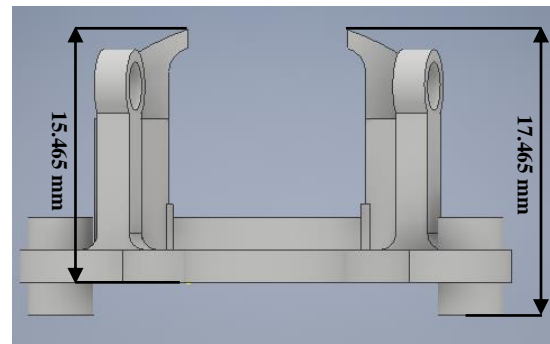
This prototype solves all the problem encountered in the previous versions but has some limitations. Even if it perfectly locks the motor and allows a good functioning of the device, it is quite heavy. This can affect the index finger movement. Moreover, it adds an important drawback to the entire platform: since the total size is bigger than the other versions, it influences and impedes the correct tracking of the hand movements performed by the infrared camera sensor. In addition it covers the motor for its entire length reaching the back part and interfering with the cable connection. The idea that there is behind the fourth and last prototype is to address this problem and to lighten the whole element structure so as not to interfere with the movements of the fingers. It was noticed that, in reality, the motor was locked thanks to the upper corners of the new closed configuration. Consequently, the solution was to leave the structure open and print just slight inward projections creating the upper corners to secure the engine. The open space on the upper part allows also the ventilation of the motor, avoiding eventual overheating due to a long use. Secondly, it allows a direct access to the motor to be able to intervene in case of problems and extract or insert it



**Figure 2.9** The final version of the base. Real piece (left) and CAD model.



**Figure 2.8** Top view of the base element.



**Figure 2.7** Front view of the base element.

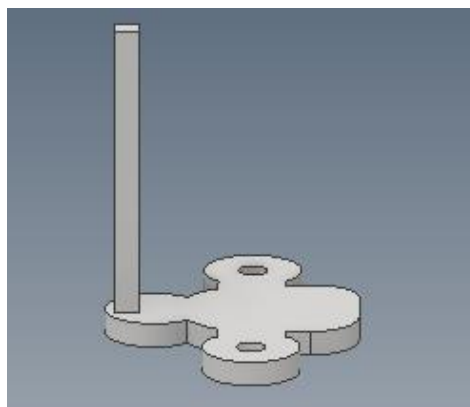
in an easy way. The final prototype used is shown in Figure 2.9 where the CAD image is placed side by side with a photo of the real piece to highlight the reached printing precision. The device is 45 mm long and 30 mm wide while the distance from the contact surface with the finger to the highest point of the device is of 15.465 mm (see Figure 2.7 and Figure 2.8). With the aim of lighten the device also in terms of weight, the thickness of both the walls and the base platform are reduced to 2mm. Thanks to these characteristics, the base element of the thimble represents a good balance between solidity and compactness. It is important to state again that all the characteristics have been designed in order to include a motor already present and

available in the laboratory. Nevertheless, it would be easy to integrate another type of motor just by changing the part of the element that contains it. This first base element is fundamental for its functions, so particular attention has been dedicated to its design to allow integration with the other pieces.

### ***The parallel platform.***

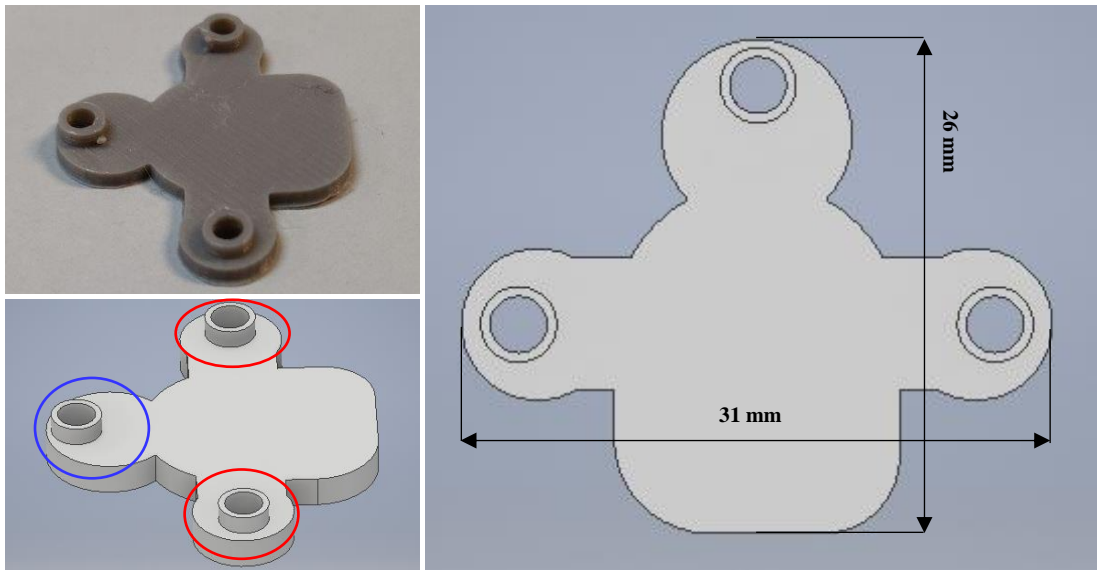
The second element of the thimble is the parallel platform that is placed below the user's fingertip. The function of this element is to tighten or loosen the grip on the fingertip with the purpose of activating the user's sensory receptors, thus providing the haptic feedback. To do so, the platform is controlled in its movement through the cable system connected to the engine.

As for the other parts of the thimble, several prototypes have been evaluated before reaching the ideal solution that ensures a good functioning. The main idea is to connect the platform to the reel element through cables hooked onto two side holes in the platform. This solution has been maintained in all versions of the element, while the changes concerned the part relating to the passive guide positioned in the front part. The first prototype, for example, included a vertical column printed in 3D that must fit vertically into the guide hole of the base element placed above the finger (see Figure 2.10). Obviously, this solution has many disadvantages because it forces the platform to move vertically without the possibility of slightly tilting when it meets the obstacle



**Figure 2.10 The first prototype of the platform. The vertical column guide is visible.**

of the finger. Moreover, it breaks easily due to the long length of the column and its small section. The best solution found with the second prototype was to emulate the connection of the towing cables and so connect the guide holes with a cable and a spring, allowing the alignment of the two parts but giving a certain freedom in the inclination. The final version of the mobile parallel platform is shown in Figure 2.11. It consists of a central flat structure that is in contact with the fingertip from which three arms extend outward. Each of these includes a hole that serves to pass the cables connected to the first element. In particular, the two side arms (Figure 2.11, red circles) are designed for the cables connected to the motor, while the arm in the front (Figure 2.11, blue circle) is connected to the passive cable that acts as a guide. The last prototype differs from the previous ones because it is reduced in size (31mm long and 43mm wide) and thickness (2.5 mm) so as to lighten the weight and improve the hand-tracking of the camera sensor. Another difference is that it includes three 1.5mm long tube structures positioned on the holes that are used to hook the springs.

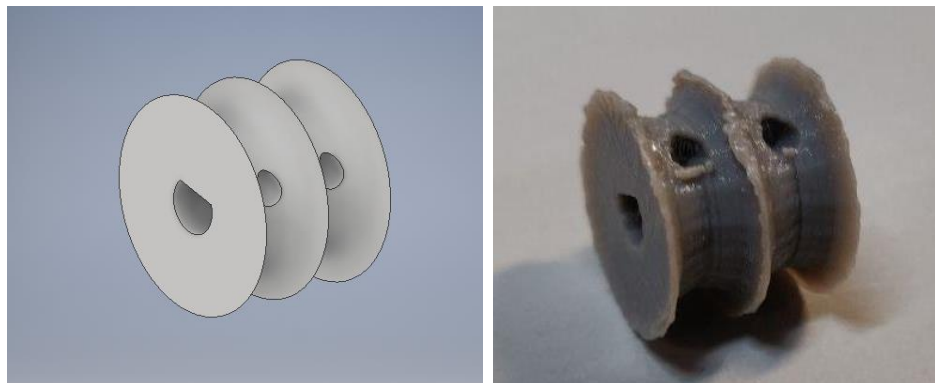


**Figure 2.11** The final prototype of the platform. Real object and CAD model (left) are shown together with the measures of the element (right). The side arms (red circles) and the front arm (blue circle) are highlighted.



### ***The reel.***

The third element is a structure that is mounted on the connector of the gearmotor. Its function is to hook the cables to the connector in such a way as to transform the rotation of the motor into a movement of the mobile platform by tightening or releasing the cables. In the first prototype versions of this reel element the problem faced was hooking the cables to the connector. The solution was to print in 3D a cylindrical structure with a hole in the center characterized by the same shape and almost the same size (3mm diameter) of the gearmotor connector. In addition, two small transverse holes (1.8mm diameter) were designed to allow the fixing of cables. The disadvantage encountered with this initial prototype is that the cables were wrapped directly around the connector, preventing it from working properly as they stuck together. The final prototype fixes this problem thanks to its structure: the entire length of the element is divided into two coils around which the cables can be wound without getting stuck (see Figure 2.12). Since the radius of the circumference around which the cables are wound measures 4.5mm, with a complete round of the connector this piece can roll up a length of the cable equal to the length of the circumference, i.e. about 28.27 mm. In this way this element acts as a transmission, allowing to pull a cable length about 3 times greater than the connector alone. The transmission ratio is given by the ratio of diameters (9mm circumference of the printed element around which the cables are wound, 3mm motor connector). The dimensions of the piece are highlighted in Figure 2.13, where it is also possible to appreciate the D-shape hole used to connect the element to the gearmotor connector.



**Figure 2.12 Final version of the reel. CAD model and real piece are shown.**

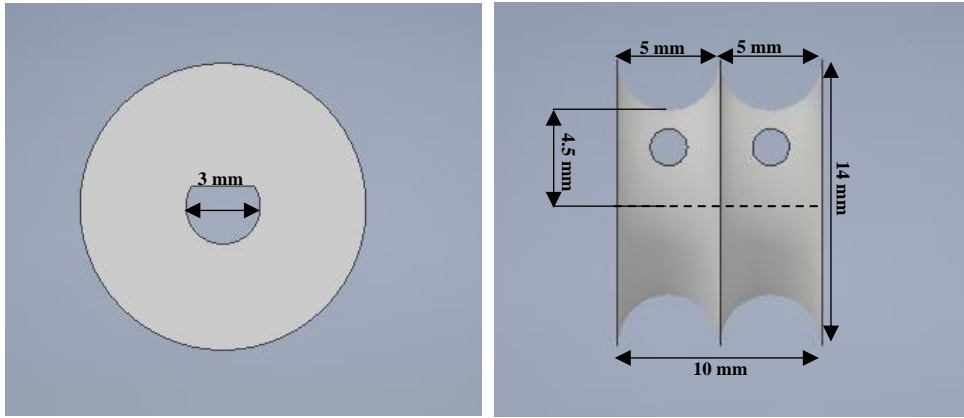


Figure 2.13 Dimensions of the reel.

### 2.1.2 Motor Description

It is essential to introduce the description of the motor included in this work before understanding how all the parts of the thimble are assembled. The motor used is the 298:1 Micro Metal Gearmotor HPCB 12V with extended motor shaft produced by Pololu (Las Vegas, USA) (see Figure 2.15). This gearmotor is a miniature high-power, 12 V brushed DC motor with long-life carbon brushes and a 297.92:1 metal gearbox. It has a cross section of  $10 \times 12 \text{ mm}^2$ , and the D-shaped gearbox output shaft is 9 mm long and 3 mm in diameter. This version has also a  $4.5 \times 1 \text{ mm}^2$  extended motor shaft. The gearbox (see Figure 2.14) is a mechanical device, part of the power transmission, which is used to reduce the speed of the output shaft compared to the input shaft. In the same time, it causes an increase of the output force, with respect to the input one. This motor performs a complete rotation of the output shaft (the one then connected to the mechanical parts, in particular to the third element of the thimble) for every 297.92 complete rotations of the input shaft. A high value of this reduction coefficient increases the precision with which the movement of the motor is provided.

In this work, different transmission coefficients have been tried (e.g 30:1 or 210:1) but the version 297.92:1 has been found as the most suitable for a fluid movement of the platform. Moreover, this reduction ratio exerts a force such as to be able to lift the



**Figure 2.14** The D-shaped connector attached to the metal gearbox (Pololu, 2019).



**Figure 2.15** 298:1 Micro Metal Gearmotor HPCB 12V with Extended Motor Shaft (Pololu, 2019).



**Figure 2.16** Magnetic Encoder mounted on the rear motor shaft (Pololu, 2019).

platform element of the thimble and tighten the fingertip succeeding in providing a considerable sensation of pressure.

The 4.5 mm-long rear shaft rotates at the same speed as the input to the gearbox and offers a way to add the encoder to provide motor speed or position feedback (see Figure 2.16). The encoder used is a magnetic encoder composed of a magnetic disc and hall effect sensors to provide 12 counts per revolution (CPR) of the input motor shaft when counting both edges of both channels. To compute the counts per revolution of the gearbox output shaft, it is necessary multiply the gear ratio by 12: the total is 3575.04 counts per revolution of the output shaft. This combination provides a resolution of nearly 0.1 degrees at output shaft. The encoder operates from 2.7 V to 18 V and provides digital outputs that can be connected directly to a microcontroller or other digital circuit (Pololu, 2019).

The D-shaped shaft connector (see Figure 2.14) is used to connect the motor and the third part of the thimble around which the cables are rolled up.

### 2.1.3 Mechanical Connections and Final Assembling

In this section all the mechanical connections and the materials used to realize them are illustrated. As already explained, the feedback haptic device is composed by three different elements mounted together with a small DC motor: the base element on the back of the finger has to support the motor that, thanks to the reel element mounted on the output motor shaft of the gearbox, has to move the parallel platform situated below the fingertip. The base and the platform are kept aligned and spaced thanks to the use of three compression springs. The functioning of the device is possible thanks to a system of metallic cables. Figure 2.17 shows the exploded model. All the components and connections can be appreciated in this schematization.

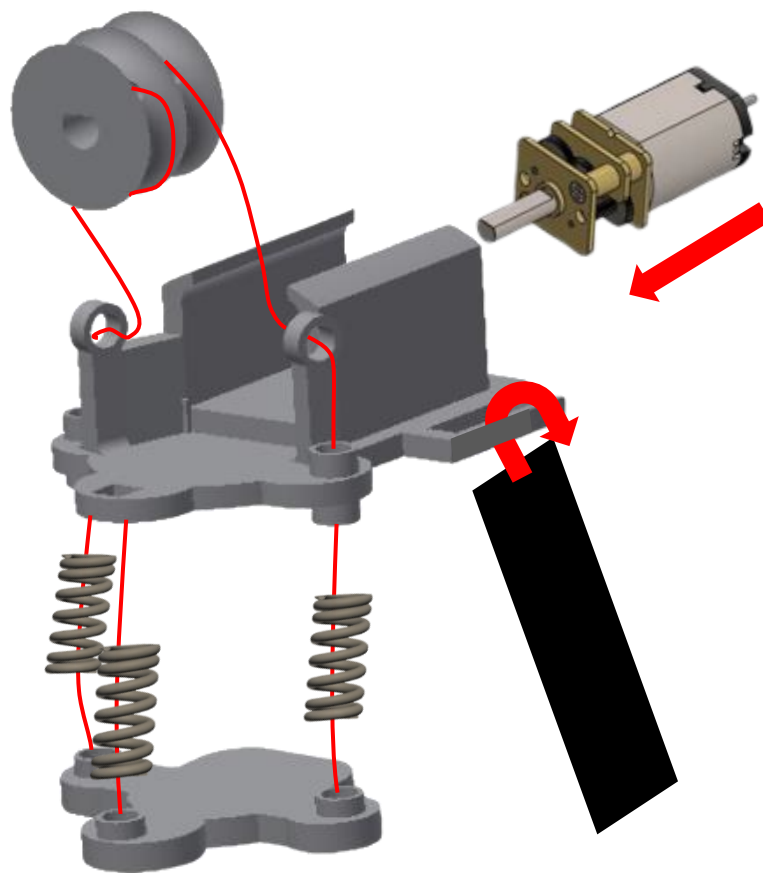
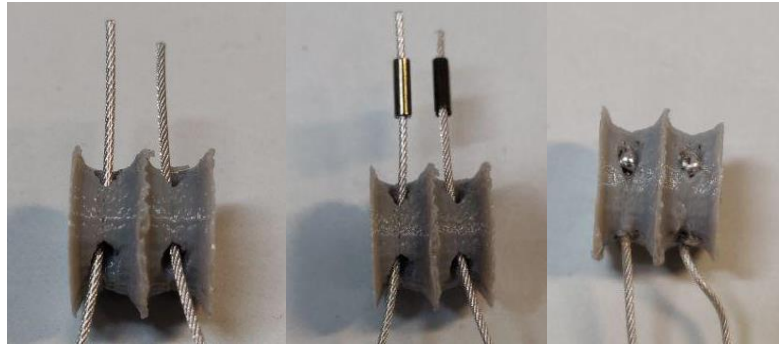


Figure 2.17 Exploded model of the thimble.

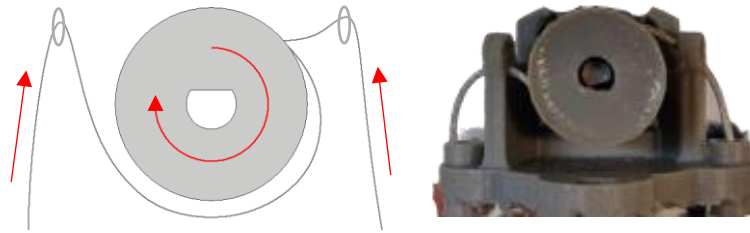


**Figure 2.18 Cable fixing. First the cables are inserted into the small holes in the third element (left), then two double copper ferrules are placed on the tips and tightened (centre). The cables are pulled into the holes and fixed with solder tin (right).**

The first essential connection requires to anchor the metallic cables to the drive shaft. In order to do this, the cables need to be inserted into the 1.8mm diameter holes in the coil element designed for this purpose. The cables are first placed inside the holes and then two copper ferrules are placed on the tips and tightened so as to create a thickness: in this way, when they are pulled back into the holes, they get stuck. Since this coupling is fundamental for the mechanical functioning of the device, a strong seal of the cables is ensured by welding both ends with solder tin to close the holes (see Figure 2.18).

At this point, the two platforms are connected using springs. The springs embrace the tube structures present on the holes of both platforms to ensure that they can remain aligned. The motor is then inserted in the specially designed box on the first element, showing the connector of the gearbox at the front and the rear shaft with the encoder mounted at the back. The 3D printed reel plastic element with the cables hooked is then placed on the output shaft. The cables are rolled up only once to ensure a slight pre-tensioning and arranged in such a way as to move in the same direction to both sides. This configuration, shown in Figure 2.19, allows to pull or release them both with the motor revolution.

Next, the cables are passed through the raised guide holes of the lateral structures and then into the base element towards the holes of the parallel platform by crossing the



**Figure 2.19 Cable configuration system. This solution allows to move the cables in the same direction.**

springs in the inner part. The solution adopted to stop the cables and prevent them from passing through the holes is to place nuts below the platform, pass the cables inside them and finally add double copper ferrules. In this way they cannot go back. Once the cables have been tensioned and calibrated, the copper ferrules are tightened and cut (to prevent the protrusions from affecting the fingertip recognition by the sensor) (see Figure 2.20). This system allows easy changes in case of need. The assembly of the device is then completed. The final version of the real device is shown in Figure 2.21.



**Figure 2.20 The solution adopted to stop the cables outside the platform.**



**Figure 2.21 Final model of the device.**

## 2.2 Electronics

This section deals with the electronic control of the device. The electronic hardware including the microcontroller, the extension board and the encoder of the motor will be shown, with special emphasis on the importance of their connections.

It is important to introduce the functionalities and objectives that the electronic control system must implement to better understand the selection made. The system implemented must be able to:

- tighten the user's fingertip in proportion to the data received from the game software that is responsible for recognizing the interactions;
- ensure a stable communication and avoid anomalous behaviour.

