



Prototype of a low-cost open source prosthetic hand

*Relatório submetido à Universidade Federal de Santa Catarina como requisito para a
aprovação na disciplina
DAS5511: Projeto de Fim de Curso*

Martin Vincent Bloedorn

10 August 2015

Prototype of a low-cost open source prosthetic hand

Martin Vincent Bloedorn

ESTA MONOGRAFIA FOI JULGADA NO CONTEXTO DA DISCIPLINA

DAS5511: PROJETO DE FIM DE CURSO

E APROVADA NA SUA FORMA FINAL

CURSO DE ENGENHARIA DE CONTROLE E AUTOMAÇÃO

Prof. Hector Bessa Silveira

Assinatura do Orientador

Banca Examinadora:

Prof. Alexandre Trofino
ORIENTADOR NA EMPRESA

Prof. Hector Bessa Silveira
ORIENTADOR NO CURSO

AVALIADOR

DEBATEDORES

Resumo

Atualmente, as próteses mais avançadas para membros superiores usam diversas tecnologias de ponta para restaurar um pouco da funcionalidade perdida após uma amputação. No entanto, tais dispositivos podem custar dezenas de milhares de dólares, os tornando virtualmente inacessíveis sem o suporte de algum plano de saúde. Ainda pior, muitos dos usuários destes dispositivos os reportam como muito lentos ou muito pesados, apesar da tecnologia que embarcam. Nos últimos anos, muitos hobistas e inventores ao redor do mundo tem explorado o potencial da impressão 3D em criar próteses melhores e mais baratas. Este trabalho visa colaborar nesta busca. Nele, apresentar-se-á a especificação, projeto e implementação de um protótipo de uma mão prostética *open source* de baixo custo impressa em 3D. O protótipo proposto é testado e comparado a próteses líderes de mercado, atingindo um resultado satisfatório.

Abstract

Current cutting-edge upper limb prosthetic devices employ various advanced technologies to restore some of the functionality lost after an amputation. However, these devices may cost several thousands of dollars, making them virtually unaccessible without some sort of health-care funding. Even worse, many users report these devices as being too slow or too heavy, in spite of the technology they embed. In recent years, many hobbyists and inventors around the world have explored the power of 3D-printing in creating better and more affordable prosthetics. This work aims at collaborating to this pursuit. It presents the specification, design and implementation of the prototype of an *open source*, low-cost 3D-printed prosthetic hand. The proposed prototype is tested and compared to market-leading prosthetic devices, reaching a satisfactory performance.

Contents

Contents	viii
List of Figures	xi
Standard Terms	xvi
Abbreviations	xix
1 Introduction	1
1.1 Background and motivation	1
1.2 Goals and achieved results	3
1.3 Structure of this work	3
2 Anatomy and Analysis of the Human Hand	5
2.1 Physiology of the hand	5
2.1.1 Regions and bones	5
2.1.2 Motion and muscles	6
2.2 Characterization of the hand	8
2.2.1 Basic modeling	8
2.2.2 Activities and grips	10
2.3 Types of upper-limb amputations	11
3 Prosthesis Requirements	15
3.1 State-of-the-art	16
3.1.1 Commercial products	16
3.1.1.1 Bebionic3	16
3.1.1.2 i-Limb	17
3.1.1.3 Michelangelo Hand	18
3.1.2 Open source developments	19
3.1.2.1 InMoov	21
3.1.2.2 Open Bionics	22
3.2 Discussion on current designs	23
3.2.1 Weights and sizes	23
3.2.2 Finger mechanisms and actuation	24

3.2.3 General usability	25
3.3 Technical requirements	28
4 Prototype Design	31
4.1 General specifications	31
4.1.1 A robotic hand	32
4.1.2 Manufacturing	32
4.1.3 Power source	32
4.2 Mechanical design	32
4.2.1 Actuator selection	33
4.2.2 Finger mechanism design	34
4.2.3 Issues with backdrivability	37
4.2.4 Designing a linear actuator	38
4.2.5 Modeling the fingers and the palm	41
4.2.6 Thumb design and modeling	44
4.2.7 Completing the model	45
4.3 Electrical design	47
4.3.1 Motor drivers	47
4.3.2 Microcontroller	51
4.4 Software outline	52
4.4.1 On threads and microcontrollers	52
4.4.2 Development environment	54
4.4.3 Template Structure	55
5 The Prototype and Overall Results	59
5.1 Printing and assembling the prototype	59
5.1.1 3D-printing and part orientation	59
5.1.2 The build process	61
5.1.3 Electronics	62
5.2 Overall results of preliminary tests	64
5.2.1 Weight, speeds and strengths	64
5.2.2 Grasps	66
5.2.3 Final cost	66
5.2.4 Overall evaluation of the prototype	67
6 Conclusions	69
6.1 Future works	70
Appendices	
Appendix A Designing and Building a 3D-Printer	73
A.1 Motivation	73
A.2 Mechanical design	73
A.3 Electronics setup	75
A.3.1 Choice of actuators	75

CONTENTS

x

A.3.2 Control board and motor drivers	76
A.4 Software setup and results	77
Bibliography	79

List of Figures

1.1 <i>Right:</i> cosmetic prosthesis. Offers only aesthetic functionality. <i>Left:</i> split hook. Opened/closed by a cable fixed to a body harness. Source: [Wikipedia 2015d]	2
2.1 Main bone groups in the hand. Source: [Wikipedia 2015c]	7
2.2 Movements of a finger. Source: [Kang Li et al. 2011]	8
2.3 Movements of the thumb ¹	8
2.4 Detailed view of index finger's structures. Source: [Bundhoo et al. 2005]	9
2.5 Kinematic chain for a finger. Source: [Bundhoo et al. 2005]	9
2.6 Hand grasp taxonomy, with 33 different patterns. Source: [Feix et al. 2009]	11
2.7 Reduced grip list. (a1) palmar prehension (tripod), (a2) palmar prehension (pinch), (b) tip prehension, (c) lateral prehension, (d) hook prehension, (e) spherical (power) prehension, (f) cylindrical prehension. Source: [Feix et al. 2009]	12
2.8 Main classifications for upper-limb amputations.	13
3.1 <i>Bebionic3</i> from manufacturer <i>RSLSteeper</i> ²	17
3.2 <i>Touch Bionics' iLimb Ultra</i> . Source: [Touch Bionics 2014]	18
3.3 <i>Touch Bionics' iLimb Digits</i> , an alternative for patients with partial hand amputation.	19
3.4 <i>Michelangelo</i> hand from <i>Ottobock</i> . Hand without the rubber cover on the left; covered on the right.	20
3.5 Boy wearing a <i>Robohand</i> .	21
3.6 <i>InMoov's</i> forearm and hand. The motors are housed in the forearm, and cables running up to the hand actuate on the fingers and thumb.	22
3.7 Fully assembled <i>InMoov</i> robot.	22
3.8 Assembled <i>InMoov Prosthetic Hand</i> .	23
3.9 <i>Dextrus hand</i> , first version.	23
3.10 <i>Dextrus hand V2</i> .	23
3.11 Finger mechanisms for: a) the <i>i-limb</i> , b) the <i>Bebionic</i> , c) the <i>Michelangelo hand</i> . θ_1 and θ_2 are the angles of the Metacarpophalangeal (MCP) and Proximal Interphalangeal (PIP) joints, respectively. Source: [Belter et al. 2013]	26
3.12 <i>Dextrus'</i> cable-driven finger mechanism, frontal (top) and lateral (bottom) views.	27
3.13 Exploded view of the <i>InMoov Prosthetic Hand's</i> finger. Source: [Langevin 2015]	27