At the hardware level, a microcontroller and the encoder of the motor need to work together to ensure these functionalities.

It is essential to emphasize that all connection parts and hardware components must be solid and stable, since small variations can interfere with its normal operation. It is also important to consider that the thimble is designed as a “wearable device”, with the intention of helping the patient in his rehabilitation process even at home, preventing him to wear heavy tools. This must be a key point in the design phase, since the connections must be solid and easy to replicate and must not disturb the user during the execution of the training session.

### 2.2.1 Boards and Connections

The fundamental elements that compose the electronic system capable of managing the motor control are:

- Arduino Due, which incorporates the microcontroller Atmel SAM3X8E.
- X-NUCLEO-IHM04A1, dual brush DC motor driver based on the H-bridge L6206 expansion for STM32 Core.
- Motor magnetic encoder.
- Cables and connectors for communication.

The Arduino Due (see Figure 2.22) is the core of the electronic control system. It is a microcontroller board based on the Atmel SAM3X8E ARM Cortex-M3 CPU equipped



with 54 digital input/output pins (12 of them can generate PWM outputs) and 12 analog inputs. The Arduino Due board works at 3.3V and it has a flash memory of 512 kB and a clock speed of 84 MHz. There are two USB ports available on the board where one is programming USB port and other is native USB port (Arduino, 2019). Just one USB port is needed in this Thesis as a serial port for sending data and load the firmware onto the device. These characteristics make the Arduino Due a good electronic board for the management of large-scale projects that require several connections and that

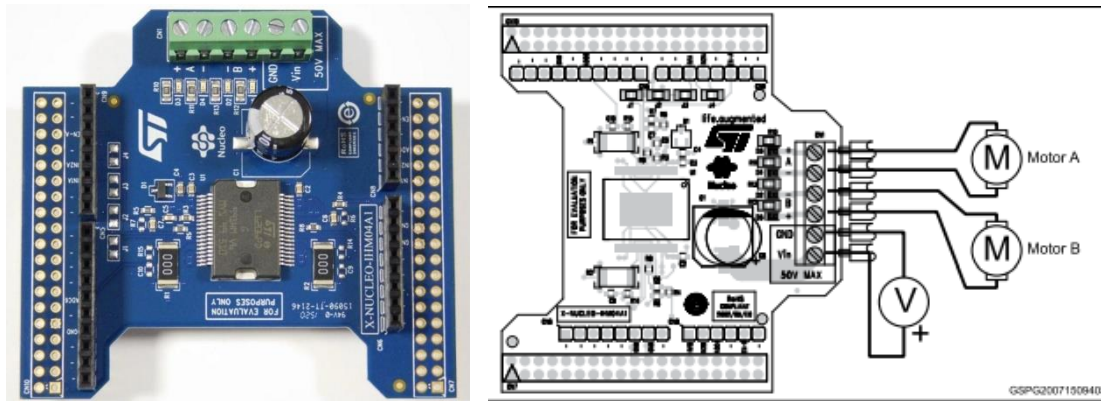


**Figure 2.22 Arduino Due Board (Arduino, 2019).**

need discrete computing power. In the case of this Thesis, the Arduino is required to control just one motor so many of its functionality and pins will remain unused. Nevertheless, it is an excellent system for this first prototyping phase, since is not required a detailed level of control and it is possible to use the Arduino Software, the Integrated Development Environment (IDE) with the included libraries to implement the firmware. Further improvements are possible relating to which electronic board is better to use, but the Arduino Due is a good solution to achieve the goals set in this Thesis.

In addition to the Arduino board, it is necessary to use a power stage for the motor control. In this work, the X-NUCLEO-IHM04A1 board by STMicroelectronics is used (St, 2019) (see Figure 2.23). This is a dual brush DC motor drive expansion board based on the L6206 (DMOS dual full bridge driver) that can be used to drive dual bipolar DC or quad unipolar DC motors. It allows to control the direction and the





**Figure 2.23 Power Stage, X-NUCLEO-IHM04A1, dual brush DC motor drivers based on L6206 expansion for STM32 Core (left) and its schematic representation in the configuration that allows connections with two motors and with the power supply system (St, 2019).**

velocity of the motor through an H-bridge circuit. The board is equipped with a power supply connector through which it is powered and attached to the electric current with a maximum voltage of 50 volts. It has four motor phases connectors, two for each motor that allow the activation of the motor in two possible direction thanks to an H-bridge system (St, 2019).

The H-bridge is a circuit that allows to apply the load voltages in two directions and therefore to make the current flow in the two possible ways so that the motor can turn clockwise and anticlockwise. Depending on the level of voltage and current set, the motor can run at different powers and turn at different velocities.

The PWM (pulse width modulation) technique is used to control the transistor of the H-bridge. It is based on the modulation of the duty cycle of a square wave that allows to obtain real intermediate values between the digital values of 0 and 1 and therefore to adjust the motor power.

The X-NUCLEO-IHM04A1 board is compatible with the Arduino board and can be mounted directly on top of it. Some pins only act as intermediaries to the Arduino pins whereas others are used to manage the power stage board itself. Therefore, particular attention is required for the connection between the power stage and the Arduino Due board, in order to properly control the motor (see Table2.1). Once the two boards have been connected, it is essential to create the connection with the motor.

Pin Power Stage	Function	Pin Arduino
Enable H-bridge Motor A	Activation control motor A	Digital pin (2), fixed high
Input1 MotorA	PWM Control motor A, direction 1	Digital Pin (34), generating PWM signal
Input2 MotorA	PWM Control motor A, direction 2	Digital Pin (37), generating PWM signal
GND	Ground	Pin GND

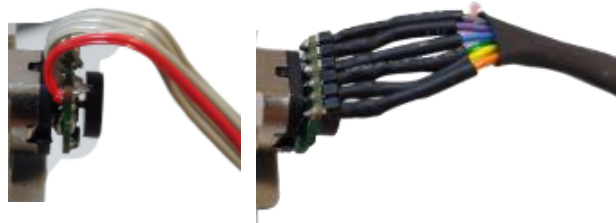
**Table 2.1 Connection between the power stage and the Arduino Due.**

The encoder board is designed to be soldered directly to the back of the motor, with the back shaft of the motor protruding through the hole in the middle of the circuit board. It has six fundamental pins to which six cables are connected (see Figure 2.24). Two cables (grey and purple) are used for the bipolar control of the motor. They are connected to the motor phases connector of the power stage, and according to the current flow direction, the motor runs in the two possible ways with different velocities. Two other cables connected to Arduino are fundamental to power the encoder magnetic sensors between power supply (blue) and ground (orange). The remaining two pads are used to access the two quadrature outputs sent to Arduino with two cables (green and yellow). The two outputs are square waves from 0 V to Vcc. The sensors' comparators have built-in hysteresis, which prevents spurious signals in cases where the motor stops near a transition point. The connections are listed in Table 2.2.

Pin Motor Encoder	Cable Color	Boards Connections
Motor power, positive pole	Gray	Motor connection +, power stage
Motor power, negative pole	Purple	Motor connection -, power stage
GND encoder	Orange	GND Pin, Arduino
Vcc encoder	Blue	5V Pin, Arduino
Encoder output A	Green	Digital pin (7), Arduino
Encoder output B	Yellow	Digital pin (3), Arduino

**Table 2.2 Connection between the motor and the electronic system.**

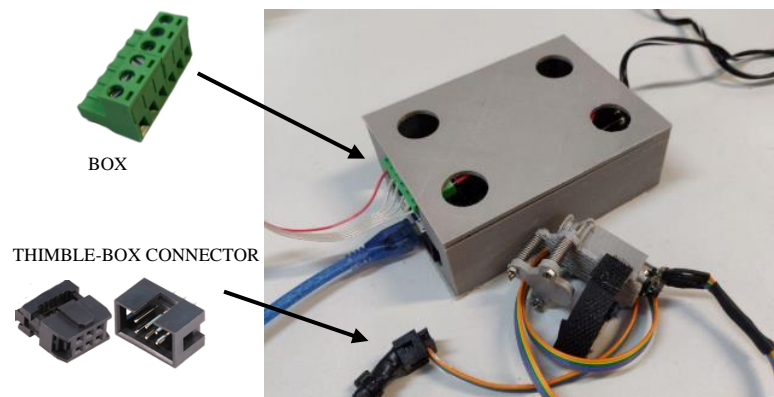
As mentioned before, all connections must be strong enough to withstand several movements. This is particularly true regarding the thimble, since once worn it is moved by the user during the exercises. This means that the cable connection must be stable to ensure proper operation of the device, otherwise abnormal behaviour will occur and could hurt the user. Different configurations have been tested to prevent the welding of the cables from failing. The first solution was to solder the cables directly on the board (see Figure 2.24, left) but since both the cables and holes are small, it was easy for them to detach. The alternative found is to solder a 2mm-pitch connector to the board and in turn solder the cables to it (see Figure 2.24, right). Heat-shrink tubing is added on each connection in order to make it more solid.



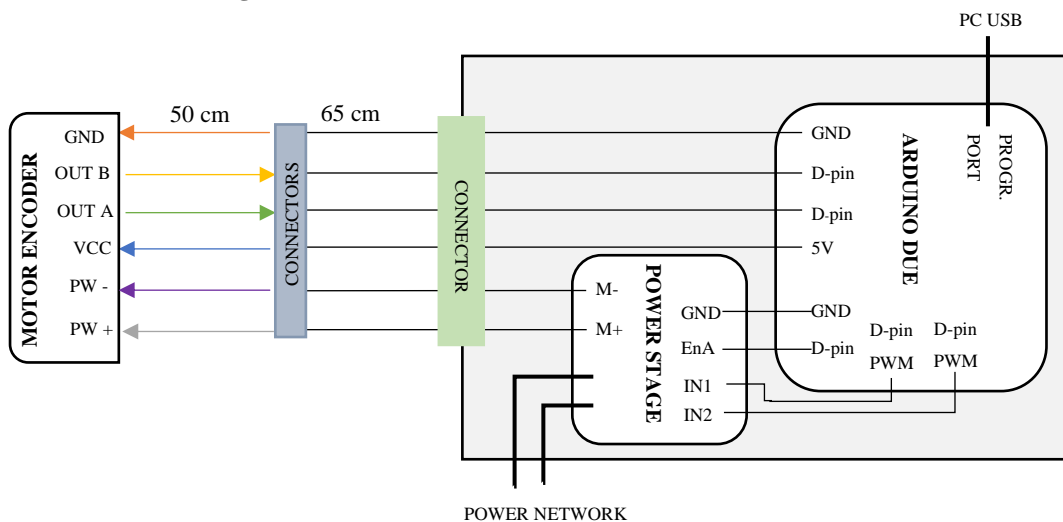
**Figure 2.24 First welding configuration (left) and last stronger configuration (right).**

The entire system is designed to be portable and wearable. However, the connection between the Arduino Due and the power plate makes the electronic part of the control very cumbersome. Consequently, the idea of wearing this element on the arm as a bracelet was discarded also because the connections would have been fragile. In order to give rigidity to the entire system and to ensure a full capacity of movement during the exercises, it was thought to keep the electronics away from the user. For this reason, a case has been printed in 3D to contain both boards and connections (see Figure 2.25). The box is easily transportable and allows to place the electronics close to the computer. Moreover, it makes easier to connect the power board to the power grid and the Arduino to the computer via serial port. On the side of the case there is a connector that is used to insert all the cables connecting the boards to the thimble. This is attached to the housing cover so as to avoid the movement of the internal connections and withstand the movement of the connecting cable of the thimble. A 65cm extension

cable is attached to the connector and ends with a male connector. This configuration allows the thimble's cables to be easily connected and disconnected: they are cut to a length of 50cm and are housed in a female connector. The system of connections and extensions guarantees a big mobility since the user only has to wear the thimble (the total length of the cable is 115 cm) and fix the cable to the wrist so as to avoid interference with the recognition of hand movements. In addition, this system ensures easy interchangeability of elements. The entire connection system including all the boards, connectors and cables is schematized in Figure 2.26.



**Figure 2.25** Final outcome of electronics and connections.



**Figure 2.26** Electronic components and connections.

## 2.3 Firmware

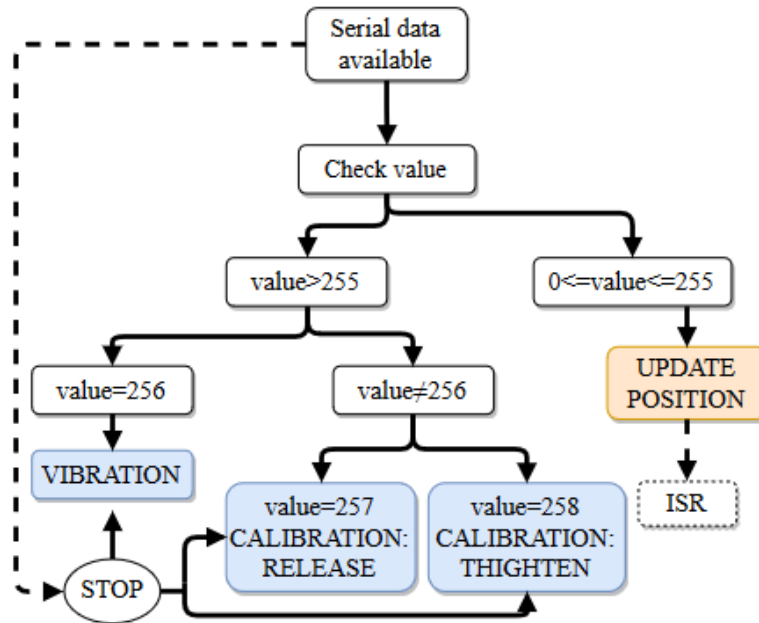
This section describes the firmware developed for the control of the feedback device. The firmware is the code that allows the communication and the data transfer between the hardware elements of the system, so between the computer, the microcontroller and the thimble device. The software used in this Thesis to develop the firmware is the Arduino IDE. It is suitable for the level of this project since it is an intuitive and easy-to-use platform for a high-level programming. The firmware developed is loaded on the board and takes care of managing all the functions that the microcontroller must perform.

Before the description of the code, it is important to remember its objectives. The Arduino code must receive data from the computer and then manage it in order to:

- Generate a vibration of the motor.
- Make the motor rotate clockwise or anticlockwise to pull or release the cables with respect to the input from the games, using a developed PD controller.
- Make the motor rotate slowly in both directions to allow the user performing a manual “calibration” of the device.

An intuitive overview of the general functioning is given in Figure 2.27. It explains the code from the moment that a data arrives from the serial port to the microcontroller. The incoming value is checked. If it is comprised between 0 and 255, it activates the position controller. If the value is higher than 255, then allows the vibration function (value=256) or the calibration of the motor. In this case if the value is equal to 257, it enables the releasing of the cables, while if it is equal to 258, it allows to tighten the cables. The vibration and calibration functions can continue until a new serial data arrives.

The operations are managed using different functions in combination. Some of them are controlled using an interrupt handler, also known as an interrupt service routine (ISR). It is a call back subroutine that intervenes in an operating system and its execution is triggered by the reception of an interrupt. Interrupts can be occasional if due to the occurrence of a hardware event (change of state of a pin) or regular if linked



**Figure 2.27 Essential functioning of the code.**

to a sampling period. In the last case, interrupt service routines are useful for carrying out certain actions automatically in microcontroller programs. Furthermore, when working with several ISRs, every new ISR does not start before the end of the previous one. This is useful specially to accomplish fast computation with small functions.

For an easier management of timers for ISRs, the “*DueTimer.h*” library has been used, which is a fully implemented timer library for Arduino Due by Ivan Seidel (Seidel, 2018). In case of the Arduino Due, the SAM3X8E CPU has 3 timer counters (TC). Every timer counter contains 3 channels. Every channel has its own counters and an interrupt handler that are independent of other channels. In other words, each channel can be considered as a separate “timer” and is like having 9 separate timers.

The general idea of the code is to manage the main functionalities using both the cyclic iterations of the microcontroller and the ISR. In this way, it is easier to have a control of the system timing.

The first part of the code includes the libraries used. In this work only two libraries have been used, namely the “*DueTimer.h*”, timer library already mentioned above, and the “*Arduino.h*” library that contains all the pre-programmed functions available to

manage Arduino. Next, all the global variables useful within the code are initialized. These will be used, for example, to save the input values to the microcontroller and to carry out the following operations.

The Arduino programs are basically composed of two main functions:

- *Void setup()* is a function invoked only once at the beginning of a program, to be used for the initial settings that will remain unchanged during the execution.
- *Void loop()* is a function invoked repeatedly, whose execution is interrupted only when the power to the card is removed or when an ISR is activated.

In both functions it is possible to call other functions to achieve different task so as to divide the code into modules that can be better understood.

In the developed firmware, the *setup()* function is used to open the serial communication with computer at a speed of 115200 bauds and to activate the H-bridge for the motor control. This is done by fixing high the Arduino pin connected to the enable pin of the power board. Here it is also established that the triggers of two ISR are related to the changes of state of the two Arduino pins that receive data from the motor encoder. The *encoder()* function (explained below) is called every time these ISR are triggered. Other two interrupts are initialized and connected to timers: every 400ms the first ISR is called and executes the function *controllerPD()*, while every 70000ms the other ISR execute *componentD()* function (both are explained in what follows) (see Figure 2.28).

```
// interrupts to get the value registered by the encoders
attachInterrupt(digitalPinToInterrupt(pinEncoder1), encoder, CHANGE);
attachInterrupt(digitalPinToInterrupt(pinEncoder2), encoder, CHANGE);
// timers for the PD controller
Timer4.attachInterrupt(controllerPD).start(400);
Timer3.attachInterrupt(componentD).start(70000);
```

**Figure 2.28 Activation of the interrupts.**

The last task accomplished in the *setup()* function is to configure the PWM at a hardware level by calling the *configure\_PWM\_HW()* function. This function sets the bits needed to configure the various options of the PWM.

The most important part of the firmware is the *loop()* function, because it is invoked repeatedly thus allowing the platform to achieve the three objectives of the motor control. The first thing to be done is to save the value received from the serial port in order to analyze it. At this point it is fundamental to explain how the analysed data is constructed. The original idea was to check values between 0 and 255 since this is the maximum value representable by an eight-digit binary number, and therefore the maximum representable by an unsigned 8-bit byte (which is the smallest common variable size used in high level programming languages). Later, it was verified that the exchanged data in the communication between the game software and Arduino were encoded in ascii. The solution was therefore to ensure that the numbers sent by the pc were always three digits, so as to fill a char array of three elements. Only when the array was filled, the complete data obtained was transformed into float (see Figure 2.29) and the analysis can begin.

```
if (Serial.available() > 0) {  
    data[data_index] = Serial.read();  
    data_index++;  
  
    if (data_index >= 3 )  
    {  
        float input = atof(data);  
        data_index = 0;  
    }  
}
```

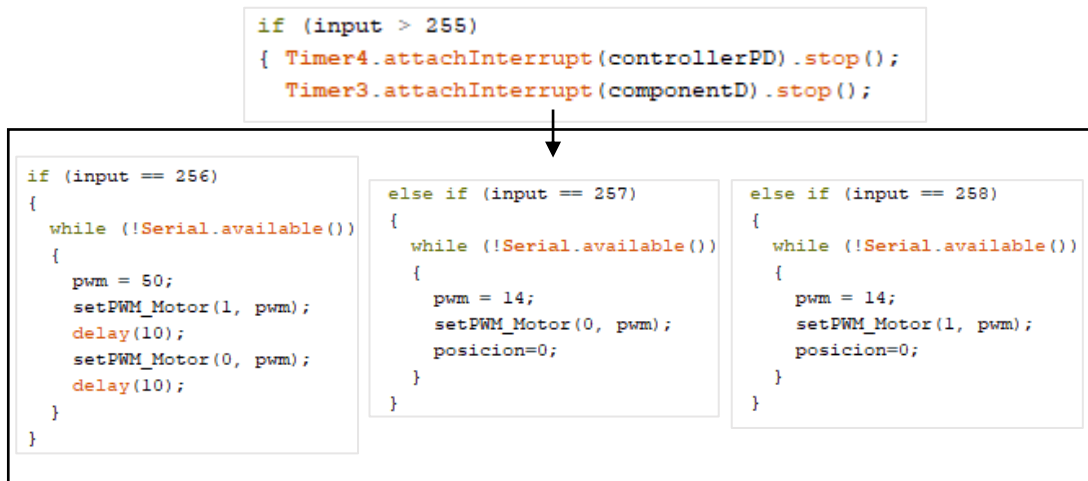
**Figure 2.29 Construction of serial data.**

At this point two options are opened: if the value is greater than 255 then we proceed with either vibration (value=256) or calibration (value=257 or value =258), while if it is smaller, we proceed with the PD control (see Figure 2.27).