3.14 Behavior of a <i>rigid</i> (top) and an <i>adaptive</i> (bottom) grasp . Source: [Open Bionics 2015]	27
3.15 Bottom view of the <i>Bebionic</i> 's thumb. The dashed line originates on the thumb's circumduction axis and highlights its plane of motion. Source: [Belter et al. 2013]	28
3.16 Preferred operating plane for a single Degrees of Freedom (DOF) thumb. Source: [Weir 2003]	28
4.1 <i>Pololu's Metal Gearmotor</i> ³	34
4.2 Example of simple 4-bar mechanism.	35
4.3 (a)	36
4.4 (b)	36
4.5 (c)	36
4.3 Finger poses used to define the fingers precision points. (a) is a pointing position, (b) a finger hook, and (c) represents a fully flexed finger (the distal phalanx touching the palm). \vec{u} is normal to the hand's palm.	36
4.4 Finger's mechanism. Arc c describes the output trajectory.	37
4.5 MN lever and simplified forces in cylindrical grasp	37
4.6 <i>Left</i> : a standard RC servo . <i>Right</i> : a servo with its top lid removed, exposing the gearing.	38
4.7 A disassembled RC servo . 1) Position feedback potentiometer. 2) Drive and control electronics. 3) Direct Current (DC) motor. 4,5,6,7) Reduction gears. 8) Bearings and metal dowels.	39
4.8 <i>Top left</i> : Second gear in a servo 's reduction. <i>Bottom left</i> : Removing the small top gear. <i>Right</i> : Attaching a M3 screw.	40
4.9 Computer-Aided Design (CAD) model of the custom linear actuator. One of the parts was set transparent to make the internal gear arrangement visible.	41
4.10 Exploded view of the designed linear actuator. On the far right, a part can be seen, that holds a M3 nut and gets displaced according to the screw's rotation.	41
4.11 Tiling shape of the linear actuators allow compact side-by-side positioning (a few parts were suppressed for better visualization).	42
4.12 Initial model of the index finger and knuckles made with the open source FreeCAD	42
4.13 Model of index finger, coupled to the linear actuator . The proximal phalanx was set transparent for better visualization of the finger's mechanism.	42
4.14 Linkage between the linear actuator and the proximal phalanx highlighted in red.	43
4.15 Finger reaching the poses defined during the mechanism design phase. <i>Left</i> : Finger in hook position. <i>Right</i> : Finger fully flexed.	43
4.16 Palm's back plate with attached fingers and linear actuators . The fixed abduction angle provides a more natural look, and enables thin objects to be held when transitioning from fully extended to fully flexed fingers. The palm is 90mm at its tallest, and 90mm at its widest, dimensions similar to the <i>bebionic</i> 's large model.	44

4.17 Hand's bottom view. The highlighted z axis is normal to the palm and centered on the middle finger, and the x axis is parallel to the internal palm surface. The thumb's circumduction axis runs through point T_A and is perpendicular to the zx -plane.	45
4.18 Designed thumb. 1) part that attaches the whole thumb to the palm and encloses the <i>micro servo</i> . 2) part that holds the linear actuator and the finger segment. 3) single fixed distal finger segment. 4) enclosed variation of the linear actuator — its top cover was set transparent for visualization of the internal gearing.	46
4.19 Modeled prototype in all the poses previously defined in Figure 2.7. A1) Palmar (tripod) prehension. A2) Palmar (pinch) prehension. B) Tip prehension. C) Lateral grip. D) Hook grip. E) Spherical grip (partially achieved due to the lack of active finger abduction). F) Cylindrical (power) grip.	46
4.20 Theoretical H-bridge arrangement. Source: [Wikipedia 2015b]	48
4.21 A signal with pulse-width modulation . The signal is on during D percent of a fixed T_p time period (D is known as the <i>duty cycle</i>). As a result, an inductive load "sees" an mean voltage u_{pwm}	48
4.22 Controlling a servo with a Pulse-Width Modulation (PWM) signal.	49
4.23 A few Integrated Circuit (IC) packaging types. <i>Left:</i> PQFN package (top and bottom views). <i>Center:</i> HTSSOP package (top and bottom views). <i>Right:</i> 64-pin LQFP package.	50
4.24 Minimal schematic for the DRV8801. The <i>enable1</i> input receives a PWM signal, varying the motor's rotation speed. <i>phase1</i> defines the direction of rotation. <i>vprop1</i> outputs a voltage proportional to the current in the IC's internal <i>H-bridge</i>	50
4.25 Schematic of the STM32F103 and its connections.	52
4.26 The board after routing. <i>Left:</i> Top view of the board, showing the processor and two DRV8801. <i>Right:</i> Bottom view of the board; the remaining three DRV8801 can be seen.	53
4.27 Setup for programming and debugging of the processor. The <i>host computer</i> uses a Universal Serial Bus (USB) <i>probe</i> that connects it via Single-Wire Debug (SWD) or Joint Test Action Group (JTAG) to the <i>target processor</i> . All queries executed by the probe are coordinated by <i>OpenOCD</i> , a software being executed on the <i>host computer</i>	55
4.28 The <i>STM32 Nucleo F103RB</i> , a development board for the <i>STM32F103</i> microcontroller ⁴	56
4.29 Structure of the developed software template.	57
 5.1 Commercial Fused Deposition Modeling (FDM) printer in action. The filament in the spool is pushed by the extruder into a hot nozzle. The extruded plastic is deposited to form the desired part.	60
5.2 <i>Left:</i> Test parts were printed in six different orientations. <i>Right:</i> Each part was then subjected to a destructive stress test. Source: [Sparx Engineering 2015] . .	61