In the first case, the timers relative to the ISR that manages the PD control are stopped and one of the three options continues to be executed until a new serial data arrives. For the vibration a PWM value of 50% is imposed and a change in direction every 10 ms. For the calibration, a PWM value of 14% is imposed in one direction or the other and, at every iteration position, values of “0” are imposed. In this way, if the calibration option is chosen, then the motor would rotate really slowly releasing or tightening the



cables and when the desired length is reached, the new configuration represents the starting position. To impose the direction and the value of the PWM is necessary to use the implemented function *setPWM\_Motor()* that sets and manage the duty cycle. The *loop()* function lines of code for these options are shown in Figure 2.30. The first “if” statement is entered if the received value is higher than 255. In this case the ISR timers of PD controller are stopped. The value is controlled again with if/else statements to access the other options inside this block of code (256 vibration, 257 releasing cables, 258 tightening cables). In each option the lines are closed in a *while()* statement that enable to iterate the lines until a new serial data is received.



**Figure 2.30 Options for values > 255. After stopping the timers (upper figure), vibration(left), releasing cable (center) or tightening cable (right) options are eligible.**

In the second case, if the value in input is lower than 255, the motor is controlled in position thanks to a PD control. In order to use the ISRs that call the functions related to this control, the timers are reactivated. In the *loop()* function the data regarding the

```

else {
  // timers for PD controller
  Timer4.attachInterrupt(controllerPD).start(400);
  Timer3.attachInterrupt(componentD).start(70000);
  posiz_unity = (int) ((input / 255) * 100);
}

```

**Figure 2.31 PD control activation in *loop()* function.**

imposed position received in this case are normalised between 0 and 100 and transformed into integers (see Figure 2.31).

As it can be noticed, others small functions different from these two main blocks are continuously used and called in different parts of the code. Before introducing the one that actually handles the control, it is important to highlight the objective of one of them: the *encoder()* function. These lines of code are activated thanks to an ISR triggered by the change of state of the pins connected to the encoder sensors. In this function the position of the motor shaft is measured and consequently saved in a variable that is constantly updated. This is possible using bit masks compared to the sensors signals in order to read the value and understand the direction of motion. The position variable increases its value if the motor is turning in one direction or decreases it if the motor is turning in the opposite direction.

As aforementioned, the control is managed with two functions. The most important one is the *controllerPD()*. This function accesses both position variables: the one measured using the *encoder()* function, and the one received from the computer (i.e. the one that should be set). Both variables are first normalized between 0 and 1 and

```
void controllerPD(void) {
    y_k = (float)(position) * TWO_PI / counts_per_revolution;
    r = posiz_unity * constant * 2 * PI / 100;
    error_k = r - y_k;

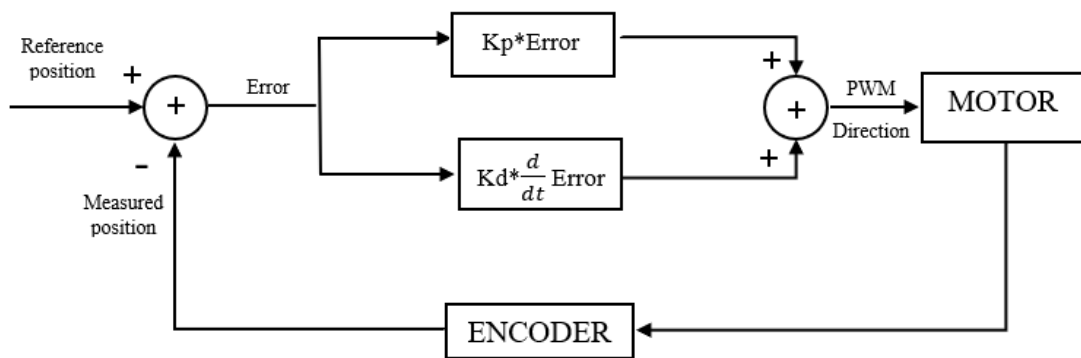
    // control signal
    u = error_k * Kp + Kd * d_error;

    // control signal transformed in pwm value
    if (abs(u) >= max_v) {
        pwm = 100;
    } else {
        pwm = (abs(u) * 100 / max_v);
    }
    if (pwm < 0) pwm = 0; if (pwm > 100) pwm = 100;

    // change direction with the sign of the error
    if (error_k > 0) setPWM_Motor(1, pwm);
    else setPWM_Motor(0, pwm);
}
```

**Figure 2.32 PD control function.**

then transformed in radians. Moreover, the position received by computer is multiplied by an empiric constant to adjust the entity of movement of the motor. At this point, the error between the desired position and the measured position is calculated. Meanwhile, the other ISR calls the *componentD()* function which estimates the error derivative by subtracting between errors calculated at different times. In *controllerPD()* the error is weighted by Kp and Kd constants. This value is saved and, once normalized for the maximum voltage of the motor, it is used to set the PWM value for the motor. The direction of the motion, instead, is set by looking at the sign of the proportional error (see Figure 2.32). Figure 2.33 shows the block diagram of the PD controller to better illustrate its functioning.



**Figure 2.33 Block diagram of the PD controller.**

## 2.4 Evaluation of the Device

After the explanation of the mechanical parts, the assembly of the entire thimble, the connections and the firmware that controls it, it is necessary to make an evaluation of the device. The most relevant aspects are related to the behaviour of the thimble in response to the values sent by the computer and therefore by the games. This is useful to understand how the thimble will then respond during the game session based on the movements of the user. To carry out this evaluation, the distance between the base and the parallel platform after sending the serial data has been measured. By blocking the position of the base, it was possible to make measurements with a digital caliper with a sensitivity of 0.01 mm.

For each measurement the thimble was connected to Arduino. The same firmware implemented for the platform was used to ensure a consistent evaluation. The measurement procedure consists of sending data to the serial port and measuring the distance between the base and the platform, following the movement of the motor. Values from 0 to 240 in steps of 20 been sent. The last value sent is 255, being the higher value for the position controller.

For this evaluation, data from 30 measurements were collected and the arithmetic mean and standard deviation of the distance between the base and the parallel platform in millimeters are shown in Table 2.3.

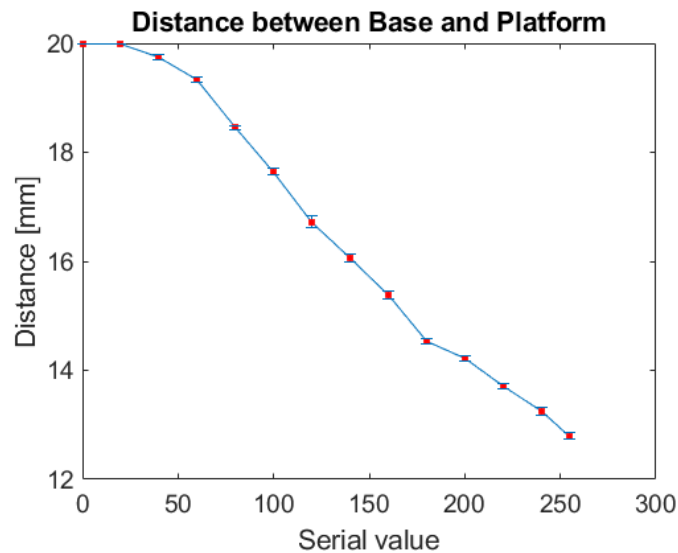
<b>Val</b>	<b>0</b>	<b>20</b>	<b>40</b>	<b>60</b>	<b>80</b>	<b>100</b>	<b>120</b>
<b>Mean[mm]</b>	20	20	19.74	19.34	18.45	17.65	16.73
<b>S.D.[mm]</b>	0	0	0.04	0.05	0.04	0.06	0.10
<b>Val</b>	<b>140</b>	<b>160</b>	<b>180</b>	<b>200</b>	<b>220</b>	<b>240</b>	<b>255</b>
<b>Mean[mm]</b>	16.06	15.38	14.53	14.21	13.70	13.25	12.80
<b>S.D.[mm]</b>	0.07	0.07	0.05	0.05	0.04	0.06	0.06

**Table 2.3 Arithmetic mean and standard deviation of the distance between the base and the parallel platform in correspondence of the value sent.**

As observed in the Table, the thimble goes from the maximum distance of 20mm corresponding to the minimum value sent (0) up to a minimum distance of 12.80mm, corresponding to the maximum value sent (255). These distance values have been

established in the design phase of the thimble. A maximum distance of 20 mm allows the user with an average finger size to wear the thimble without having the parallel platform in contact with the tip of the finger, thus avoiding receiving incorrect feedback. The minimum distance ensures that the thimble does not tighten the tip of the fingertip too tightly, causing damage. In any case, it should be remembered that thanks to the virtual calibration process these quantities can be moved simultaneously, increasing or decreasing them.

The collected data are represented in Figure 2.34. As it can be observed, the thimble assumes an almost linear behaviour. The initial stall is due to the beginning of the movement of the motor to which the springs are opposed.



**Figure 2.34 Distance between the base and the parallel platform in correspondence of the value sent.**

The force provided by the thimble on the index fingertip was not measured. However, the maximum torque provided by the motor reported by the manufacturer is 3.3 kg·cm, or 0.32361945 N·m. The force provided by the thimble will depend of the user finger characteristics.



## Chapter 3: Virtual Platform

This chapter illustrates the development and implementation of the virtual platform. The virtual reality platform allows a complete engagement of the patient in the rehabilitation session, thanks to the integration of the haptic feedback. It has been studied that the use of virtual games can complement the traditional rehabilitation. In fact, the latter presents a set of exercises that can result boring. On the contrary the virtual reality offers the ability to create any desired rehabilitation environment characterized by different features.

The purpose of this Chapter is to explain three different games for rehabilitation purposes. These games will allow the user to interact directly with the movement of the hand and will ensure communication with the hardware to provide haptic feedback. First the technologies used to accomplish this objective will be illustrated. Next, the implementation of the games with their main characteristics will be explained.

### 3.1 Enabling Technologies

This section describes the commercial technologies used to implement the virtual platform. Basically, to succeed in the realization of the games and to allow the participation of the user through the use of hands, two technologies have been used: Unity software (Unity, 2019) and Leap Motion controller (LeapMotion, 2019). The combination of the two provides the possibility to implement an appealing virtual environment that can encourage the patient to continue his rehabilitation.

#### 3.1.1 Unity

Unity is a multi-platform game engine developed by Unity Technologies (Unity, 2019). It is primarily used to develop both 3D and 2D video games or other interactive contents, such as architectural visualizations, three-dimensional settings, shorts films and small videos. Unity is defined as "multi-platform" because its engine allows to create the game once and transfer it to different platforms including computers, consoles and mobile devices.

The Unity developing platform consists of a graphics engine, a very powerful physics engine and a live game preview. The latter allows to see real-time changes made to the game during programming operations.

In Unity, the games are composed of scenes that the user can edit. The main Unity editor window presented to the developer is made up of customisable tabbed windows. The Project Window contains all the scripts, 3D objects and assets available to use for the project. The Scene View is the window that allows to visually navigate and edit the scene. Every element in the scene is an object which has certain properties and behaviour. Each object has a hierarchical text representation in the Hierarchy Window, while their properties are shown in the Inspector Window that also allows to change them. The behaviour of an object is set by associating a script to it. The programming language used for creating scripts uses C# or JavaScript.

To create a game, the developer must first create the environment by placing a background image or 3D ambient. Then he must add the main objects of the game, conferring them all the important characteristics needed. In order to use the object and decide all the dynamic characteristics and interactions, it is fundamental to add a script to it. Subsequently, all the obstacles and decorations are added to make the game more appealing. Designing and building the game in pieces gives the developer the opportunity to continuously modify a scene or add objects. When all the scenes to be developed are finished, the game is built for different platforms.

Unity has a great potential since provides the opportunity to develop games for free, giving the opportunity to import several pre-built assets containing objects and scripts. Moreover, the work of creating the game is facilitated by a series of tools that enable the developer to integrate the programming part with the graphics needed, being compatible with softwares for 3D modeling.

In this Thesis, all these characteristics were fundamental to achieve the implementation of the three games.



### 3.1.2 Leap Motion Controller

The Leap Motion controller (see Figure 3.1) is a small USB peripheral device designed for hand tracking. It was developed by Leap Motion (LeapMotion, 2019) that, in May 2019, merged with UltraHaptics and came together as “UltraLeap” (Ultraleap, 2019). The heart of the device consists of two monochromatic IR cameras and three infrared LEDs. These LEDs generate pattern-less IR light with a wavelength of 850 nanometers, which is outside the visible light spectrum. The light is reflected by the objects and reaches the two cameras which generate almost 200 frames per second capturing these data. The wide-angle lenses that the device has, permits a large interaction space of eight cubic feet, with an inverted pyramid shape that is the intersection of the binocular cameras’ fields of view. The viewing range is of 60 cm above the device, and it is limited by LED light propagation through space. In fact, it becomes much harder to infer the hands position in 3D when they are situated beyond



**Figure 3.1 Leap Motion controller (LeapMotion, 2019).**

a certain distance from the device. The data received by the camera are read and treated with necessary resolution adjustments in the memory included in the device. At this point the obtained data are sent through an USB to the host computer. Here, the Leap Motion Service software processes the images using heavy mathematical tools. It first compensates for background objects and ambient environmental lighting, then reconstructs a 3D representation of what the device sees. At this point the tracking algorithms extract information relative to the positions of hand and fingers. After a

filtering process, the information is managed by the software and expressed as a series of frames, or snapshots, containing all of the tracking data. The client library organizes the data into an object-oriented API structure, manages frame history, and provides helper functions and classes in order to enable the developers to use the info in their own application (Colgan, 2014).

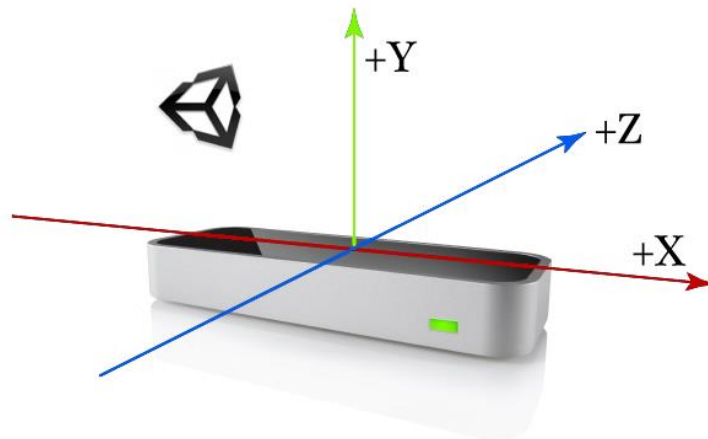
This device was chosen because of his small observation area and high resolution that make it suitable for the upper limbs and hands rehabilitation. Moreover, the controller can be used on a tabletop or mounted on a VR headset. However, considering that one of the most important aim of the project is to develop a less intrusive platform, the first configuration was chosen.

### **3.1.3 Elements Integration**

In this Thesis the Leap Motion controller device is integrated into the Unity thanks to a plugin. As already said, the Leap Motion Service software analyses images produced by the hardware and sends tracking information to the applications. The Leap Motion-Unity plugin connects to this service to get data. Scripts included with the plugin translate Leap Motion coordinates to the Unity coordinate system. These scripts and additional graphic assets make it easy to add 3D motion-controlled hands to a Unity scene. The version used in this work is the v2.3. All the documentation regarding this version is available in Leap Motion archive in the internet at the developers section<sup>1</sup>. Unity and Leap Motion Controller use a different coordinate system and units of measurements. A left-handed convention is used in Unity, whereas the Leap Motion API uses a right-handed convention. Unity also uses a default unit of meters, whereas the Leap Motion API uses millimeters. This can cause several difficulties when trying to control the device in Unity. For this reason, the plugin scripts internally transform the tracking data to use the left-handed coordinate system (see Figure 3.2). Moreover, there is the possibility to scale distance values to meters by using functions already available with the plugins in the Leap Motion C# library.

---

<sup>1</sup> <https://developer-archive.leapmotion.com/documentation/v2/unity/index.html>

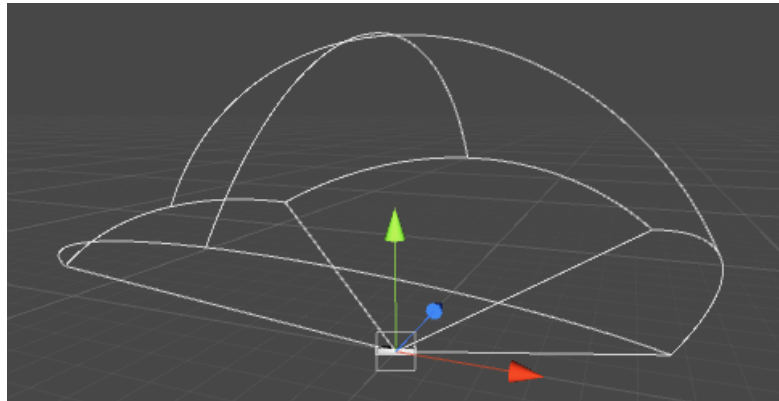


**Figure 3.2 The Unity left-handed coordinate system superimposed on the Leap Motion device in its desktop orientation (LeapMotion, 2019).**

Once the Leap Motion plugin is downloaded in “.unitypackage” format, it can be integrated in the unity project as an asset. At this point, in the Project window appears a folder which contains all the objects related to the Leap Motion device. Demos, example scenes, prefabs and usable scripts are included in this folder to allow the developer to include the device on his project.

The fundamental initial step followed to develop all the games of this work was to introduce the *HandController* prefab in the Hierarchy Window. The prefab serves as the anchor point for drawing hands. In this way, in the Scene View the controller device with the Unity coordinate system is visible. As shown in Figure 3.3, white lines are drawn in order to provide the developer an idea of the working area, corresponding to the approximate field of view of the Leap Motion device.

The *HandController* is seen in the scene as an object, thus having its own characteristics. Changing these properties allows to mount different visual model of the hand included in the Unity plugin. Moreover, from the Inspector Window it is possible to change other parameters such as the physical models or the scale of their representation in the scene. After placing it, the developer can build the entire game always having the reference of the position of the device and therefore of the hands.



**Figure 3.3 Representation of the device and the working space in the Unity Scene View.**

## 3.2 Games

Incorporating a virtual environment in the rehabilitation process offers the opportunity to address different therapeutic objectives. For this reason, in order to provide flexibility and elasticity to the rehabilitation session, three serious games with different characteristics have been implemented. The haptic feedback implementation for each game, as well as their specific therapeutic objectives will be shown in this Section.

Developing games for rehabilitation purposes is very different from doing it with the sole purpose of entertaining. Multiple variables related not only to the physical condition, but also to the emotional and psychological state of the patients must be considered. If the game is too difficult or too complicated, the patient would be discouraged from using the platform, obtaining the opposite effect to the intended one. For these reasons, game design specifications should be considered.

First of all, the design of the games must ensure that the games characteristics do not compromise the rehabilitation effort. For example, adding over-decorated environments may distract the patient from the purpose of the game.

Moreover, the exercises conceived for rehabilitation must be necessarily simple and must have clear goals and limitations. Too many tasks to complete or too many

executable actions can unnecessarily complicate the game, while a few specific actions allow the patient to work on specific skills and refine them.

The developed platform must provide games with tasks easy to repeat in order to ensure the performance of correct and effective movements to bring benefits to the patients. Easy and repeatable activities can be iterated over time and this is fundamental to promote plasticity recovery. Performing the same actions triggers the reorganization of the nervous system that allows to accomplish a specific task.