5.3	The proximal phalanx of the index, as well as all phalanges in each digit are printed laterally, as in the image. Thus, the forces applied at the PIP and MCP joints, and at the coupling of the linear actuator (all three represented by the dashed lines) get distributed along the print layers.	61
5.4	<i>Left:</i> Distal segment and proximal phalanx of the index finger being prepared for 3D-printing in specialized software. Phalanges are printed parallel to the print bed; the infill pattern is purposely exposed for the picture. <i>Right:</i> Distal segment of the thumb — a complete and a failed part. The rectilinear infill is visible.	62
5.5	<i>Left:</i> Various iterations of the linear actuator 's <i>knuckle</i> parts and the phalanges until the first working finger version, on the far right (mounted on the <i>finger tester</i>). <i>Right:</i> The <i>finger tester</i> ; and <i>Arduino</i> driving a commercial DRV8801 board, powered by a 7.4V Lithium-Polymer (LiPo) battery.	62
5.6	First assembled finger and linear actuator . The square potentiometer connected to the finger's back link can be seen.	63
5.7	Detailed view of the thumb's enclosed linear actuator	63
5.8	Lateral view of the assembled thumb. The square potentiometer attached to the Thumb's Interphalangeal (TIP) joint is visible.	64
5.9	Digits attached to the palm's back plate. Back and front views.	64
5.10	<i>Left:</i> Breadboard arrangement while testing the microcontroller. An <i>Arduino</i> (top left corner) is only used to supply 3.3V to the setup, and a commercial SWD probe (white board, bottom) programs and debugs the microcontroller. <i>Right, bottom:</i> Routed board with a minimal circuit for the DRV8801. <i>Right, top:</i> Functional board with soldered components.	65
5.11	The prototype is placed in a power grip , and statically loaded with 40N. The test was completed with no damages.	66
5.12	Tested grips . A) Lateral prehension, holding a bank card. B) Palmar prehension (<i>pinch</i>), holding a coin. C) Cylindrical grip , holding a water bottle. D) Securing an object between the fingers thanks to their abduction angle. E) Palmar prehension (<i>tripod</i>) pose.	67
A.1	<i>Left:</i> CAD rendering of the custom designed 3D-printer, with highlighted axis directions. <i>Right:</i> Assembled and functional printer.	75
A.2	A commercial 3D-printer, the <i>Prusa I3</i> , using the same cartesian arrangement of the custom designed machine.	75
A.3	<i>Left:</i> Disassembled stepper motor. The coils and around the spinning core can be easily seen. <i>Right:</i> Schematic view of the coils in the motor.	77

A.4 Electrical connections of the elements in the 3D-printer. In the center, the <i>RAMPS</i> board, connected to an <i>Arduino Mega</i> . The <i>RAMPS</i> drives the three <i>Powerlolu</i> boards and a single DRV8825 (for the extruder's motor), on the right. The extruder's hot end is displayed in the bottom-center of the image. On the top-left corner, the heated bed and a thermistor (that reads the bed's temperature) is depicted. The whole system runs off a 12V power supply, shown in the bottom-left corner of the image.	78
A.5 Assembled electronics in a wall-mounted enclosure.	78
A.6 One of the parts produced with the custom 3D-printer.	78

Standard Terms

Commonly used terms and names will have clearly defined meanings in order to avoid ambiguities along the development of this work. Those terms are summed up in the following glossary — and will show up in **bold** along the text.

Back link part

Refers to the part in the developed prototype. Is the rod that runs through the device's **proximal phalanx** up to the distal segment, coupling the **PIP** joint to the **MCP**. Presented in Subsection 4.2.5.

Carpal bones

Group of bones that articulates the wrist with the **metacarpals**.

Compliance

The extent to which an object will deform in response to an applied force.

Digit

Generic naming for any of the four fingers or the thumb.

Distal phalanx

Bone that articulates with the **proximal phalanx** in the finger.

Grasp

Every static hand posture with which an object can be held securely with one hand.

Grip

See: **grasp**.

Intermediate phalanx

Bone that articulates with the **proximal phalanx** and the **distal phalanx**.

Knuckle part

Refers to the part in the developed prototype, that connects the **linear actuator** to the device's **proximal phalanx**. It implements the **MCP** joint and holds the finger feedback potentiometer. Presented in Subsection 4.2.5.

Linear actuator

Refers to the linear actuator designed and developed in this work (detailed in Subsection 4.2.4).

Maker

Movement of professional and hobbyist individuals interested in engineering-related topics, such as robotics, 3D-printing, electronics, and others.

Metacarpal bones

Group of bones that connects the **carpals** and forms the main portion of the hand's palm.

Myoelectric

Denotes a device/system that relies on **Electromyography (EMG)** signals.

Open hardware

The **open source** philosophy oriented at hardware development.

Open source

Development model promotes a universal access via a free license to a product's design or blueprint, and universal redistribution of that design or blueprint, including subsequent improvements to it by anyone. This definition is generally applied to software products.

Power grip

Cylindrical, spherical and hook grasping patterns (figure 2.7). See also: **grasp**.

Precision grip

Palmar (both tripod and pinch), tip and lateral grasping patterns (figure 2.7). See also: **grasp**.

Proximal phalanx

First bone in the base of the finger.

RC Servo

Radio Controlled (RC) servo motor. See: **servo**

Residual limb

See: **stump**

Servo

Servo motor, usually employed in radio-controlled planes and vehicles. Features a DC motor, a mechanical reduction, a feedback potentiometer and driving electronics in single enclosure. An external analog signal serves as setpoint, and the device uses an internal **PID** loop to control the position of its output shaft. See: **RC servo**

Socket

Custom-made, removable structure mounted on an amputee's **stump**, coupling it to the prosthetic device.

Stump

Distal end of a limb left after amputation, also called **residual limb**.

Thumb's circumduction axis

Axis around which the thumb rotates in an artificial limb. Aims at simplifying the **Thumb's Carpometacarpal (TCM)** motion.

Transradial amputation

Amputation between the elbow and the wrist, through the radius bone.

Wrist disarticulation

Hand amputation along the wrist. Leaves a mostly undamaged forearm.

Abbreviations

Acrylonitrile Butadiene Styrene (ABS)

Common thermoplastic polymer (and "engineering plastic") also used in *Lego®* bricks.

Activites of Daily Living (ADL)

Basic daily self-care, mobility and environmental activities executed by the average health person.

Analog-to-Digital Converter (ADC)

Carpometacarpal (CMC)

The joint(s) between the **carpals** and **metacarpals**.

Computer-Aided Design (CAD)

Computerized Numeric Control (CNC)

Degrees of Freedom (DOF)

Number of independent parameters needed to specify the configuration of a system.

Direct Current (DC)

Distal Interphalangeal (DIP)

The joint(s) between the **intermediate phalanx** and **distal phalanx**.

Do-it-yourself (DIY)

Electrocecephalography (EEG)

Electrodiagnostic medicine technique for measuring, evaluating and recording the electrical activity produced by brain activity.

Electromyography (EMG)

Electrodiagnostic medicine technique for measuring, evaluating and recording the electrical activity produced by skeletal muscles.

Electronics Design Automation (EDA)**Fused Deposition Modeling (FDM)**

3D-printing process. Composes objects by successive deposition of plastic layers.

General-Purpose Input/Output (GPIO)**Inertial Measurement Unit (IMU)****Integrated Circuit (IC)****Integrated Development Environment (IDE)****Joint Test Action Group (JTAG)****Light-Emitting Diode (LED)****Lithium-Polymer (LiPo)****Metacarpophalangeal (MCP)**

The joint(s) between the **metacarpal** and **proximal phalanx**.

Non-Backdrivable Mechanism (NBDM)**Proportional-Integral-Derivative (PID)**

Feedback control structure.

Proximal Interphalangeal (PIP)

The joint(s) between the **proximal phalanx** and **intermediate phalanx**.

Pulse-Width Modulation (PWM)**Radio Controlled (RC)****Real-Time Operating System (RTOS)****Single-Wire Debug (SWD)****Thumb's Carpometacarpal (TCM)**

Carpometacarpal joint at the base of the thumb's **proximal phalanx**.

Thumb's Interphalangeal (TIP)

Joint between the thumb's proximal and distal phalanges. In other works, is generally referred to simply as Interphalangeal (IP) joint.

United States (US)**Universal Serial Bus (USB)**

1

Introduction

1.1 Background and motivation

Many aspects of everyday life may easily be taken for granted, and the functionality of human hands is one of such cases. Every year, more than 50,000 amputations occur in the United States, of which 25% are of upper limbs [McGimpsey et al. 2009]. These people are forced to engage in new habits and to relearn how to perform various daily tasks — a process that is difficult and ponderous. Dedicated medical and psychological care is required during adaptation to this new reality (occasionally, even after), and some activities may become permanently inviable.

The story of humanity is filled with wars and diseases that made limb losses an unfortunate frequent issue. The will to replace some of the lost function — or simply to conceal the damaged extremity — has been present for millennia. The first use of a prosthetic device (from the greek προτηγέσεις, *prothéseis*, "addition, application, attachment") is attributed to a wooden toe, built and worn in ancient Egypt [Wikipedia 2015d].

While technology in general has been vastly improved since this first reported device, modern day prosthetics still fail to offer lifelike functionality. Nowadays, more than 30% of all upper-limb amputees do not regularly wear a prosthesis of any kind, and at least 50% of them opt for simple terminal devices, such as cosmetic gloves or mechanical hooks, as shown in Figure 1.1 [ACA 2011]. The latter, known as *split hooks*, are still regarded by many amputees as one of the most functional prostheses, despite being a technology developed in the Victorian era — more than 120 years ago.

Current cutting-edge technology is employed in the research of **myoelectric** artificial limbs. These robotic hands interpret electric muscle signals off the amputee's **residual limb**, triggering various **grip** motions. Albeit using the most advanced technology available, most users report them as too heavy and too slow [Pylatiuk et al. 2007]. On top of that, such devices usually cost several dozens of thousands of dollars. Lacking the money, the majority of people turn to health-care reimbursements. These generally limit their collaborations to few thousand dollars, or to a single prosthesis during a person's lifetime [McGimpsey et al. 2009]. Both issues are very alarming, specially when considering that most prosthetics are designed to last 3 to 5 years [HSS 2015].



Figure 1.1: *Right:* cosmetic prosthesis. Offers only aesthetic functionality. *Left:* split hook. Opened/closed by a cable fixed to a body harness. Source: [Wikipedia 2015d]

Fitting a prosthesis to its user is another challenge. As each person has different sized limbs, **residual limbs** may have severely different shapes across patients. To couple a terminal device to it, a custom **socket** has to be built — an expensive process that has to be executed by a trained professional. Furthermore, the device obviously needs to have a suitable size for the patient. With most of the recent research in prosthetics directed at wounded veteran soldiers (the majority of which are men), female amputees still face difficulty when searching for adequately sized prostheses. Children with malformations or limb losses are at an even worse stand: since they quickly outgrow a prosthesis, fitting them with any device at all is oftentimes simply not feasible.

With no apparent relation to these issues is the technology of 3D-printing, developed for industrial applications in the early 1980s. In the last decade, the expiration of major patents in the field allowed it to rapidly expand to the consumer market. Nowadays, desktop 3D-printers can be purchased for as low as U\$400. Most of these consumer-grade devices operate using the **Fused Deposition Modeling (FDM)** technique, in which a moving printhead squirts molten plastic into layers, forming a desired object. Thanks to this working principle, complex parts can be created almost trivially — requiring less effort, resources and technical knowledge than regular manufacturing processes do. This powerful and affordable tool fueled creations in communities of hobbyist and professional tinkerers and hackers — the self-entitled **maker** community. This once small group of individuals grew, in the last decade, to a very large community that shares designs and knowledge online, in a typical **open source** setting. **Makers** have already designed a myriad of successful and affordable projects, from home-automation equipment to drones and functional satellites.

The potential of 3D-printing in creating cheaper, custom-sized artificial limbs was soon recognized. It did not take long until members of the **maker** community started directing efforts at tackling the aforementioned issues with current prosthetics. Various hobbyists started developing 3D-printed artificial hands with promising functionalities, some of them also using **myoelectric** technology. Colorful, 3D-printed mechanical grippers were developed and custom fitted for hundreds of children with congenital hand malformations. All of those projects are freely shared, and can be built at a fraction of the cost of commercial devices.