Since these games are intended for SCI patients, they must account for different capability levels that depend on the type, the level and the seriousness of injury. For this reason, the limitations and the objectives must be adjustable and customizable for each patient in order to not deliver stress situations.

One design consideration should be to warn the patient about the targets and the movement instructions of the session before the game starts. In addition, at the end of the game the results obtained should be clear to both physiotherapists and patients. At the same time, the patient should be encouraged to repeat the exercises, trying to do better by achieving more goals and making fewer mistakes.

Regarding the therapist part, he/she must be allowed to modify the main settings of the game. Furthermore, the therapist must have the possibility to save the patient's data to carry out a subsequent analysis of the user's performance. In this way, it is also possible to keep track of the patient's evolution in order to verify the progress in the rehabilitation phase.

For the part relative to the integrated haptic feedback with the thimble device, the virtual platform must ensure the opportunity to choose whether to include it or not during the session. Moreover, in order to adjust the device to the patient needs, it must provide the option of calibrating the device by pulling or releasing the cables.

In order to give the reader a clear idea of the previous observations, these have been summarized into a list of concrete design guidelines to follow for the implementation of the virtual reality games. The specifications are:

- **Essential environment:** the virtual platform must be free of any additional elements that could take the user away from the final goal.

- **Simplicity:** both the exercises and the objectives to be achieved must be easy and well explained. They must encourage the user to repeat the activity in order to trigger plasticity.
- **Flexibility:** the characteristics of the game, their limitations and the objectives must be adjustable to each patient.
- **Feedback:** the game must include the possibility to use or not the haptic device. Moreover, it must include a “Calibration” option.
- **Collection of data:** patient identification and performance during sessions must be available to the therapist in order to conduct following evaluations.
- **Reward system:** a motivating and encouraging feedback system increases the involvement of the player with the game and reduces the risk of abandonment.

Following these guidelines, the games were designed for SCI rehabilitation, also thanks to the support and advices of the therapists. The system offers visual and haptic feedback during the sessions, to increase the engagement, to facilitate the comprehension of exercises and to deliver a clear sense of progress.

As aforementioned, the rehabilitation platform is formed by three games. The goal of all of them is to achieve as many objectives as possible in a limited time window. If the patient does not achieve the objectives in the available time, the game will finish. All serious games follow the same coordinate reference that is the same used by Unity and superimposed on the Leap Motion controller. The games differ from each other by the scene presented, but more importantly by the objective of rehabilitation and the type of haptic feedback provided. A summary of the main characteristic of each game can be visualized in Table 3.1.

Game	Exercise Type	Feedback Provided	Therapeutic Objective
<b>Maze</b>	Path guidance	Vibration	Arm movement, Accuracy
<b>Path</b>	Path guidance	Pressure	Arm movement, Accuracy
<b>Grab</b>	Reaching objects, Grasp and release	Pressure	Coordination, Hand-grasp improvement

**Table 3.1 Main characteristics of each game.**

### **3.2.1 Introduction and Patient Management**

The virtual platform is composed of various panels in which the user can navigate before entering the games. These panels are essential because they give the patient and the therapist an idea of the set of information they can access before starting the rehabilitation session. It is essential to stress that the development and implementation of the virtual part of the project must have as its sole and real objective the creation of a platform that is centred on the therapy and on the patient, not on the game itself. For this reason, the interactable first part is made up of a simple design panel where thanks to few large buttons, it is possible to navigate between the options by just clicking them with the mouse. In particular, the scenes have always only a couple of buttons to execute the actions and inevitably one of them is the “back button”, which enables to return to the previous screen.

Figure 3.4 shows the path that can be taken between the panels to get to the choice of games. The instructions in the scenes must be clear to abstain from wasting time in the configuration process. The game is presented with the title "Rehabilitation" on the home screen, where with a button is possible to access the following panel. In this section the user is put in front of the choice of the profile that wants to access the games. The opportunity given is to choose between the selection of already registered patients or the addition of a new patient. In the last case the panel asks the user to introduce some parameters useful to create a new profile and add important information. This task is performed by the therapist and can be made before the therapy session.



**Figure 3.4** Path through the initial windows.



The required data are the name, the medical record number, the level of the injury in terms of ASIA classification, the etiology of the injury and the clinician who solicited the treatment (see Figure 3.5). These parameters are important for clinicians to control which patient is accessing the rehabilitation platform, thus to verify all his/her information.



**Figure 3.5 Essential information for the new patient.**

For each new patient added to the system, a folder with his/her name is created in the same location of the executable file of the game. Inside this folder a data file with the same name is placed. Here the therapist can find all the profile information saved in the game and can add new details such as the last session performed or medical notes. The creation of the folder is fundamental in order to collect all the files related to the same patient created after each game session. These files contain information such as the game played, the score obtained or the registered hands position that would be used in order to perform a posterior analysis. In this way, the clinicians can follow the evolution of each patient visualizing these data after each therapy session.

Once the new profile is created, it is visible in the Register panel. The button “Games” allows to access the last panel. In this panel, four icons are shown. Three of these are related to the respective games named Maze, Path and Grab, while the last one is linked to the option of calibration. The four icons are interactive and, when clicked, they open to the corresponding scene.

### 3.2.2 Common Functions

Before introducing the three games, it is essential to explain the scripts and functions that are used in all of them. The common features are related to data storage, recognition of interactions between objects in the scenes and communication with the Arduino to control the haptic device.

During the performance, the positions of the user index fingertip is continuously controlled and saved. To obtain these positions, a C# script called *HandFinder* has been developed. This script is used in the games and is always added to the *HandController* object. In this script the content is iterate every 0.02 seconds (fixed frame-rate). For each iteration, the code accesses the variables of the Leap Motion library. In particular, it accesses the frame generated by the camera in that specific moment and identifies, among the available variables, the tip of the index of the right hand. Then it identifies the position and speed variables of the fingertip and saves them in two vectors of three elements each (X,Y,Z), continuously updated. They are defined as public static vectors. This definition allows them to be visible to any script in the Unity scene.

The solution used to detect collisions between objects in the scenes is to use Unity colliders for each object that may be affected by interactions. The colliders are elements of the Unity class *Collider* that can be added to objects by inserting them in the Unity Inspector window. They can be 2D or 3D shaped, depending on the characteristics of the game. The colliders can be predefined geometric shapes or can be drawn as polygons in order to be perfectly superimposed on the objects in the scene. Colliders detect all collisions between them and send to all the scripts the content of three events:

- *OnCollisionEnter()*, that occurs in the frame in which the collision started.
- *OnCollisionExit()*, that is activated in the frame where the collision ends.
- *OnCollisionStay()*, that is active in all frames where the collision persists.

The communication with the Arduino board to activate and control the haptic device is managed thanks to the function *SendVibrationToArduino()* (see Figure 3.6).

```

public void SendVibrationToArduino(float value)
{
    if (MenuManagerMaze.IsArduinoConnected == true)
    {
        //send data to arduino board
        if (serial.IsOpen == false)
            serial.Open();
        serial.Write((value).ToString("000"));
    }
}

```

**Figure 3.6** The function used for the communication with the microcontroller.

This function first checks if the user is using the thimble and then if the serial communication is available. If both are true, it sends the value to the Arduino board, where the firmware analyzes the incoming data and manages the engine with the appropriate function.

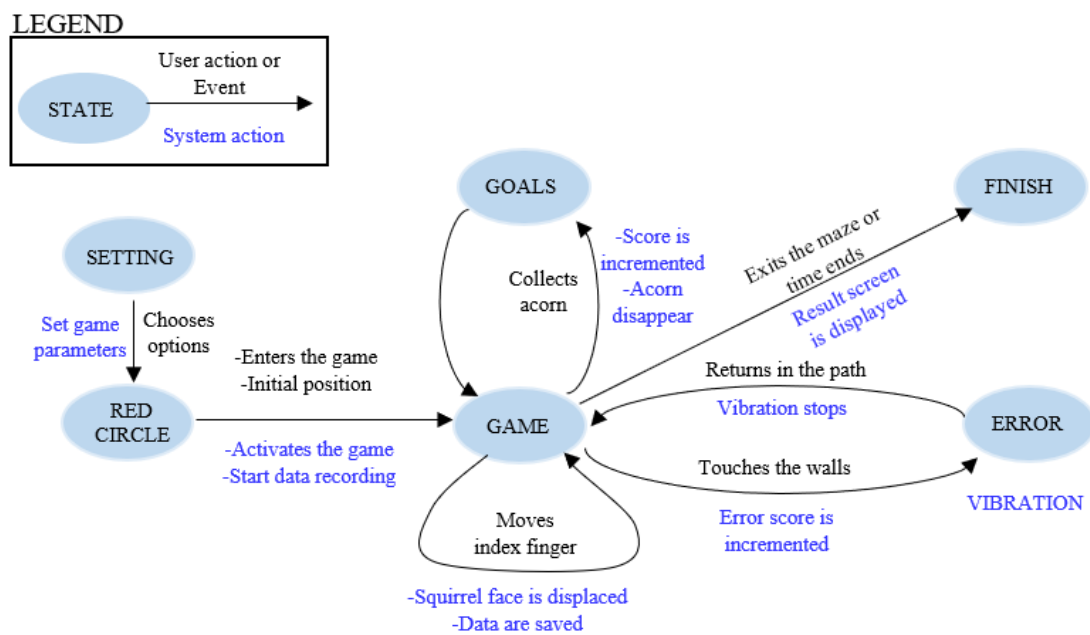
### 3.2.3 Maze

The first game developed is entitled “Maze”. The environment in this scene is in 2D. The name of the game comes from the scenario developed that is composed of a wooden structure that draws a labyrinth located on a grass lawn background (see Figure 3.7).



**Figure 3.7** Game window of Maze. Time, errors and points are continuously update. The red circle highlights the head of the squirrel controlled by the user.

The main idea is that the user controls the position of a squirrel with the index fingertip position. After setting the desired parameters, the user touches a red circle and the game starts. The objective is to guide the squirrel to collect the acorns positioned in different places of the scenario, thus gaining points. When the squirrel touches the walls, an error counter is increased and vibration feedback is provided to the user through the haptic device. The user must exit from the maze before the time ends. Figure 3.8 shows the main functioning of the game, highlighting both the user and the system actions that cause the change of game states.



**Figure 3.8 State transition networks (STN) for the Maze game.**

The most important elements used to design the Unity scene are five macro objects:

- The first one is the *HandController* that, as aforementioned, allows the integration of the Leap Motion controller in the Unity scene, thus enabling the user to see his/her hands in the scene and interact with other objects.
- The *GameManagerMaze* is an object that is not visible in scene but is fundamental since, thanks to the scripts attached, it controls the main variables exchanged between the scene elements, the data storage and the order in which the macro objects are displayed.

- The *MenuPanel* is the third element. It is the one that shows the setting panel to the users at the beginning. It contains toggles and sliders to controls important setting variables and a red circle that allows the passage to the *Game* object.
- The *Game* object contains all the element composing the proper game, i.e. the labyrinth structures, the acorns, the squirrel head controlled by the user and the panel showing time, points and error.
- The *End* object is activated once the game is over. It displays the screen with the results.

As repeatedly pointed out in the previous sections, this is a game intended for rehabilitation and therefore requires flexibility. In this way the therapist can adapt the game to the possibilities and characteristics of the patient. For this reason, the first screen shown is that of settings (see Figure 3.9).



**Figure 3.9 Settings window of Maze.**

In this window the user (or the therapist) is given the possibility to set the parameters. The most important parameter concerns the time. The duration of the game session can be set between 1,2 or 3 minutes by moving the corresponding slider. This is essential to allow, for example, patients with significant limitations to have more time to complete the objectives. The user is given the opportunity to play with or without the use of the haptic device. If he/she wants to receive the feedback, once the device is

worn, it is necessary to activate the toggle to enable communication between the computer and the hardware part of the device. This option has also been used to make a comparison between the game performances obtained with and without the device, as shown in Chapter 4. The last option available is whether or not showing the virtual hand while participating in the game. This opportunity has been implemented since it has been noted that the virtual hand can represent a visual obstacle. As shown in Figure 3.9, there is also another slider concerning the level. In this game this option is not selectable, but the slider has been left to add this option in future developments. For example, by choosing the level, the number of acorns could be decided or more complex labyrinth could be shown. In the Settings window, the hand is reproduced in virtual reality with a minimal hand. This allows the user to become familiar with the representation of movement. In the window there are two sentences that explain the objective of the game to the user and invite to place the hand on the red circle once the setting of the parameters is completed.

As soon as touched, the red circle deactivates the Settings window and activates the *Game* object. When switching from the Settings window to the Game window, the user touches the red circle and the representation of the hand remains blocked at that point. From that moment the positions of the user's index fingertip, retrieved thanks to *HandFinder* script, are imposed to the squirrel's head. The *GameManagerMaze* object collects the set parameters and passes them to the game. The game window shows a lawn of grass in the background and the structure of the maze (see Figure 3.7). The user looks at the scene from above, as if it were positioned over the structure. This is composed of two elements spaced out from each other that have a rectangular section. The two elements branch off with randomly positioned arms. The distance between the two elements represents the corridor of the labyrinth. The timer, errors and points are shown on the screen. These are continuously updated according to the user's actions.

Since the game is in 2D, only the vertical and horizontal movements of the fingertip affect the positions of the squirrel's head. At this point, the user navigates in the maze

trying to collect as many acorns as possible before exiting the maze at the point indicated by the rotating red arrow.

The solution used to detect collisions between objects is to use 2D Unity colliders for each object that may be affected by interactions, i.e. the squirrel head, the acorns and the labyrinth walls. For this game, only *OnCollisionEnter()* and *OnCollisionExit()* events are considered.

The C# script that uses the colliders is entitled *Contact\_maze*. It is attached to the object of the squirrel's head that has a circle 2D collider. Within this script, the interactions of the collider of the same object with other colliders are controlled. If a collision occurs, the name of the object containing the collider touched is checked. If the object is an acorn, then the points score increase and the acorn object is eliminated. If in *OnCollisionEnter()* is checked that the collider touched is the polygon figure superimposed on the wall objects, then the script activates the vibration. To do this, it sends the value "256" to the function *SendVibrationToArduino()* (see Figure 3.6).

If the user keep the position inside the wall, the thimble continues to vibrate but no data are sent using *OnCollisionStay()*. This behaviour is obtained thanks to the firmware, that continues the vibration option until a new serial data is received. For this reason, when the squirrel's head goes out of the wall, in the *OnCollisionExit()* the value "0" is sent to Arduino through *SendVibrationToArduino()*.

The game continues until the user exits the maze or the time ends. In both cases the *Game* object is disabled, and the *End* object is activated. This contains a panel that displays the points gained, the errors made and the eventual remaining time (see Figure 3.10). This information is useful for the patients to have the awareness of their own performance and for the therapist to follow the evolution during different sessions. In the same window a motivational phrase and a button to close the game are displayed. This rehabilitation game is based on a path guidance exercise. The therapeutic objective is to improve the precision in the UL movement made by the patient and to recover fine motor control. The game scene has been implemented to force the patient to move the entire arm. In fact, to reach the acorn in the upper part of the maze or just





**Figure 3.10 Results window of Maze.**

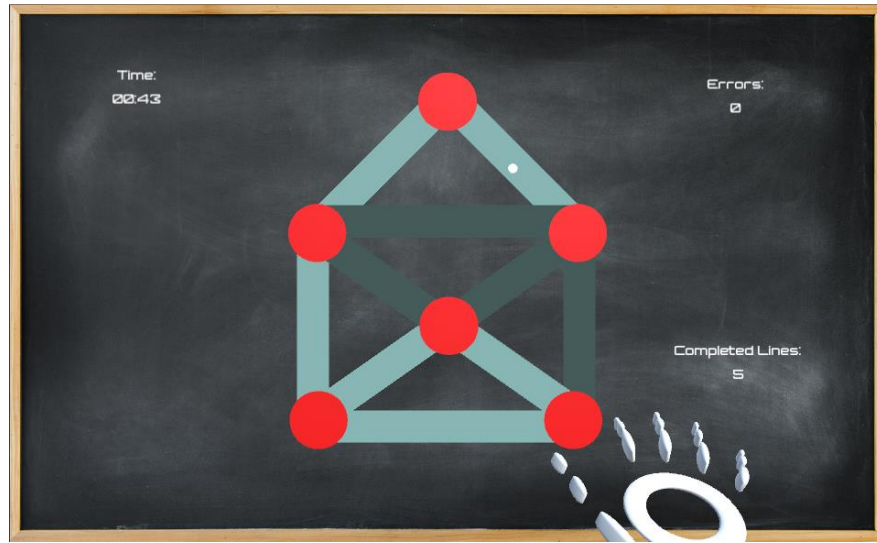
to travel the path avoiding errors, the patient is forced to raise the arm. Moreover, the therapeutic effect implies an improvement of accuracy because the user sees the objectives (acorns) and tries to reach them, paying attention not to touch the walls. The haptic feedback tries to make him aware of the errors, thus trying to reduce them.

A fundamental aspect of the implemented platform is the saving of movement data. When the game is started, the system creates a file ".txt" in the same path of the executable file of the game. When the user touches the red circle, the saving of data is enabled. In each frame the script *SavePositionMaze* accesses the position and velocity variables of the script *HandFinder* and saves them in a line of text in the txt file along with the value sent to the haptic device and the time. This information is used to perform a subsequent analysis useful for the therapist to study the progress of the rehabilitation process.



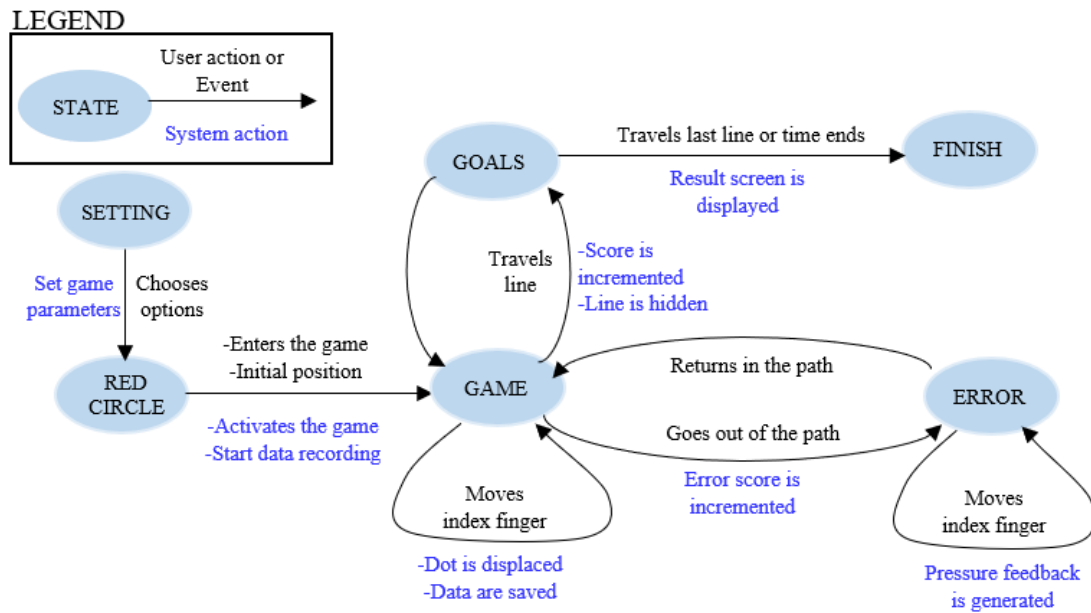
### 3.2.4 Path

The second game developed is entitled “Path”. This game is similar in the functioning to the Maze but is different in some aspects. The environment in this scene is in 2D too. The environment is composed of a blackboard inserted as a background and a figure drawn on it. This figure is a shape composed of different circles coloured in red and connected by blue lines (see Figure 3.11).



**Figure 3.11 Game window of Path. The white dot is on the line in the upper part. The travelled lines are obscured.**

The main idea in this case is that the user controls the position of a white dot with the index fingertip position. After setting the desired parameters, the user touches a red circle and the game starts. The objective is to guide the dot and to travel the line between the circles to gain points. When the dot goes outside the path, an error counter is increased. In this game the feedback implemented is different from the vibration of the Maze. If the user moves the white point out of the path, the haptic device provides pressure feedback by squeezing the finger in proportion to the distance from the path. The user must travel all the lines before the time ends. Figure 3.12 shows the main functioning of the game, highlighting both the user and the system actions that cause the change of game states.

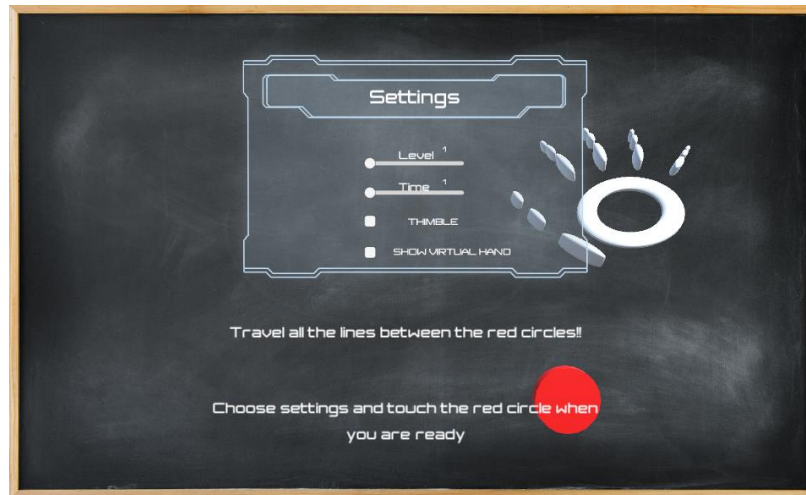


**Figure 3.12 State transition networks (STN) for the Path game.**

For this game, the same five macro objects were used to design the Unity scene. The *HandController* draws the same minimal hands in the virtual scene and allows the interaction with the objects in the scenes. The *GameManager* controls the main variables exchanged between the scene elements, the data storage and the order in which the macro objects are displayed. The *MenuPanel* shows toggles and sliders to control important setting variables and the red circle that allows the passage to the the *Game* object. Once the game is over, the *End* object is activated, displaying the screen with the results.

The setting window is similar to the one of the Maze (see Figure 3.13). The variables controlled are the time, the possibility to use the haptic device or not and the possibility to see the virtual hand or not. Nevertheless, in this game the variable “level” is selectable with value from one to three. This choice allows to decide the intensity of the pressure feedback received: the higher is the value, the higher is the pressure applied when going out of the path. This means that at the highest-level value, the patient feels almost the maximum pressure as soon as he/she goes outside the path.

In the Settings window, the user can freely move his hands. Two sentences explain the objective of the game to the user and invite to place the hand on the red circle once the setting of the parameters is completed.

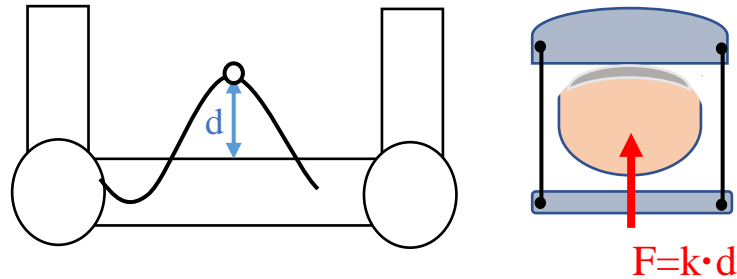


**Figure 3.13 Setting window of Path.**

To switch from the settings window to the game window, the user touches the red circle and the representation of the hand remains blocked at that point if the toggle “show virtual hands” is not selected.

As in the maze game, the *GameManager* object collects the parameters chosen in the Setting Window and passes them to the game. The *MenuPanel* is deactivated and the *Game* object is activated. The game window shows a blackboard with a figure drawn on it (see Figure 3.11). Passing to this window, the user starts the game in the initial position, i.e in the red circle situated in the bottom right corner. From that moment the user controls the white dot with the index fingertip position, obtained with the *HandFinder* script attached to the *HandController* object. Since the game is developed in 2D, only the x and y axis position components of the fingertip are used to control the white dot position. The user must travel the lines between red circles. Each time he passes between two red circles, the “completed lines” score is increased, and the light blue line is obscured. This counter is shown in the game window along with the countdown timer and the errors counter. The latter is incremented each time the user goes out of the path. For this game, the implemented feedback is different from the Maze game. In this case, when the dot goes outside the path, a pressure force feedback

is implemented. The principle on which this feedback is based is the Hooke's law. This states that the force  $F$  exerted by a spring is equals to a constant  $K$ , that depends on the



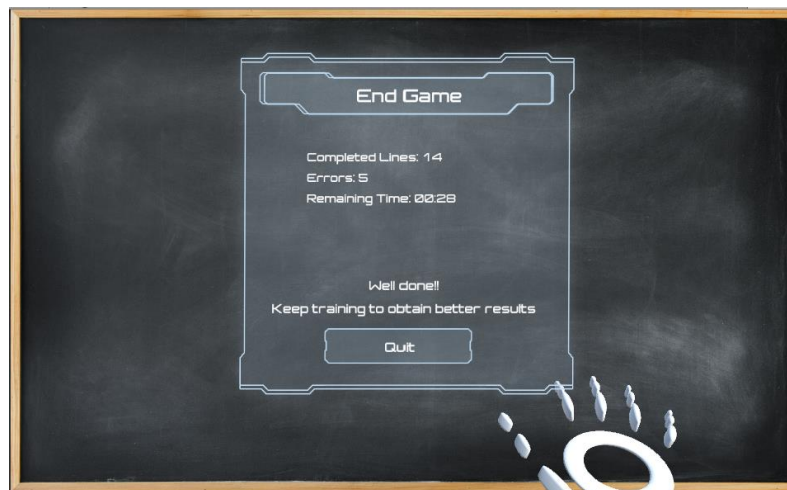
**Figure 3.14 Force feedback applied in Path depending on the distance of the white dot from the path. On the left the trajectory of the white dot from the shape moves away from the path of a distance “d”. On the right, the thimble, consisting of a base and a parallel platform (gray), tightens the finger (pink) by applying a force proportional to the distance.**

spring, times the displacement or change in length  $d$ . The force  $F$  used to express feedback, in the case of this game, is proportional to the distance  $d$  from the path and the constant  $k$  imposed by the user that chooses the level (see Figure 3.14).

To produce this feedback, the game software uses 2D Unity colliders. Specifically, it uses a function that allows to calculate the distance between two colliders within the scene. In this way, the developed C# script called *ContactPath* measures the distance between the collider attached to the white dot and the one attached to the whole figure. This is done in each frame, specifically each 0.02 seconds. If the distance is equal to or less than 0, it means that the white point is inside the path. In this case, a zero is sent to Arduino. If, on the other hand, the white point leaves the path, the distance increases until it reaches a maximum value established by the software developer. In this way, if the calculated distance exceeds this maximum value, the threshold value is saved. The entry and exit from the path are still controlled through the use of *OnCollisionEnter()* and *OnCollisionExit()* that activate safety flags. The distance to the path calculated when going outside the path is multiplied by the constant  $k$  set by the level selected at the beginning of the game and sent to Arduino. However, before being sent, this value is first mapped between 0 and 255 and then filtered by a median filter of three elements. When this process is completed, it is sent to the Arduino. This

activates the PD controller managed by the firmware and updates the motor position of the haptic device, thus exerting pressure on the fingertip.

The game continues until the user travels all lines or the time ends. In both cases the *Game* object is disabled, and the *End* object is activated. The last window displays the lines travelled, the errors made and the eventual remaining time (see Figure 3.15).



**Figure 3.15 Results window of Path.**

This information is useful for the patients to be conscious of their own performance and for the therapist to follow the performances during different sessions. In the same window a motivational phrase and a button to close the game are displayed.

As the Maze, this rehabilitation game is based on a path guidance exercise. The therapeutic objective is to improve the precision in the UL movement made by the patient when travelling the lines and to recover fine motor control. It strengthens the accuracy of the movement by imposing to remain in the path. The game scene has been implemented to force the patient to move the entire arm. In fact, the patient has to raise the arm to reach the red circles in the upper part. The haptic feedback tries to make the user aware of the errors and tries to redirect him/her to the path applying force proportional to the distance.

Also in this game, for each frame, the index fingertip position and velocity are saved in a file along with the values sent to the thimble and the time. The script implemented to accomplish this task is *SavePositionPath*. This information will be used for posterior analysis (see Chapter 4).

### 3.2.5 Grab

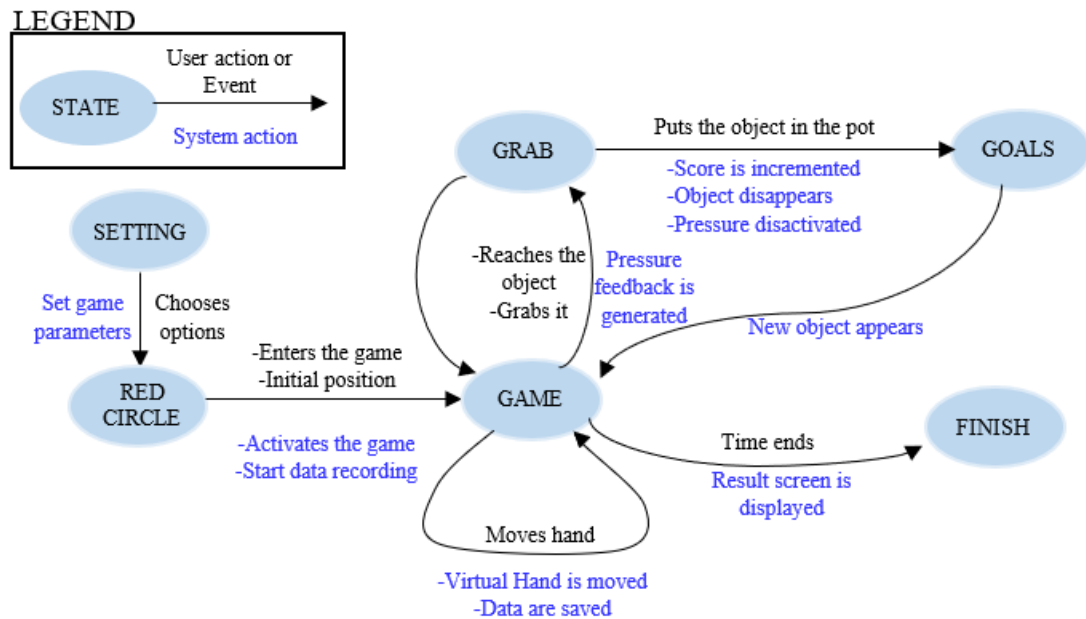
The third game developed is entitled “Grab”. This game is completely different from the other two, both in terms of scenario and rehabilitation purposes. In addition, it is a 3D game in which the user can freely move and see his hand in 3 dimensions, interacting with solid objects.

In this game, the user is situated in a virtual kitchen, located precisely in front of the cooker hob. The surface of the kitchen shows a wooden tray and a pot from which comes out steam. The main idea of this game is that, after setting the initial options, the user picks up the objects that appear in the tray and leaves them in the pot. Each time an object enters the pot, it is hidden, the points counter is incremented, and a new object appears on the tray (see Figure 3.16).



**Figure 3.16 Game window of Grab.**

When the user grasps an object, a pressure feedback is applied to the index fingertip. The sensation is different for different elements. The objective of this game is to leave as many objects as possible inside the pot in a given time. When the time ends, the results are shown. Figure 3.17 shows the main functioning of the game, highlighting both the user and the system actions that cause the change of game states.



**Figure 3.17 State transition networks (STN) for the Grab game.**

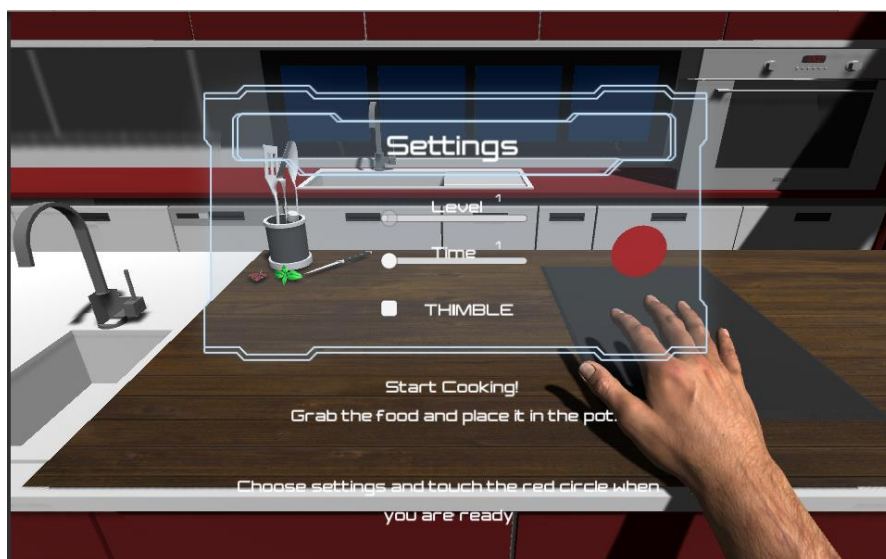
To design this scene, several objects were used. First of all the *HandController* is used this time to render a right virtual hand that has realistic aspect of an adult male hand. Since the scene is three-dimensional, this object was complemented with six invisible planes delimiting a box-shaped working area around surface positioned just in front of the user. This is useful to prevent objects from falling to the ground. The second object used is the *Kitchen*, the compound of the elements composing the scene. The models were downloaded for free from “*cgtrader*”<sup>2</sup> and “*free3d*”<sup>3</sup> as prefab models useful for ambient design rendering. The objects in the scene such as the camera, the light and the kitchen elements have been arranged to give the game user the idea of being located in the middle of the room, in front of a surface where the cooker hob is positioned. As in the other two games, the other elements present in the scene are the *MenuPanel*, the *Game*, the *End* and the *GameManager*.

<sup>2</sup> <https://www.cgtrader.com/>

<sup>3</sup> <https://free3d.com/>



The *MenuPanel* shows a setting window with less option than the others game (see Figure 3.18). It is only possible to set the Time value between 1,2 and 3 min and to decide if using the thimble or not. The “Level” slider is not selectable but leaves the opportunity to implement different improvements in future developments. For example, it would be possible to design different configuration of the space to allow different reaching-object exercises, or to decide the speed at which objects appear. In this window the user can freely move the hands. This is fundamental in this game since the user can understand how the representation of the 3D virtual hand movement



**Figure 3.18 Setting window of Grab.**

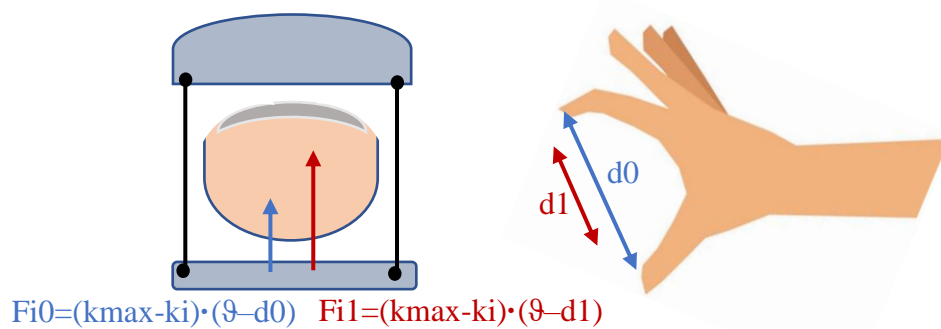
is similar to the reality, thus becoming familiar with the system before the game starts. When the user is ready, he/she can enter the game by touching the red circle. At this point the *MenuPanel* object is deactivated and the *Game* object activated. When this happens, the system starts recordings data relative to the user actions.

The *Game* element contains the wooden tray, the set of objects to be grabbed and the pot with the animation of the steam. Moreover, in the game window, the countdown timer and the points counter are shown (see Figure 3.16).

For this game, the solution used to interact with the objects relies on the Leap Motion library scripts adapted to achieve the objectives of this Thesis. The original solution requires the insertion of a game object called *PinchingHand* that contains the



*GrabbingHand* C# script. Moreover, it requires to add the script *GrabbableObject* to the objects. The insertion of these elements in the scene allows the user to grab the objects with the virtual hands. The scripts communicate with each other exchanging flags and variables. These scripts are used to recognize the distances between fingers thus allowing to grasp the object closest to the hand when this distance reaches a certain value. The principle is to avoid the use of the Unity physic engine that would require a lot of calculation, slowing the rendering of the scene. Once the object is grabbed, scripts allow to move the object at the same speed as the hand moves. The changes made to the code allow to apply feedback to the index fingertip when an object is grabbed. The feedback is proportional to the distance between the thumb and the index finger. The stronger the grab, the stronger the pressure exerted on the fingertip. Moreover, the modifications allow to add values of different hardness to objects, so that the user can perceive the difference between the objects. The principle for providing feedback requires these passages. If an object is close when fingers approach each other beyond a certain value named “grabbing distance”, it is grabbed. In this moment the object passes its hardness value to the feedback proportioning script (called *GrabbingHand\_Mod*). This, from the moment of the beginning of the grip, continuously checks the distance between the index finger and the thumb. The distance value is multiplied by the hardness of the object. This value is remapped to a value



**Figure 3.19 Haptic feedback provided for the same object. On the right the distance between fingers is shown. On the left, the device applies a force depending on the distance and on the k-value of the object (fixed in this case). 9 represents the grabbing distance, i.e. the threshold distance that the user needs to pass to grab an object. The smaller is the distance between fingers, the greater the force applied.**

between 0 and 255 so that smaller distances are matched by larger values to be sent (see Figure 3.19). Considering this proportionality, the hardness value given to the softer objects is higher in order to decrease the applied force.

The formula used in this case is:

$$F_{ij} = (k_{max} - k_i) * (\vartheta - d_j)$$

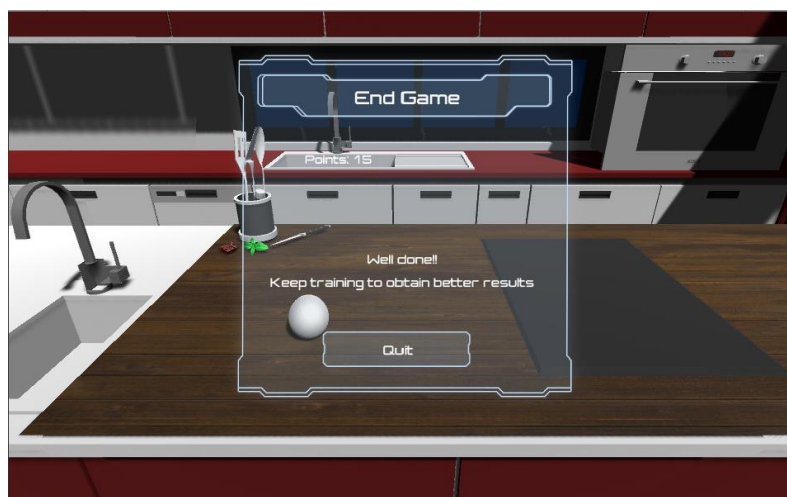
Where  $k_{max}$  represents the maximum value that  $k$  can acquire and  $\vartheta$  the grabbing distance, i.e. the threshold distance that the user needs to pass to grab an object.

The obtained data are filtered with a median filter as in the Path game. This time to make smoother the thimble behaviour, the system fills an array of 5 values that are rearranged in ascending order, and then sends the central value.

When the time ends, *Game* object is disabled and the *End* object is activated, showing the result panel with the points obtained (see Figure 3.20).

The rehabilitative objective of this game is to improve the reaching object capability, thus the coordination. Moreover, it aims to strengthen the hand grasp movement forcing the patient to repeat the same gesture several time. Thanks to the possibility of providing different hardness sensation, the objective of this game is also to improve the sensorial perception of the objects.

The *SaveDataGrab* script saves in a txt file the distance between index and thumb, the hardness value of the object grabbed, the time and the value sent to Arduino in each frame of the game.