After coming across the difficulties and costs still associated with commercial prosthetics, it seems clear that this is a good moment for contributing to these **open source** initiatives. The majority of ongoing **maker** projects are developed by individuals with little background

in engineering or medicine and, while the results are already impressive, major improvements are still possible. The motivation to improve upon these low-cost, **open source** prosthetic hand designs is the foundation of this work.

1.2 Goals and achieved results

Following the outlined motivation, the goal of this work is to *develop the prototype of a low-cost open source 3D-printed prosthetic hand*. This project intends to pursue a more formal approach than current **maker** initiatives, attempting to follow some basic medic concepts for prosthesis design while exploring the capabilities of consumer-grade 3D-printers. Furthermore, the prototype to be designed must be as affordable as other **open source** devices, while still withstanding (less intensive) daily activities.

These defined goals were successfully met, as described in this document. After familiarization with basic anatomical concepts of the human hand, the current state-of-the-art in prosthetics — in both commercial and **open source** fields — is explored. Based on these informations, relevant requirements for the proposed prototype are defined. Drawing inspiration from similar devices, a 4-bar linkage mechanism is synthesized for the prototype's fingers. A custom linear actuator is then conceived and tested to suit the project's needs. Since 3D-printing is chosen as the manufacturing method for the prototype, a printer was required. Lacking a supplier that offered a cost-effective machine, a custom 3D-printer is designed and built for this project.

The modeled prototype is printed and assembled using only the custom 3D-printer and off-the-shelf tools and components. After assemblage, the device is tested, successfully reaching a minimal set of **grip** poses needed for everyday activities. The prototype has a cost 200 times inferior to the average commercial device, being even cheaper than similar **open source** artificial hands. When compared to one of the leading commercial prosthetics, the prototype can exert over 80% of that device's maximum force, moving fingers at more than 90% of its maximum speed, while being almost 40% lighter. It also undergoes a static loading test of 40 N with no damages. Lastly, completing the **open source** nature of the project, all the files and documentation needed to replicate and modify the prototype are available online¹.

1.3 Structure of this work

This document begins by familiarizing the reader with basic aspects and nomenclature of the human hand in *Chapter 2*. The main joints, bones and muscles are presented, and aspects of the hand's motion are discussed. The chapter then presents a more systematized understanding of the hand's functions: daily activities and **grip** actions associated to them are briefly addressed. A list of the minimal set of poses required by such activities is presented — upon which much of this work is based. The chapter concludes with a brief exposition of the main types of upper-limb amputations.

¹ Available at <https://github.com/MartinBloedorn/biohand>

Chapter 3 opens up with a concise discussion about the current prosthetics research, then presenting its state-of-the-art. Some of the most advanced commercial hand prosthetics are exhibited, as well as some of the most relevant **open source** projects. The design choices in these devices is subsequently discussed, and technical requirements for this work's prototype are synthesized.

The most relevant design steps of the prototype are presented in *Chapter 4*. After a brief discussion on some design specifications, the prototype's mechanical and electrical elements are thoroughly detailed. At the end of this chapter, the software structure of the project is analyzed. Its current state is discussed and the used development tools are introduced.

Chapter 5 addresses the printing, assemblage and tests of the final prototype. Its performance in comparison to existing commercial and **open source** devices is addressed, and its final cost is calculated. *Chapter 6* concludes the document, with a discussion on this work's results, implications, and future activities.

Lastly, *Appendix A* presents the 3D-printer designed and implemented for this work. Its mechanical and electrical design are debated, and its software setup is briefly addressed.

2

Anatomy and Analysis of the Human Hand

The human hand is a very capable instrument. Shaped by millions of years of evolution, it carries a symbolical relationship to the very notion of being human. As it could have been expected, a hand is an intricate piece of biological machinery, with 27 major bones, over 30 muscles and at least 27 **degrees of freedom (DOFs)** [Weir 2003].

In the effort of artificially approaching such capable tool, knowledge about its working and constitution is mandatory. With that in mind, this chapter aims to supply basic anatomical concepts and nomenclature on the hand's structures. This knowledge will be relevant when specifying and designing the proposed prototype.

The following chapter is divided into three sections. Section 2.1 presents relevant naming and information on the hand's regions, bones, joints and types of motions. Section 2.2 discusses how the fingers and the thumb can be modeled as kinematic chains, and how one can classify the main prehension motions of the hand. Finally, Section 2.3 introduces some basic concepts on the types of upper limb amputations.

2.1 Physiology of the hand

This section presents basic concepts and naming of the human hand. Initially, the hand's principal regions are described and named, and the main bones and joints are exhibited and discussed. In the following, relevant concepts about the hand's musculature are addressed, and a more detailed example of the structures in an index finger supports part of the discussion in the next section.

2.1.1 Regions and bones

Initially, one can intuitively identify the major structures on the human hand: the palm and five **digits** — four fingers (the index, middle, ring and little fingers) and one thumb. The palm can be further subdivided into three regions:

- *Volar*: Anterior (internal) part of the palm.
- *Dorsal*: Also known as *opisthenar* region, is the corresponding posterior part of the hand.
- *Heel of the hand*: Basal region of the palm, above the wrist. Supports the most pressure when the hand is loaded.

In average, the human hand weights 400g — about 0.6% of the body mass for men and 0.5% for women [Belter et al. 2013]. It consists of 27 major bones approximately arranged in 5 groups, as shown in Figure 2.1. Those comprise:

- The **Carpals**: first two rows of bones in the hand, that articulate the wrist with the **metacarpals**. The joints between those groups are known as the **carpometacarpals**. Along the text, the **Thumb's Carpometacarpal (TCM)** will receive special focus, and will thus be represented by a specific abbreviation.
- The **Metacarpals**: main structural bones of the palm. They articulate with the **digits' proximal phalanges**. The hinge between the two, informally referred to as the knuckles, forms the **Metacarpophalangeal (MCP)** joints.
- The **Proximal phalanges**: in the fingers, they articulate with the **intermediate phalanges**, forming the **Proximal Interphalangeal (PIP)** joints. The thumb, however, lacks a intermediate phalanx, so the proximal one articulates directly to the **distal phalanx**, forming the **Thumb's Interphalangeal (TIP)** joint.
- The **Intermediate phalanges**: intermediate bony section in the fingers; articulates with the **distal phalanges** forming the **Distal Interphalangeal (DIP)** joint. Both structures are not present in the thumb.
- The **Distal phalanges**: smallest of the phalanges, located at the end of each **digit**.