**Figure 3.20 Results window of Grab.**

### 3.2.6 Calibration Scene

The last scene developed serves for the calibration of the thimble. In this scene, two big circles are shown. The green one untight the thimble when touched, while the red one tighten it. Each circle is accompanied by the indication of its function (see Figure 3.21). The user has to touch the circle for the desired function and wait until the desired configuration is reached. When the index goes out of the circle, the new position is saved as the zero-reference position that will be considered in the games. In the code these functions are achieved using the 3D colliders and their related events *OnCollisionEnter()* and *OnCollisionExit()*. When the index fingertip enters one of the two circles, the correspondent value (257 for releasing cables or 258 to tighten them) is sent to Arduino. The firmware continues to update the reference position until it receives a new serial data, i.e. the zero value sent by *OnCollisionExit()* when the index leaves the circle.

This scene is useful to adapt the device to different index fingers. Moreover, it can be used when, passing from one game to another, the device remains stationary in an unwanted configuration, due to some glitch or anomaly.

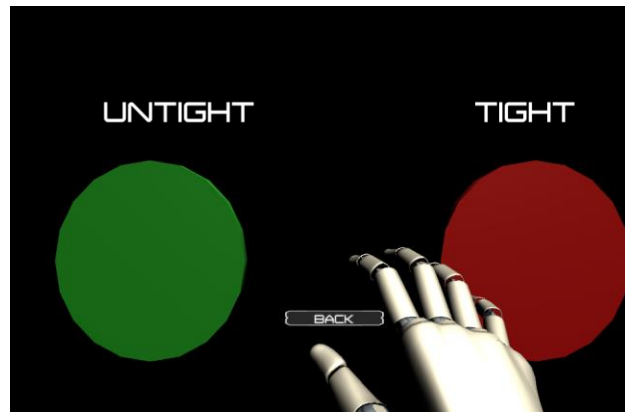


Figure 3.21 Calibration scene.



## **Chapter 4: Results**

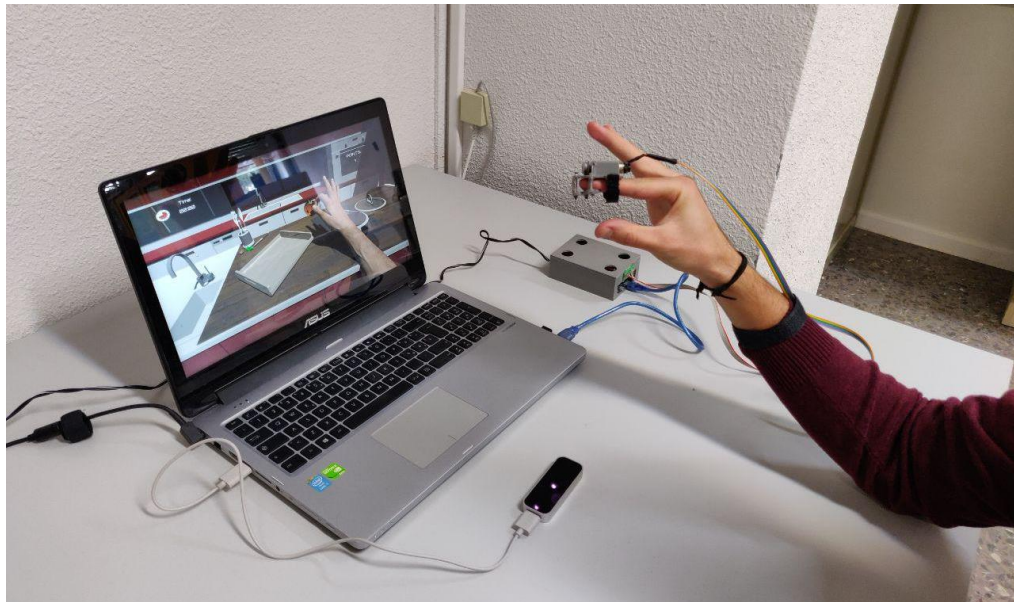
In this section the results of the Thesis are illustrated. First, the final version of the entire platform is shown. This represents the setup used in the trials carried out both with healthy subjects and SCI patients. Next, the result of the proof of concept performed in the Biomechanics and Technical Aids Department of National Hospital of Paraplegics in Toledo (Spain) is shown. The trials are aimed to test the three virtual reality games and check the usability of the rehabilitation platform.

The study was carried out to obtain data useful to support the feasibility of the platform developed for rehabilitation and to examine if SCI patients have the ability to properly manipulate the application. The collected data represent the key aspect of the platform. In fact, the objective information relative to the patient performance is useful for controlling the progress made during the therapy.

The information acquired is been analysed using the mathematical software tool Matlab to extract the main results. In this way, comparisons between the healthy subjects and SCI patients can be performed. In addition the participant were asked to compile system usability questionnaires to collect personal subjective evaluation of the platform.

### **4.1 Final platform prototype**

The previous chapters have highlighted all the problems and difficulties encountered during the implementation of the haptic thimble device and the virtual platform. This Thesis has been focused on developing a platform that can be used by the patients in their rehabilitation sessions at home. The final results show a low-cost platform prototype solid enough to be easily transported and used at home. The final version of the prototype is shown in Figure 4.1.



**Figure 4.1 Complete final platform.**

The computer represents the visual interface between the user and the games, but it is also the element that manages the execution of the games and ensures the communication between the different parts. No particular high performances are required for this platform, so a laptop can be used, thus allowing high portability and flexibility. The second element of the platform is the Leap Motion controller. It is a small affordable device that can be easily connected to the laptop through USB and can be placed on the desk. It is really compact and do not impede patient movements, while maintaining good performances in the hand movements recognition.

The computer is also wired to the Arduino board. Since this is associated with the power stage board, its dimensions result cumbersome. The solution encountered for this problem is to 3D print a case. This allows to connect the Arduino to the computer and the power stage board to the power network without interfering with the movement of the patient. In addition it closes and ensures all the connection between the boards. This hardware part is connected to the thimble through connectors and long cables. In this way the patient must wear only the thimble, thus being able to move without interferences. To sum up, the final prototype of the platform respects the requirements listed at the beginning.

## 4.2 Clinical Validation as a Proof of Concept

Once the platform has been completed in all its part, it needs to undergo to a proof of concept validation prior to a big clinical validation. The validation is important to understand the interaction of new users with the platform, especially the patients for whom it is intended. During the session, the user performance is monitored objectively by saving data through game scripts. At the end of the session, subjective opinions regarding the entire platform and game experience are collected from the users.

Thirteen people have participated to the platform validation, eight healthy subjects between 23 and 45 years of age and five cervical SCI patients. In Table 4.1, the clinical characteristics from all SCI patients are provided. It is fundamental to validate the platform with both groups to be able to compare their performance and evaluate the meaning behind the differences of results. It is important to highlight that, given the small number of participants, no statistical evaluation can be made. The arithmetic mean of the values of the two populations was considered for the comparisons.

The patients were contacted by the Biomechanics and Technical Aids Department of National Hospital of Paraplegics in Toledo (Spain). They were asked to participate in the trials voluntarily. All patients involved in the study need to have control of their UL including shoulder and wrist movement allowed. For the validation, it is extremely important to plan in detail the setup and the protocol to be used.

PATIENT	AGE	SEX	INJURY LEVEL	ASIA
1	40	F	C4 (incomplete)	D
2	33	M	C7 (incomplete)	C
3	43	M	C8 (complete)	A
4	40	M	C5 (incomplete)	B
5	43	M	C6 (complete)	A

**Table 4.1 SCI patients characteristics.**

The setup is exactly the same for the healthy subject and the patients. The only difference is in the structure on which the platform elements are placed. For the patients that are in a wheelchair, in fact, the normal table used for healthy subjects is not suitable since they assume uncomfortable positions. In their case, a height-adjustable table was used to allow the test to be carried out in the same position. The setup is mounted locating the Leap Motion controller at 10 cm in front of the computer. The box containing the boards is wired to the computer and positioned to its right, in order to insert the thimble on the index fingertip of the right hand of the user. The user places his/her elbow about 20 cm in front of Leap Motion controller. This is the starting position. Before the session starts, the user is introduced to the platform with a brief explanation of its characteristics. The three games are briefly explained in their main functioning and objectives along with the feedback haptic thimble device and the Leap Motion controller. Once the introduction part is completed, the game session starts. First of all, the initial windows are navigated and the data relative to the user are saved. In this way a folder and an excel file for each user are created. The first game accessed is Maze, then Path and at the end Grab. In the setting window the user is asked to move the hand in order to get used to the virtual representation of the hand movements. Two sessions for each game are completed. In the first one, the user is asked to wear the thimble, but no haptic feedback is applied. In the second one, the toggle “thimble” is ticked in each game, thus allowing the system to provide the feedback. Keeping the thimble, even if it is not activated, is useful to impose the same conditions in both sessions. In this way it is possible to make a meaningful comparison in the posterior analysis. There is an important difference in the data collected for the different game. In particular, for Maze and Path games the X, Y and Z position and velocities of the index fingertip are acquired along with the minutes and seconds from the beginning of the game. Moreover, the signal sent to the thimble is saved. These data are inserted in a file named with the game name, the date and exact time of the session. In the exercise Grab the important parameter saved is the distance between the thumb and index. Moreover, to have an idea of the performance, the hardness value of each picked object is saved along with the serial data sent to the thimble and the time for each frame.



In what follows, the analysis, results and considerations for each game are shown. In addition, examples of graphical representation of the data is performed to present the results.

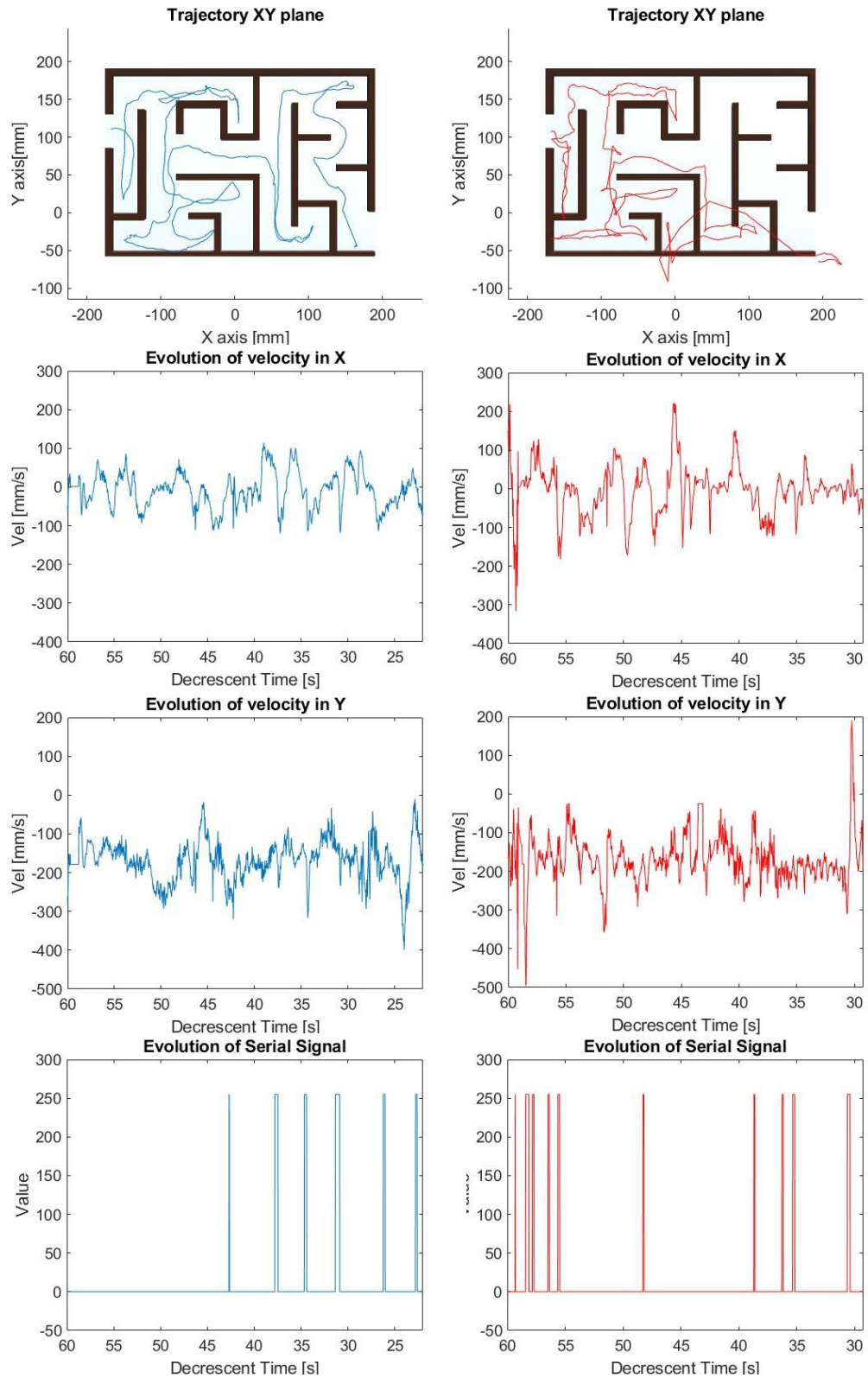
#### **4.2.1 Maze Results**

The Maze game requires the user to travel the maze reaching the acorns and collect them to increase points in a given time. The user must avoid touching the labyrinth walls, otherwise the error counter is increased. Once the user exits the maze or the countdown timer ends, the game is over.

For the clinical validation, the trials with both patients and healthy subject had the same configurations. The time imposed to complete the task and exit the maze was set to one minute. It was decided not to show the virtual hand during the exercises, since it represented an obstacle while travelling the maze in the left part. Two sessions for each patient were performed.

The data collected during the exercise are shown in Figure 4.2. The figure illustrates the graphs relative to one session performed by a health subject on the left part. On the right, it shows the graphs related to the performance of one patient.

The top graphs show the trajectory of the index fingertip during the game. Since the game is in 2D, only the X and Y axes are considered for analysis. The data collected are shown in millimetres. To ensure a perfect understanding of the trajectory, the graph created is superimposed on the image of the maze, integrated so as to ensure perfect scaling with the data. In this way, in the analysis phase a clearer view of where the errors were generated is provided. From the image, it is visible that the movement of the healthy subject is more fluid and accurate than that of the patient. The graphs in the second and third lines show the speeds in mm/s assumed by the user along the X axis and along the Y axis as a function of time respectively. On the abscissae axis the decreasing time is shown. These graphs are useful in the analysis to understand the direction of movement over time. The speed in the x-axis assumes decreasing values



**Figure 4.2** Graphs of data relative to Maze game. A healthy subject (left) and a patient (right) are compared.

if the user moves the index fingertip to the left while increasing values if he/she moves it to the right. In the Y-axis, the speed assumes decreasing values if the user moves his finger downwards and increasing values if he moves it upwards. The bottom graphs show at what time an error may have occurred and how long it lasted.

The data obtained from healthy subject and SCI patients have been analysed using Matlab. The values presented in Table 4.2 are the result of the arithmetic mean between the data obtained by each user in both trials (with and without feedback). The two groups have been analysed separately in order to make comparisons.

	<b>ROM X [MM]</b>	<b>ROM Y [MM]</b>	<b>TIME [S]</b>	<b>POINTS</b>	<b>ERRORS</b>
<b>HEALTHY</b>	362.86	231.79	36.62	8.50	10.69
<b>SCI</b>	369.33	277.34	49.50	6.90	16.70

**Table 4.2 Results of Maze game. The values represent the arithmetic mean of the data in both groups: healthy subject and SCI patients.**

The first value analysed for each user is the “range of motion”. In this Thesis, it is intended as the maximum distance reached in the game. For this reason, this is calculated as the difference between the maximum and the minimum value of the position reached with the index. The range of motion has been calculated for the X and the Y axes. The result shows that healthy subjects and SCI patients have the same ROM in the X axis, while in the Y axis the mean ROM is about 4 cm greater than the one of patients. Indeed, since the data are collected between the entry and exit of the maze that are placed respectively on the right and left, in the X-axis there is no difference between the two groups. The difference in the Y-axis instead suggests that the patients have left the upper and/or lower extremes of the maze (see Figure 4.2, top right). This is due to the less controlled movement of the limbs: once the movement has started, it is more difficult for the patients to stop. This is also reflected in the higher average error rate for patients. The average of the points obtained shows that in the exercise it is more difficult for the patients to reach all acorns before time ends.

Alternatively, this could mean that they cannot get to the places where the acorns are positioned. The results show that on average patients take longer to complete the exercise. This suggests slower movement. In the data collected, an attempt was made to evaluate the differences between the sessions completed with and without haptic feedback. The number of errors has been considered as the most significant for this evaluation. This consideration derives from the fact that the objective of the feedback is to make the user more aware of the interactions with the virtual environment (in this case errors). In both groups, the average of errors without feedback is higher than the average of errors made with the use of the haptic device (see Table 4.3).

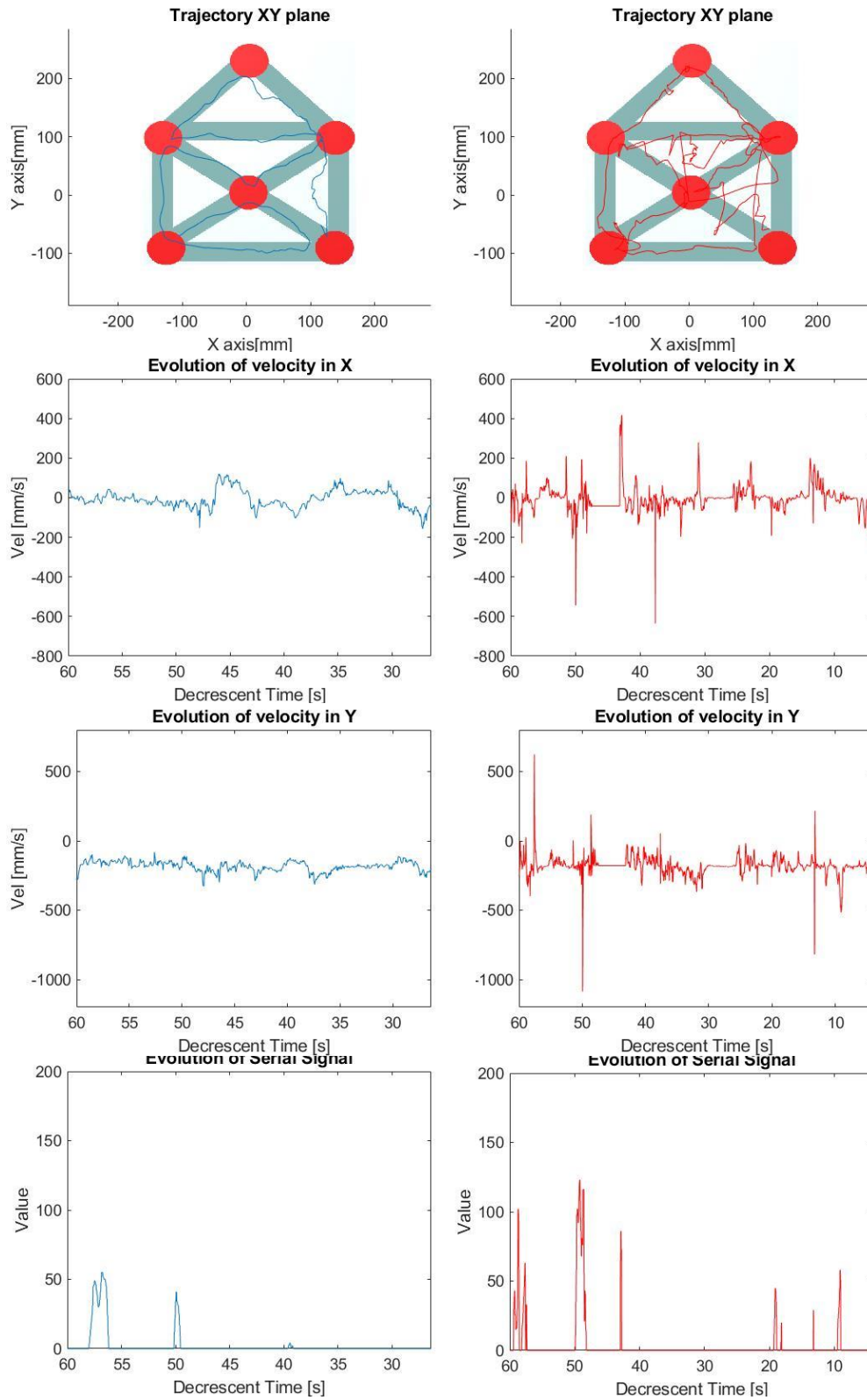
	<b>ERRORS</b>	<b>NO FEEDBACK</b>	<b>FEEDBACK</b>
<b>HEALTHY</b>	10.69	12.12	9.25
<b>SCI</b>	16.70	22.80	10.60

**Table 4.3 Errors in the Maze game. The average of the error is lower when the feedback is enables**

#### **4.2.2 Path Results**

The Path game requires the user to travel the blue lines in the shape, passing through the red circles to increase points in a given time. The user must avoid going out of the path, otherwise the error counter is increased. Once the user travels all the lines or the countdown timer ends, the game is over.

For the clinical validation the trials with both patients and healthy subject had the same configurations. The time imposed to complete the task was one minute. It was decided not to show the virtual hand during the exercises, since it represented an obstacle while travelling the figure, especially in the upper part. Two session for each patient were performed. In the first one the user is wearing the thimble, but no feedback is applied. In the other one, the thimble provides a pressure force feedback proportionally to the distance from the path in case of error. The data collected during the exercise are shown in Figure 4.3.



**Figure 4.3** Graphs of data relative to Path game. A healthy subject (left) and a patient (right) are compared.

The Figure illustrates the graphs relative to one session performed by a healthy subject on the left part. On the right side, it shows the graphs related to the performance of one patient. As for the Maze game, the top graphs show the trajectory assumed by the user during the game. The graph created is superimposed on the image of the figure drawn on the blackboard, scaled to perfectly fit the data.

The graphs in the second and third lines show respectively the speeds in mm/s assumed by the user along the X axis and along the Y axis as a function of time. On the abscissae axis the decreasing time is shown. Also for this game, the speed in the X-axis assumes decreasing values if the user moves the index fingertip to the left while increasing values if he/she moves it to the right. In the Y-axis the speed assumes decreasing values if the user moves his finger downwards and increasing values if he moves it upwards. The graphs in the bottom-line show at what time an error may have occurred and how long it lasted. Moreover, it displays the value sent to the Arduino which is proportional to the distance from the path.

The data obtained from healthy subject and SCI patients have been analysed using Matlab. The values presented in Table 4.4 are the result of the arithmetic mean between the data obtained from each user in both trials (with and without feedback). The two groups have been analysed separately in order to make comparisons. The results show no important differences in ROM both in X axis and in Y axis. The slight difference in Y can be attributed to the fact that the patients have difficulty in reaching the highest red circle. As for the previous game, the time needed by the patients to complete the exercise is higher than the one needed by healthy subjects.

	ROM X [MM]	ROM Y [MM]	TIME [S]	POINTS	ERRORS	MAX VALUE SENT
<b>HEALTHY</b>	288.65	330.04	38.62	13.94	7.75	114.31
<b>SCI</b>	282.86	319.98	54.40	14.1	10.10	90.3

**Table 4.4 Results of Path game. The values represent the arithmetic mean of the data in both groups: healthy subject and SCI patients.**

No important difference can be highlighted in the points gained by both groups because they just completed the path without passing between the same red circles more times. An interesting aspect is represented by the number of errors and the maximum value sent to the feedback device. Even if, as for the previous game, the average of errors is higher for the patients, the healthy subjects have a maximum value higher than patients. The explanation consists in the fact that, guided by curiosity, all the healthy subjects wanted to try a high value of force provided by the thimble. This was observed during trials but can also be inferred from Table 4.5. For the healthy patient, in fact, there is almost no difference between the error made without and with feedback. This is because even if they made more errors without feedback, in the session with the device enabled they induced the feedback sensation by going out from the path, thus committing errors. The results are different for SCI patients. Also in this case, the number of errors made without thimble is higher than the ones made with the thimble enabled.

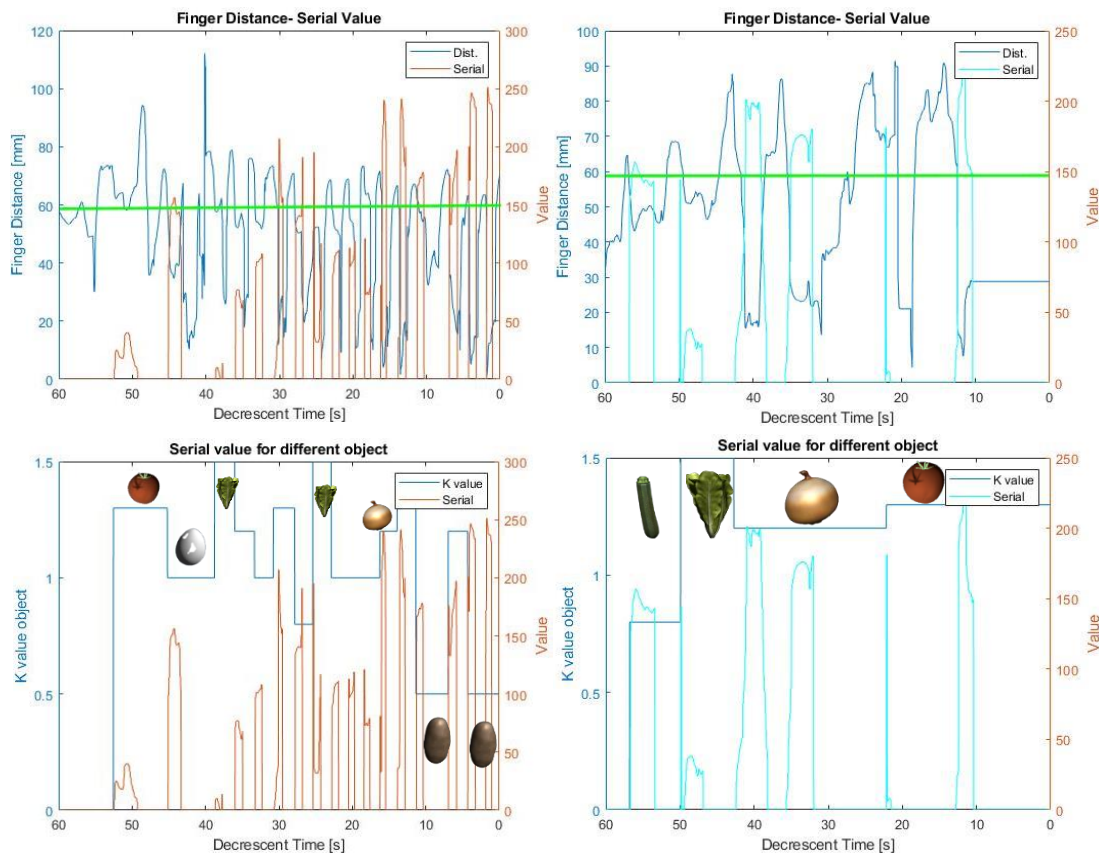
	<b>ERRORS</b>	<b>NO FEEDBACK</b>	<b>FEEDBACK</b>
<b>HEALTHY</b>	7.75	7.87	7.62
<b>SCI</b>	10.10	11.00	9.20

**Table 4.5 Errors in the Path game. The average of the error is lower when the feedback is enables for the patients. The errors are almost the same in both cases for the healthy subjects.**

### 4.2.3 Grab Results

The Grab game requires the patients to reach the object that appears in a wooden trail, grab it, move it above the pot and release it. Each time an object is left on the pot, the point counter increases. The objective is to put in the pot as many objects as possible in a given time. For the clinical validation the trials with both patients and healthy subject had the same configurations. The time imposed to complete the task was one minute. Two session for each patient were performed. In the first one the user is wearing the thimble, but no feedback is applied. In the other one, the thimble provides

a pressure force feedback each time the user grabs an object and holds it. The feedback is proportional to the distance between the index and the thumb. When the object is released a '0' value is sent. During the performance, data completely different from those of the other two games are collected. In particular, in the file created, the values of distance between index and thumb, the value sent to the haptic device, the K-value of the object taken and the time of play are saved. These data are then analysed with Matlab. Figure 4.4 illustrates the graphs relative to one session performed by a health subject (left) and by a patient (right).









**Figure 4.4** Graphs of data relative to Grab game. A healthy subject (left) and a patient (right) are compared. Examples of object are shown in correspondence of their k-value.



In the top figures, the value of distance between the fingers is superimposed to the value sent to the device. Both recorded signals are time dependent. In the bottom figures, the K-value of the collected object is superimposed to the value sent to serial. Also in this case, the two signals are time dependent.

In the top line images of Figure 4.4, a green line is shown on the distance and serial values. This represents a threshold level at about 6 cm. If the distance between fingertips is less than this value, the object is grabbed and so the proportional signal is sent to serial. As the distance between the fingers decreases, the signal sent in the serial port is greater. If the distance is above the threshold value, the object has not been taken yet. The value of the threshold was imposed considering that the user is wearing the thimble, so the parallel platform is placed between fingertips. If the value was lower, it would be possible to touch with the thumb the parallel platform even before grasping the virtual object, thus generating an incorrect and incomprehensible sensory information. By imposing the value at 6 cm, there is enough space to perform the grasping movement without touching the platform with the thumb.

The images in the bottom line in Figure 4.4 show for both healthy subjects and patients that the serial value is different with objects that have different k-values. Table 4.6 shows the grabbable objects with their k-values.

GRABBABLE OBJECT	K-VALUE
TOMATO 	1.3
EGG 	1
ONION 	1.2
ZUCCHINI 	0.8
LETTUCE 	1.5
POTATO 	0.5

**Table 4.6 Grabbable objects with the corresponding k-values.**

The lower the k-value, the greater the force given by the thimble. The biggest distinction is perceived between the potato and the lettuce.

The biggest difference that can be noted between a healthy subject and a patient is the number of times an object is taken. The trend of the distance value is also more stable and rhythmic in the healthy subject than the patient.

A significant element related to the performance of the users is the range of motion ROM, intended as the difference between the maximum and the minimum distance reached. The average value of all patients is very similar to that of healthy subjects as can be seen from Table 4.7.

	ROM [MM]	POINTS
<b>HEALTHY</b>	95.92	10.88
<b>SCI</b>	106.8	6.4

**Table 4.7 Results of Grab game. The values represent the arithmetic mean of the data in both groups: healthy subject and SCI patients.**

However, the graphs cannot show how many times the entire task of reaching the object, grasping it and releasing it into the pot has been performed. For this reason, the number of points shown at the end of the exercise is saved in the txt file. The average number of points obtained by the patients is lower than that of the healthy subjects (see Table 4.7). From all this information it can be deduced that, even if the movement carried out by the patients is correct, the time they need to complete the task is much higher.

#### **4.2.4 System Usability Questionnaire**

After the performance of the therapy session, both SCI patients and healthy people provided a feedback about their experience with the platform by filling out a System Usability Questionnaire. It is anonymous and it is an excellent tool to obtain opinions about the whole platform. This evaluation can be used to improve the system and the experience with the games and the device in the future. The questionnaire provided uses a 5-point Likert scale. The points are equivalent to: 1 Strongly Disagree, 5 Strongly Agree and 3 Neutral. Two questionnaires have been developed, one for the healthy people (see Table 4.8) and another one for the patients (see Table 4.9).

The two versions share the same nine initial questions but differ in the lasts. The questionnaire referring to the group of healthy patients contains a last question “You think this would help upper limb traditional rehabilitation for SCI patients”. This

question expresses a simple subjective opinion of the healthy user to whom has been explained in what SCI consists and what are its visible consequences in patients. The other questionnaire contains questions more related to patients' opinions. In particular, the questions added are: “This type of rehabilitation session is more interesting than the traditional one”, “You would suggest combining this rehabilitation with the traditional one” and “The games are appropriate for someone with you level of injury”. The first two collect the opinion regarding the comparison with the traditional rehabilitation and the possibility of integration, while the last one focuses on the particular case of each patient.

<b>Questions</b>	<b>Mean score (<math>\pm</math> s.d.)</b>
Q1.The games were enjoyable to play.	4.75 $\pm$ 0.43
Q2.It was easy to understand how to play games.	4.75 $\pm$ 0.43
Q3.The games have reacted well to your hand movements.	4.12 $\pm$ 0.60
Q4.You would play the games in your free time.	4.12 $\pm$ 1.05
Q5.The games have motivated you to continue with the exercises.	4.50 $\pm$ 0.50
Q6.Playing with the thimble made you more aware of the interactions in VR than playing without it.	4.75 $\pm$ 0.43
Q7.It was easy to wear the thimble.	4.62 $\pm$ 0.70
Q8.The platform was helpful for tracking the progress of your performance.	4.75 $\pm$ 0.43
Q9.You would be comfortable to do this kind of exercises at home without support.	4.87 $\pm$ 0.33
Q10. You think this would help upper limb traditional rehabilitation for SCI patients.	4.50 $\pm$ 0.50

**Table 4.8 System Usability Questionnaire for healthy subjects.**

The results of the questionnaire for healthy subjects are very positive. The average response for all questions is above 4. It is important to highlight some of the answers relative to different aspects of the platform. Q6 and Q7 are relative to the thimble

device. It is clear that the perception of interactions with the virtual environment is actually improved thanks to the feedback. The lower score is related to the question number three and four. Number three is important since it shows that the representation of hand movements was not perfect. This understandable since the Leap Motion controller performance is affected by the light in the surrounding environment. Moreover, it must be considered that the thimble device is an object that modifies the silhouette of the hand and makes its representation in virtual space more difficult. Q9 is the one with the highest score. This is important because it shows that the user is not frightened by the game and is confident in using the platform without any help. This, as reported in Q2, indicates that the games are easy to understand. Table 4.9 shows the questions asked to the patients.

<b>Questions</b>	<b>Mean score (<math>\pm</math> s.d.)</b>
Q1.The games were enjoyable to play.	4.80 $\pm$ 0.40
Q2.It was easy to understand how to play games.	5.00 $\pm$ 0.00
Q3.The games have reacted well to your hand movements.	4.20 $\pm$ 0.40
Q4.You would play the games in your free time.	3.80 $\pm$ 1.60
Q5.The games have motivated you to continue with the exercises.	4.40 $\pm$ 0.80
Q6.Playing with the thimble made you more aware of the interactions in VR than playing without it.	5.00 $\pm$ 0.00
Q7.It was easy to wear the thimble.	5.00 $\pm$ 0.00
Q8.The platform was helpful for tracking the progress of your performance.	5.00 $\pm$ 0.00
Q9.You would be comfortable to do this kind of exercises at home without therapist.	5.00 $\pm$ 0.00
Q10. This type of rehabilitation session is more interesting than the traditional one.	4.80 $\pm$ 0.40
Q11. You would suggest combining this rehabilitation with the traditional one.	5.00 $\pm$ 0.00
Q12. The games are appropriate for someone with you level of injury.	4.40 $\pm$ 1.2

**Table 4.9 System Usability Questionnaire for patients.**

The results show very high scores, and this suggests a general enthusiasm of the patients in the use of the platform. All patients answered with the maximum score for half the questions. Among these, Q6 and Q7, already mentioned above, show that the integration of haptic feedback worked well and made the user more aware of the integration with the virtual environment. The score given to Q8 suggested the effectiveness of the results panel in making the subjects aware of their own performance. This allows them to have an idea of the progress achieved during the rehabilitative sessions. The answers to Q9, Q10 and Q11 show that patients would like to integrate this type of rehabilitation with the traditional one so that they can continue their training at home. It is finally necessary to highlight that the answer to Q12 indicates that the platform is suitable for patients with SCI.

In addition to these questions related to the platform's experience during the game session, users were asked one last question. It was considered important to assess the perception of effort during the rehabilitation session. For this reason, the user was asked to indicate the level of perceived effort and fatigue on a scale from 0 to 10. This scale is based on the Borg scale, a scoring method to identify the physical effort perceived by the subject. The results are shown in Table 4.10. The arithmetic mean of the scores given by users shows how healthy subjects felt almost no fatigue at all. Patient responses show how generally the perception of fatigue is greater than the perception of healthy subjects. In this case, the high standard deviation suggests that different patients with different injury levels perceive the effort very differently. However, the fact that patients feel quite fatigued after the exercise suggests that the games presented require an effort and that training of the upper limbs actually occurs.

<b>MEAN SCORE (<math>\pm</math> S.D.)</b>	
<b>HEALTHY</b>	1.87 $\pm$ 0.78
<b>SCI</b>	4.40 $\pm$ 2.24

**Table 4.10 Level of perceived effort in a scale from 0 to 10.**



# **Chapter 5: Conclusions and Future Developments**

## **5.1 Conclusions**

The presented Thesis fits into a specific context, the rehabilitation of the upper limbs of patients affected by Spinal Cord Injury. These patients have a weak control of the arm movements and of the hand gestures. In this scenario, robotics devices and VR technologies facilitate the performance of the necessary movement repetitions. In this way, they induce the UL motor learning and increase the amount of therapy that patient receives. However, these technologies are often expensive and cumbersome. Some devices are uncomfortable to wear, and some rehabilitation platforms do not provide any feedback to the affected limb. This often leads to an abandonment of rehabilitation platforms. For these reasons, they are not suitable for domestic use which would allow to further increase the amount of therapy results.

This Thesis tries to respond to the need of a low-cost platform, easy to use for the user and that allows the therapist to follow the progress of rehabilitation. The implementation of a platform, that requires the use and communication of different elements, has allowed to explore, test and improve solutions belonging to different areas, including video game programming, mechanical design and automatic control. First of all, three games have been designed and implemented for SCI patients, meeting the needed requirements for an adequate rehabilitation. By programming in Unity software, it has been possible to develop two games in 2D and one in 3D. The user can interact with the games just by using his hand, thus focusing only in moving the arm and the hand in the correct way. To achieve this, the platform takes advantage of the LeapMotion device to represent the hand movements and access the data relative to positions and velocity of the user index fingertip. The platform has been developed to provide the user with the ability to receive haptic feedback that allows him to be more aware of the interaction with the virtual environment. The work shown in this Thesis

also focuses on the development of an easily wearable device that can provide two different types of tactile feedback: vibration and pressure. The mechanical parts that compose the thimble have been designed with CAD software and have been printed in 3D. These parts have been then integrated with a low-cost motor controlled with Arduino. The implemented firmware allows Arduino to receive data from the computer, analyse them and manage the motor to provide the two types of feedback. In particular, the pressure feedback is generated thanks to a specifically implemented PD controller.

The communication between the elements allows the user to actively participate in the rehabilitation session and to receive feedback when interacting with the virtual environment. The implemented platform allows to collect data related to the patient's performance, thus allowing the therapist to follow the progress of the recovery.

This study has gone a step further and has made a proof of concept to evaluate the platform and its usability with SCI patients. Tests were conducted with both healthy subjects and patients. The data collected show some differences between the performances of the two groups. However, the results obtained from this pilot test cannot be conclusive due to the small sample size. This limitation does not allow to perform a detailed statistical analysis useful for reaching strong conclusions regarding the difference between the groups.

Nevertheless, the tests have shown the potential of the three serious games to obtain relevant information about the patient performance. Moreover, the fact that all the SCI patients finished the experimental sessions in a successful way shows that the environment and the thimble haptic device implemented were appropriate for SCI people capabilities. In addition, at the end of the trials, users were asked to fill in questionnaires to collect personal opinions about the general experience, but also specifically about the virtual environment and thimble. The results obtained are encouraging.

Some of the main benefits drawn from this Thesis are summarized as follows:

- All the devices used to build the platform are low-cost and easily adaptable for different people, with a short time preparation for the session.



- The configuration options used in this project allows to create different virtual scenarios and training situations personalized for different patients.
- The final design of the thimble and of the electronic connections ensures a good mobility and generates a good haptic response.
- The data storage makes the platform a useful tool for monitoring progresses in rehabilitation.