2.1.2 Motion and muscles

These numerous joints, combined with a complex network of muscles and tendons, enable the hand's great mobility. Before briefly exploring these nets, it is meaningful to introduce some terms associated with the hand's motion. For the fingers, *flexion* and *extension* are the motions colloquially named as closing and opening, respectively. *Abduction* corresponds to laterally spreading the finger out, away from its resting position; *adduction*, in turn, means returning the finger to it. Both sets of motions are represented in Figure 2.2. In addition to these motions, the thumb is also capable of *opposition* — the movement generally involved in pinching, as seen in Figure 2.3.

Associated with these motions, are two main sets of muscles. The *extrinsic* muscles extend from the forearm to the hand, and are responsible for crude movements and weight lifting. They are the main components in the flexion and extension of the **digits**, along with wrist movements. The *intrinsic* muscles are located within the hand itself, and are responsible for finer movements, as well as finger abduction and adduction.

¹Source: http://brooksidepress.org/basic_patient_care/lessons/

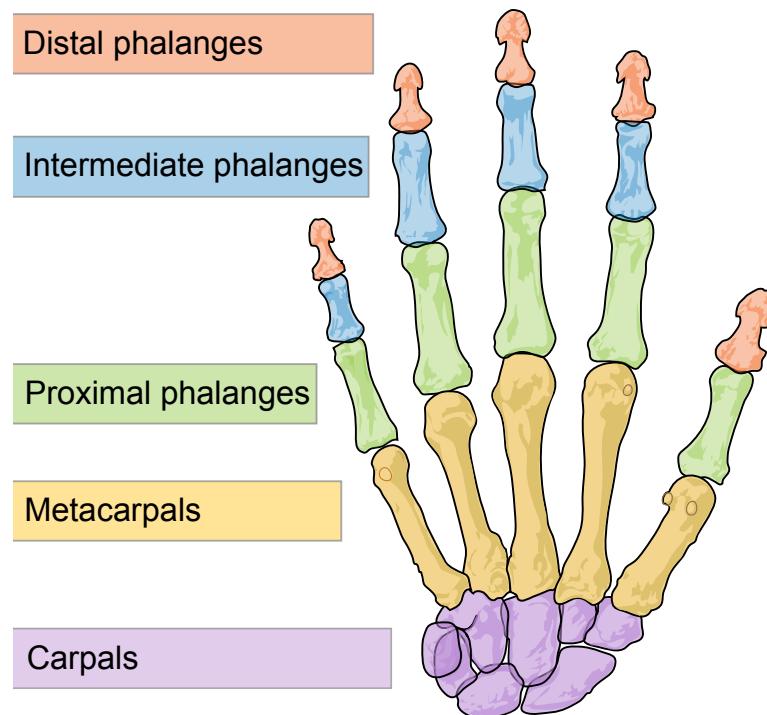


Figure 2.1: Main bone groups in the hand. Source: [Wikipedia 2015c]

As an illustration, we will further analyze the structures in the index finger, depicted in Figure 2.4. The proximal, medial and distal phalanges are highlighted, together with the **MCP**, **PIP** and **DIP** joints and the **metacarpal** bone in the palm. The *extensor indicis proprius* (EIP) and the *extensor digitorum communis* (EDC) are two main extrinsic extensor muscles. Their tendons attach to the finger's *extensor hood*, a collagen-based sheath that enables smooth motion of the joints. Also attached to this hood are three intrinsic muscles — two *interosseous* (IOs) and one *lumbrical* (LU). These muscles assist both flexion and extension, and generate adduction and abduction motions. Lastly, the main drivers of the finger's flexion are the *flexor digitorum superficialis* (FDS) and the *flexor digitorum profundus* (FDP) tendons, each one connected to its respective homonymous muscle. The FDS attaches to the **PIP** joint, while the FDP attaches simultaneously the **PIP** and the **DIP** joints.

Even though the other fingers are operated by different sets of muscles and tendons, the mechanism in each is very similar to the index'. However, further naming of such sets is of little relevance for this work, and shall not be introduced. Similarly, considerations about other structures (e.g. nerves, skin) also lie beyond the scope of this document. The interested reader is referred to [Sobotta et al. 2006].

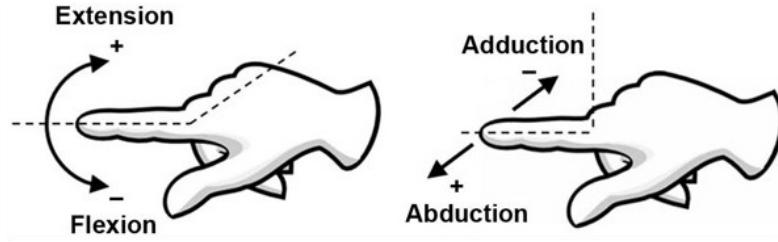


Figure 2.2: Movements of a finger. Source: [Kang Li et al. 2011]

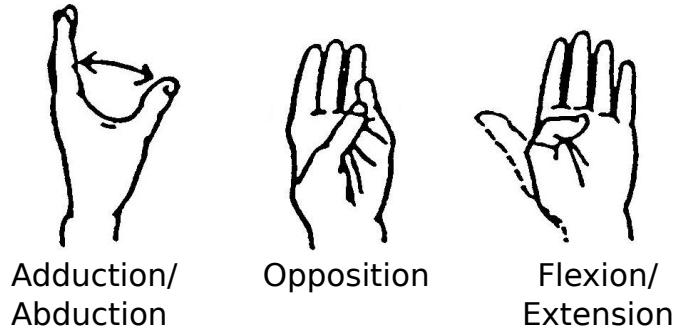


Figure 2.3: Movements of the thumb¹.

2.2 Characterization of the hand

This section further approaches the hand's **digits** and functionalities, establishing concepts that are used throughout this work. In the fist subsection, a descriptive model for the fingers and the thumb is presented. In the sequel, a minimum set of hand prehensile motions (required by everyday activities) is discussed.

2.2.1 Basic modeling

The structures of the index finger, analyzed in Subsection 2.1.2, enable a more detailed comprehension of the kinematic chains of each **digit**. In each finger, the **MCP** joint provides 2 **DOFs** — essentially a ball joint, enabling both flexion/extension and adduction/abduction. The **PIP** joint collaborates with another **DOF** of extension/flexion. The **DIP** joint however, is a passive **DOF**. As previously pointed out, the characteristics of the finger's extensor hood and the insertion of the tendon in the **DIP** joint keep its motion mostly coupled to the **PIP** joint [Bundhoo et al. 2005]. Such kinematic structure of the finger is represented in Figure 2.5.