In conclusion, this Thesis presents an innovative rehabilitation platform for training the upper limb, to improve the strength and fine movements. It is a customizable, low-cost, haptic and VR based platform that can increase the time spent in therapy sessions and can be used to complement the traditional rehabilitation, improving the motivation of the patients in their rehabilitation process.

## 5.2 Future Developments

The work presented in this Thesis represents only an initial step that opens several lines of future work to ameliorate this rehabilitation platform. Several improvements in different parts of the project have been detected:

- On the mechanical part:
  - To improve the design of the elements. This would be useful to ameliorate how the device is fixed to the user fingertip. Other materials can result in better functioning and better contact sensation with the skin.
  - To redesign the element to be stronger and lighter, thus to improve the mobility. By reshaping the elements, different solutions can be found not to impede the user movements and gestures.
  - To study a different configuration to connect the reel and the motor. For example, another solution can be found to replace the cable system with something stronger and more efficient.
  - To reduce the size of the device to allow a better recognition of the hand movement by the LeapMotion sensor. A smaller device would not interfere with the index

finger silhouette, thus permitting the sensor to track perfectly the position of the finger and its joints.

- To redesign the box containing the motor. By reshaping the base element, the possibility to integrate different motor chosen according to necessity.
- To integrate a smaller motor. This would lead to a smaller and lighter device, with the advantages already illustrated.
- To build more than one thimble. The availability of multiple elements would allow easy replacement in case of failure but would also allow to provide feedback to all fingers. Moreover, it would be possible to design a thimble appropriate for children.

- On the electronics side:

- To reduce the size of the board, including only the elements useful for the platform. This will require the design of ad-hoc electronics and connections for the device.
- To implement the control for more than one motor. This would provide feedback simultaneously to other fingers, thus allowing a full immersion of the patient hand.
- To eliminate the cables communication making all the system wireless. This would allow to ameliorate the mobility of the whole system. In this way, it would be possible to design thimbles easier to wear.

- About the Virtual Platform:

- To include more games for different rehabilitation purposes. Different movements and different tasks would ameliorate the rehabilitation performance. This would result in a more complete and faster recovery of the upper limb.
- To include the rehabilitation of both hands. This would require a considerable effort in the electronic and mechanical part, but would ensure the patient the possibility of training both limbs.
- To include the recognition of all fingers and their interaction. By accessing all the variables the LeapMotion controller saves, it would be possible to provide different

feedback sensations for each finger, thus ameliorating the whole patient involvement in the rehabilitation session.

- To ameliorate the existing games with different levels and options. By improving the games already developed, it would be possible to provide different tasks. For example, in the Maze game, it would be possible to create several mazes to force the patient to perform different movements.

# BIBLIOGRAPHY

- Adamovich, S. V., Merians, A. S., Boian, R., Lewis, J. A., Tremaine, M., Burdea, G. S., . . . Poizner, H. A Virtual Reality—Based Exercise System for Hand Rehabilitation Post-Stroke. *Presence*, 2005; 14(2):161-174.
- Andaluz, V., Salazar, P., Escudero, V., Bustamante, D., Silva, S., Quevedo, W., . . . Rivas, D. Virtual Reality Integration with Force Feedback in Upper Limb Rehabilitation. *Advances in Visual Computing*, 2016; 259-268.
- Arduino. *Arduino Due*. Retrieved from <https://store.arduino.cc/due>, 2019.
- Autodesk. *Inventor*. Retrieved from <https://www.autodesk.it/products/inventor/overview>, 2019.
- Baricich, A. M., Amico, A. P., Zampolini, M. M., Gimigliano, F. M., Cisari, C. M., & Fiore, P. M. People with Spinal Cord Injury in Italy. *American Journal of Physical Medicine & Rehabilitation*, 2017; 96(2):80-82.
- Bertomeu-Motos, A., Blanco, A., Badesa, F., Barios, J., Zollo, L., & Garcia-Aracil, N. Human arm joints reconstruction algorithm in rehabilitation therapies assisted by end-effector robotic devices. *Journal of NeuroEngineering and Rehabilitation*, 2018; 15:10.
- Bickenbach, J., Officer, A., Shakespeare, T., von Groote, P., WorldHealthOrganization, & et al. International Perspectives on Spinal Cord Injury (IPSCI), 2013.
- Bouzit, M., G., B., Popescu, G., & Boian, R. The Rutgers Master II-new design force-feedback glove. *IEEE/ASME Transactions on Mechatronics*, 2002; 7(2):256-263.
- Cappello, L., Meyer, J., Galloway, K., Peisner, J., Granberry, R., Wagner, D., . . . Walsh, C. Assisting hand function after spinal cord injury with a fabric-based soft robotic glove. *Journal of NeuroEngineering and Rehabilitation*, 2018; 15: 59.
- Chinello, F., Malvezzi, M., Pacchierotti, C., & Prattichizzo, D. Design and development of a 3RRS wearable fingertip cutaneous device. *IEEE*

- International Conference on Advanced Intelligent Mechatronics*, 2015; 293-298.
- Choi, D., Nam, G., Groh, D., Syed, S., Fridley, J., & Gokaslan, Z. Spinal Cord Anatomy. In K. Arnautović, & Z. Gokaslan, *Spinal Cord Tumors*. Springer, Cham, 2019.
- Colgan, A. *How Does the Leap Motion Controller Work?* Retrieved from <http://blog.leapmotion.com/hardware-to-software-how-does-the-leap-motion-controller-work/>, 2014.
- Connelly, L., Jia, Y., Toro, M., Stoykov, M. E., Kenyon, R. V., & Kamper, D. G. A Pneumatic Glove and Immersive Virtual Reality Environment for Hand Rehabilitative Training After Stroke. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 2010; 18(5):551-559.
- CyberGloveSystems. *CyberGrasp*. Retrieved from CyberGloveSystems: <http://www.cyberglovesystems.com/cybergrasp>, 2019.
- Demain, S., Metcalf, C., Merrett, G., Zheng, D., & Cunningham, S. A narrative review on haptic devices: Relating the physiology and psychophysical properties of the hand to devices for rehabilitation in central nervous system disorders. *Disability and rehabilitation. Assistive technology*, 2012; 8.
- Diaz, E., & Morales, H. Spinal cord anatomy and clinical syndromes. *Semin Ultrasound CT MR*, 2016.
- Dimbwadyo-Terrer, I., Trincado-Alonso, F., de los Reyes-Guzmán, A., Aznar, M. A., Alcubilla, C., Pérez-Nombela, S., . . . Gil-Agudo, Á. Upper limb rehabilitation after spinal cord injury: a treatment based on a data glove and an immersive virtual reality environment. *Disability and Rehabilitation: Assistive Technology*, 2016; 11(6):462-467.
- Ding, Y., Kestin, A. J., & Pan, W. Neural plasticity after spinal cord injury. *Current pharmaceutical design*, 2005; 11(11):1441-1450.
- Dunlop, S. Activity-dependent plasticity: implications for recovery after spinal cord injury. *Trends in neurosciences*, 2008; 31:410-418.

- Duret, C., Grosmaire, A. G., & Krebs, H. I. Robot-Assisted Therapy in Upper Extremity Hemiparesis: Overview of an Evidence-Based Approach. *Frontiers in neurology*, 2019; 10:412.
- Haisma, J., Bussmann, J., Stam, H., Sluis, T., Bergen, M., Dallmeijer, A., . . . van der Woude, L. Changes in physical capacity during and after inpatient rehabilitation in subjects with a spinal cord injury. *Archives of Physical Medicine and Rehabilitation*, 2006; 87(6):741-748 .
- Harvey, L. A. Physiotherapy rehabilitation for people with spinal cord injuries. *Journal of Physiotherapy*, 2016; 62(1):4-11.
- Harvey, L., Dunlop, S., Churilov, L., Hsueh, Y., & Galea, M. Early intensive hand rehabilitation after spinal cord injury ("Hands On"): a protocol for a randomised controlled trial. *Journal of Physiotherapy* , 2016; 62(2):88-95.
- Hocoma, S. *Armeo®Spring Pediatric*. Retrieved 10 14, 2019, from <https://www.hocoma.com/media-center/media-images/arneo-manovo/>, 2019.
- Isa, T., & Nishimura, Y. Plasticity for recovery after partial spinal cord injury – hierarchical organization. *Neuroscience Research*, 2014; 78:3-8.
- Kadivar, Z., Sullivan, J., Eng, D., Pehlivan, A., O'Malley, M., Yozbatiran, N., & Francisco, G. Robotic training and kinematic analysis of arm and hand after incomplete spinal cord injury: A case study. *IEEE ... International Conference on Rehabilitation Robotics : [proceedings]*, 2011; 5975429.
- Karimi, M. Robotic Rehabilitation of Spinal Cord Injury Individual. *Ortopedia Traumatologia Rehabilitacja*, 2013; 15(1):1-7.
- Kim, M., Jang, I., Lee, Y., Lee, Y., & Lee, D. Wearable 3-DOF Cutaneous Haptic Device with Integrated. *International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, 2016; 649.
- Kirshblum, S. C., Burns, S. P., Biering-Sorensen, F., Donovan, W., Graves, D. E., Jha, A., . . . Waring, W. International standards for neurological classification of spinal cord injury (revised 2011). *The journal of spinal cord medicine*, 2011; 34(6):535-546.

- Kleim, J., & Jones, T. Principles of Experience-Dependent Neural Plasticity: Implications for Rehabilitation After Brain Damage. *Journal of Speech, Language, and Hearing Research*, 2008; 51:225-239.
- LeapMotion. *LeapMotionController*. Retrieved from <https://www.leapmotion.com>, 2019.
- Lo, C., Tran, Y., Anderson, K., Craig, A., & Middleton, J. Functional Priorities in Persons with Spinal Cord Injury: Using Discrete Choice Experiments To Determine Preferences. *Journal of neurotrauma*, 2016; 33(21):1958-1968.
- Maciejasz, P., Eschweiler, J., Gerlach-Hahn, K., Jansen-Toy, A., & Leonhardt, S. A survey on robotic devices for upper limb rehabilitation. *Journal of neuroengineering and rehabilitation*, 2014.
- Macovei, S., & Doroftei, I. A short overview of upper limb rehabilitation devices. *IOP Conference Series Materials Science and Engineering*, 2016; 145(5):052014.
- Mangold, S., Keller, T., Curt, A., & Dietz, V. Transcutaneous functional electrical stimulation for grasping in subjects with cervical spinal cord injury. *Spinal cord.*, 2005; 43:1-13.
- Massetti, T., da Silva, T. D., Crocetta, T. B., Guarnieri, R., de Freitas, B. L., Bianchi Lopes, P., . . . de Mello Monteiro, C. B. The Clinical Utility of Virtual Reality in Neurorehabilitation: A Systematic Review. *Journal of Central Nervous System Disease*, 2018; 10.
- Mekki, M., Delgado, A., Fry, A., Putrino, D., & Huang, V. Robotic Rehabilitation and Spinal Cord Injury: a Narrative Review. *Neurotherapeutics*, 2018.
- Meli, L., Scheggi, S., Pacchierotti, C., & Prattichizzo, D. Using the Leap Motion controller for hand tracking and wearable haptic devices for contact rendering. *World Haptics*, 2015.
- Merians, A., Jack, D., Boian, R., Tremaine, M., Burdea, G., Adamovich, S., . . . Poizner, H. Virtual Reality–Augmented Rehabilitation for Patients Following Stroke. *Physical Therapy*, 2002; 82(9):898-915.

- Merrett, G., Metcalf, C., Zheng, D., Cunningham, S., Barrow, S., & Demain, S. Design and Qualitative Evaluation of Tactile Devices for Stroke Rehabilitation. *IET Seminar Digest*, 2011.
- Merritt, C. H., Taylor, M. A., Yelton, C. J., & Ray, S. K. Economic impact of traumatic spinal cord injuries in the United States. *Neuroimmunology and Neuroinflammation*, 2019.
- MichelsonMedical. *Spinal Cord – Areas the nerves control*. Retrieved 10 2, 2019, from michelsonmedical.org: <https://michelsonmedical.org/2014/11/21/spinal-cord-injury-critical-research-pva/pva-spinal-cord-3-areas-nerves-control/>, 2014.
- Murphy, C., & Chuinard, R. Management of the upper extremity in traumatic tetraplegia. *Hand clinics*, 1988; 4:201-209.
- Musselman, K. E., Shah, M., & Zariffa, J. Rehabilitation technologies and interventions for individuals with spinal cord injury: translational potential of current trends. . *Journal of neuroengineering and rehabilitation*, 2018; 15(1):40.
- Nas, K., Yazmalar, L., Şah, V., Aydın, A., & Öneş, K. Rehabilitation of spinal cord injuries. *World journal of orthopedics*, 2015; 6(1):8-16.
- NationalSpinalCordInjuryStatisticalCenter. *Facts and Figures at a Glance*. Birmingham: University of Alabama at Birmingham, 2019.
- NINDS. *Spinal Cord Injury Information Page*. Retrieved 10 2, 2019, from National Institute of Neurological Disorders and Stroke: <https://www.ninds.nih.gov/Disorders/All-Disorders/Spinal-Cord-Injury-Information-Page>, 2019.
- Pacchierotti, C., Tirmizi, A., & Prattichizzo, D. Improving transparency in teleoperation by means of cutaneous tactile force feedback. *ACM Transactions on Applied Perception*, 2014; 11(1):4.
- Peckham, P., Keith, M., Kilgore, K., Grill, J., Wuolle, K., Thrope, G., . . . Wiegner, A. Efficacy of an Implanted Neuroprosthesis for Restoring Hand Grasp in Tetraplegia: A Multicenter Study. *Archives of Physical Medicine and Rehabilitation*, 2001; 82(10):1380-1388.



- Pehlivan, A., Lee, S., & O'Malley, M. K. Mechanical design of RiceWrist-S: A forearm-wrist exoskeleton for stroke and spinal cord injury rehabilitation. *4th IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob)*, 2012; 1573-1578.
- Peñasco-Martín, B., de los Reyes-Guzmán, A., Gil-Agudo, Á., Bernal, A., Pérez-Aguilar, B., & de la Peña, A. Aplicación de la realidad virtual en los aspectos motores de la neurorrehabilitación. *Revista de neurologia*, 2010; 51(08):481-488.
- Pessina, L.-A. *Artificial skin could help rehabilitation and enhance virtual reality*. Retrieved from Soft Robotics: <https://actu.epfl.ch/news/artificial-skin-could-help-rehabilitation-and-enha/>, 2019.
- Piggott, L., Wagner, S., & Ziat, M. Haptic Neurorehabilitation and Virtual Reality for Upper Limb Paralysis: A Review. *Critical Reviews in Biomedical Engineering*, 2016; 44.
- Pololu. *298:1 Micro Metal Gearmotor HPCB 12V with Extended Motor Shaft*. Retrieved from <https://www.pololu.com/product/3056>, 2019.
- Prasad, S., Aikat, R., Labani, S., & Khanna, N. Efficacy of Virtual Reality in Upper Limb Rehabilitation in Patients with Spinal Cord Injury: A Pilot Randomized Controlled Trial. *Asian spine journal*, 2018; 12(5):927-934.
- Rice, D., Faltynek, P., McIntyre, A., Mehta, S., Foulon, B., & Teasell, R. Upper Limb Rehabilitation Following Spinal Cord Injury. *Spinal Cord Injury Rehabilitation Evidence*, 2016; 1-121.
- Riener, R. Rehabilitation Robotics. *Foundations and Trends in Robotics*, 2013; 3(1-2):1-137.
- Salisbury, K., Conti, F., & Barbagli, F. Haptics Rendering: Introductory Concepts. *IEEE computer graphics and applications*, 2004; 24(2):24-32.
- Secciani, N., Bianchi, M., Ridolfi, A., Vannetti, F., Volpe, Y., Governi, L., . . . Allotta, B. Tailor-Made Hand Exoskeletons at the University of Florence: From Kinematics to Mechatronic Design. *Machines*, 2019; 7(2):22.

- Seidel, I. *Timer Library fully implemented for Arduino DUE*. Retrieved from GitHub: <https://github.com/ivanseidel/DueTimer>, 2018.
- Sheerin, F. Spinal cord injury: anatomy and physiology of the spinal cord. *Emerg Nurse*, 2004; 12(8):30-36.
- ShirleyRyanAbilityLab. *Rehabilitation Measures Database- Jebsen Hand Function Test and Box and Block Test*. Retrieved 10 15, 2019, from ShirleyRyan AbilityLab : <https://www.sralab.org/rehabilitation-measures>, 2012.
- Singh, A. *Spinal Cord Injury Levels*. Retrieved 10 2, 2019, from boneandspine: <https://boneandspine.com/spinal-cord-injury-levels/>, 2019.
- Singh, H., Unger, J., Zariffa, J., Pakosh, M., Jaglal, S., Craven, B., & Musselman, K. Robot-assisted upper extremity rehabilitation for cervical spinal cord injuries: a systematic scoping review. *Disability and rehabilitation. Assistive technology*, 2018; 13(7):704-715.
- Slosar, P. *Cervical Spinal Nerves*. Retrieved 10 5, 2019, from Spine-Health: <https://www.spine-health.com/conditions/spine-anatomy/cervical-spinal-nerves>, 2019.
- Snoek, G., IJzerman, M., Hermens, H., Maxwell, D., & Biering-Sorensen, F. Survey of the needs of patients with spinal cord injury: Impact and priority for improvement in hand function in tetraplegics. *Spinal cord*, 2004; 42:526-532.
- St. *X-NUCLEO-IHM04A1*. Retrieved from <https://www.st.com/en/ecosystems/x-nucleo-ihm04a1.html>, 2019.
- Thuret, S., Moon, L. D., & Gage, F. H. Therapeutic interventions after spinal cord injury. *Nature Reviews Neuroscience* , 2006; 7:628-643.
- Tieri, G., Morone, G., Paolucci, S., & Iosa, M. Virtual reality in cognitive and motor rehabilitation: facts, fiction and fallacies. *Expert Review of Medical Devices*, 2018; 15(2):107-117.
- Ultimaker. *Ultimaker Cura* . Retrieved from <https://ultimaker.com/it/software/ultimaker-cura>, 2019.
- Ultraleap. *LeapMotion*. Retrieved from <https://ultraleap.com/>, 2019.
- Unity. Retrieved from <https://unity.com/>, 2019.

- Vanmulken, D., Spooren, A., Bongers, H., & Seelen, H. Robot-assisted task-oriented upper extremity skill training in cervical spinal cord injury: a feasibility study. *Spinal Cord*, 2015; 53:547-551.
- Viau, A., Feldman, A., McFadyen, B., & Levin, M. Reaching in reality and virtual reality: A comparison of movement kinematics in healthy subjects and in adults with hemiparesis. *Journal of neuroengineering and rehabilitation*, 2005; 1:11.
- Wen, K., Neculescu, D., & Sasiadek, J. Haptic force control based on impedance/admittance control aided by visual feedback. *Multimedia Tools and Applications*, 2008; 37(1):39–52.
- WHO. *Spinal cord injury*. Retrieved 10 5, 2019, from WorldHealthOrganization: <https://www.who.int/news-room/fact-sheets/detail/spinal-cord-injury>, 2013.
- Wyndaele, M., & Wyndaele, J. Incidence, prevalence and epidemiology of spinal cord injury: what learns a worldwide literature survey? *Spinal Cord*, 2006; 44:523-529.
- Yeh, S. C. The Efficacy of a Haptic-Enhanced Virtual Reality System for Precision Grasp Acquisition in Stroke Rehabilitation. *Journal of healthcare engineering*, 2017.
- Yem, V., & Kajimoto, H. Wearable tactile device using mechanical and electrical stimulation for fingertip interaction with virtual world. *2017 IEEE Virtual Reality (VR)*, 2017; 99-104.
- Yeo, E., Chau, B., Chi, B., Ruckle, D. E., & Ta, P. Virtual Reality Neurorehabilitation for Mobility in Spinal Cord Injury: A Structured Review. *Innovations in clinical neuroscience*, 2019; 16(1-2):13-20.
- Yozbatiran, N., & Francisco, G. Robot-assisted Therapy for the Upper Limb after Cervical Spinal Cord Injury. *Physical Medicine and Rehabilitation Clinics of North America*, 2019; 30(2):367-384.