The thumb, moreover, presents 4 active **DOFs**. Two of them are provided by the **TCM** ball-like joint, one by the thumb's **MCP** joint, and another one by the **TIP** joint.

In conclusion, the five **digits** together with the **Carpometacarpal (CMC)** joints in the palm result in at least 27 **DOFs** for the whole hand [Weir 2003]. A complete kinematic model of the hand (including, for example, the bones in the palm) will not be necessary for this work.

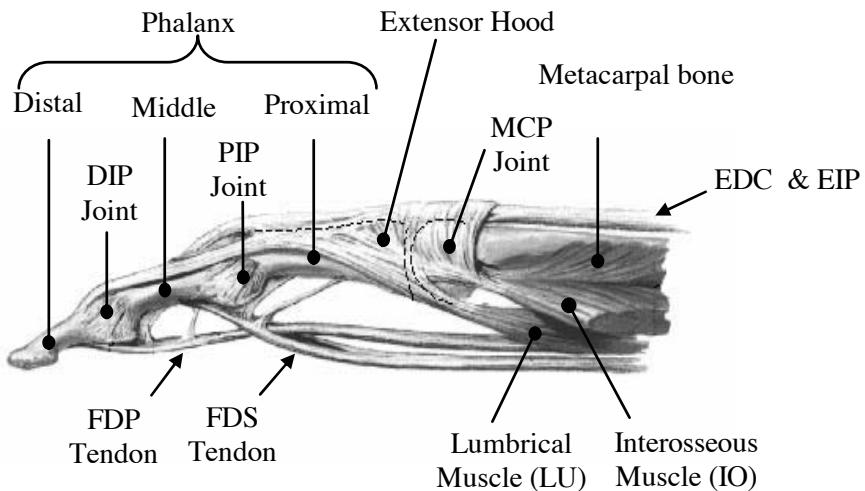


Figure 2.4: Detailed view of index finger's structures. Source: [Bundhoo et al. 2005]

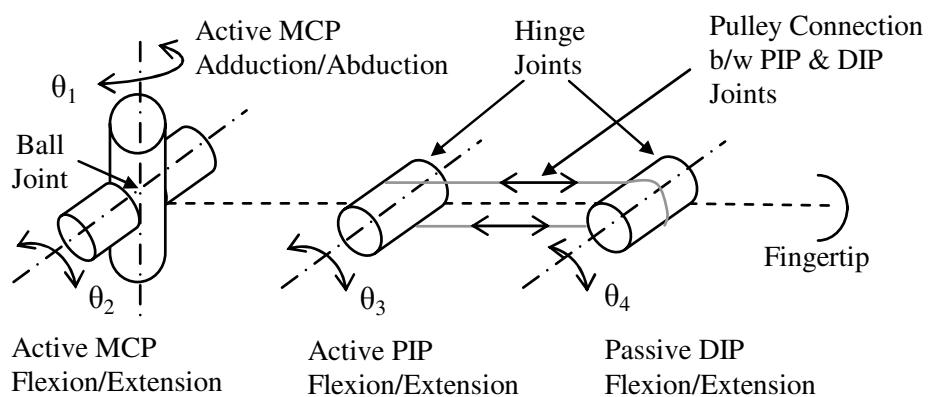


Figure 2.5: Kinematic chain for a finger. Source: [Bundhoo et al. 2005]

2.2.2 Activities and grips

All the aforementioned characteristics make the hand a very versatile tool. In order to assess the hand's functionality across various individuals, one can explore the definition of the **activities of daily living (ADLs)**. These encompass basic daily self-care, mobility and environmental activities executed by the average healthy person. They can be separated in two areas: basic (BADLs) and instrumental (IADLs). BADLs require skills to take care of one's own body, and include tasks such as bathing, eating, dressing/undressing, toilet hygiene, bowel and bladder management, among others. IADLs are more advanced, and require the manipulation of tools or instruments. These include housework, preparing meals, taking medications, use of telephone or other form of communication, transportation within a community, and a few more [Pendleton et al. 2012] [Wikipedia 2015a]. **ADLs** are very useful in healthcare management, enabling, for example, consistent evaluation of the functionality of prosthetic devices, or assessing the disability degree of a patient with paralysis [Sollerman et al. 1995].

It is intuitively clear that the hand plays a major role in such tasks. To comprehend the hand's usage during these activities, one may analyze the grasping patterns it assumes during them. As defined in [Feix et al. 2009], a **grasp** is every static hand posture with which an object can be held securely with one hand². The authors compared several literature sources to develop a comprehensive taxonomy table with 33 patterns, shown in Figure 2.6. In it, the columns are firstly organized by power or precision characteristics. Next, each column is subdivided according to thumb's position: opposing the *palm*, angled (*pad*) opposition and on the hand's *side*. The position of the **TCM** joint — adducted or abducted — is then used to characterize the two rows.

A more condensed **grip** listing is presented in [Weir 2003], and depicted in Figure 2.7. We shall further classify these patterns in two categories: **precision** and **power grips**. The **precision grips** consist of *palmar*, *lateral* and *tip* prehensions — the latter encompassing both the *tripod* and the *pinch grasps*. The **power grips** comprise *cylindrical*, *spherical* and *hook* prehensions. This listing originally predates the work of [Feix et al. 2009] in over 50 years, but it is still one of the major guidelines in specifying and assessing prosthesis functionality. Due to its classic tested-and-proven nature and reduced complexity, it will also serve as basis for other evaluations and analyses in subsequent chapters of this work.

Due to the organization of the intrinsic and extrinsic sets of muscles, the human hand can easily alternate between fast, powerful motions and slow, precise ones. The average male hand can produce 95.6N of force in palmar prehension, 103N in lateral prehension and up to 400N in a cylindrical **grasp**. In general, **ADLs** require maximum of 66.7N to be carried out [Heckathorne 1992]. However, the majority of the lighter manipulations require less than 10.4N [Smaby et al. 2004].

Maximum human finger velocities were measured at approximately 2290°/s at the **MCP** joint. Average velocities during **ADLs** are considerably lower — in the range of 172°/s [Weir 2003].

²In this work, the terms **grasp** and **grip** (noun) are used interchangeably.

Opposition Type: Virtual Finger 2:	Power						Intermediate			Precision					Side 3
	Palm		Pad				Side			Pad					
	3-5	2-5	2	2-3	2-4	2-5	2	3	3-4	2	2-3	2-4	2-5		
Thumb Abd.															
Thumb Add.															

Figure 2.6: Hand **grasp** taxonomy, with 33 different patterns. Source: [Feix et al. 2009]

2.3 Types of upper-limb amputations

How an amputation occurs varies largely from one patient to another, leaving **residual limbs** with variable lengths and shapes that require different types of prosthetic **sockets** and fitting procedures. Familiarity with the main classifications of upper-limb amputations is of relevance for the development of this work. As depicted in Figure 2.8, those are:

- Wrist disarticulation:** Amputation in the wrist joint. The forearm is left mostly undamaged, and wrist rotation can still be accomplished.
- Transradial amputation:** Occurs in the forearm, through the *radius* bone — hence the name. The **stump** may have various lengths.
- Elbow disarticulation:** Amputation at the elbow joint.
- Transhumeral amputation:** Occurs at the upper arm, through the *humerus* bone.
- Shoulder disarticulation:** Amputation at the shoulder level, with the shoulder blade remaining. If both the shoulder blade and the collar bone are removed, the case is named a **forequarter amputation**.

From all the upper-limb amputees in the **United States (US)**, it is estimated that 70% of them have amputations distal to the elbow, i.e. **transradial** amputations and **wrist disarticulations** [McGimpsey et al. 2009].

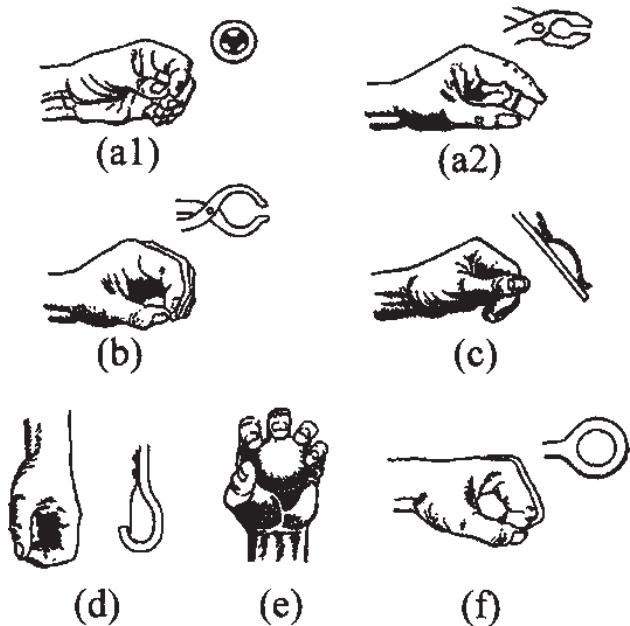


Figure 2.7: Reduced **grip** list. (a1) palmar prehension (tripod), (a2) palmar prehension (pinch), (b) tip prehension, (c) lateral prehension, (d) hook prehension, (e) spherical (power) prehension, (f) cylindrical prehension. Source: [Feix et al. 2009]

Using the concepts discussed in this chapter, one can evaluate the functionality of an upper-limb prosthesis in comparison to the human hand. Similarly, one can use this information to specify a prosthetic device that produces satisfactory results. With that in mind, the next chapter will present the state-of-the-art in both commercial and **open source** artificial upper-limbs, also devising technical requirements for the proposed prototype in the sequel.

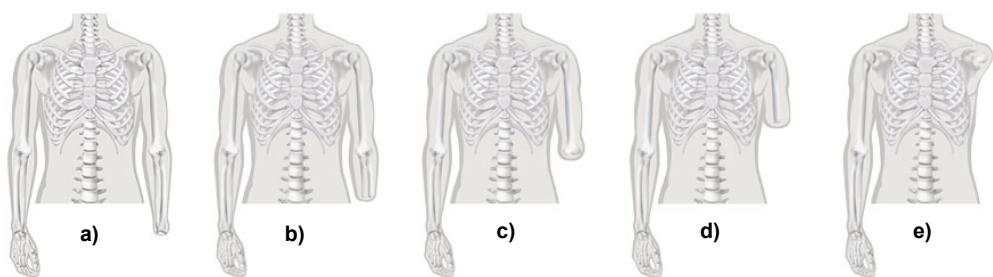


Figure 2.8: Main classifications for upper-limb amputations.

3

Prosthesis Requirements

“The human hand, with its elaborate control system in the brain, is doubtless the most widely versatile machine that has ever existed anywhere. Its notorious deficiency lies in its persistent inability to create a similar machine as versatile as itself.”

– Eugene F Murphy, Ph.D., *Engineering — Hope of the Handless*

Designing a device that mimics — or even surpasses — the functionality of a lost limb is the main goal in prosthetics research. The field has seen considerable improvements over the last two decades, departing from state-of-the-art upper limb prosthesis with one or two **DOFs** to modern ones with 6 or more [RSLSteeper 2015].

However, completely restoring lost functionality is still a rather distant objective. Compared to the human hand, even state-of-the-art devices lack high performance and durability [Belter et al. 2013]. Designing artificial hands has proven itself far more challenging than the design of robotic arms — prosthetics and robotics are actually much less related fields than one might expect. Robotic manipulators can generally be as large as necessary to perform a certain task, whereas prosthetic devices must be developed under strict constraints of weight, size and energy consumption, to name a few [Weir 2003].

Beyond the design itself, interfacing the prosthesis with its user is another great challenge, and a field of research on its own. Most current prosthetic hands are either body-powered through cables (mechanically driven) or implement some sort of **Electromyography (EMG)** based control strategy. **EMG** techniques use electrodes placed on the skin to measure the electrical activities of the skeletal muscles. Such signals can be processed and converted into control inputs for an external system [Criswell 2011]. Placing such electrodes on the residual muscles of an amputee's **stump** generally provides one or two **DOFs** for controlling an artificial limb. Effectively mapping this reduced number of signals into devices that require multi **DOF** control is difficult. Sequential control with interlocking mechanisms is often used to achieve such mapping, which may result in ponderous operation procedures.

Other types of interfaces are being explored for decades. Direct interfacing with the brain through the usage of **Electroencephalography (EEG)** aims at a more intuitive operation of an artificial limb, and its research has spawned positive results over the last decades [Lauer et al. 1999] [Srinivasan 2011]. Unfortunately, the difficulty associated with portably and reliably

obtaining usable **EEG** signals outside of a laboratory setting [Sanei et al. 2007] has hindered a widespread application of such techniques. *Targeted Muscle Reinnervation* (TMR) is a novel approach that pursues both natural prosthesis control as well as lifelike feedback for the patient. TMR attempts to establish a direct connection between the prosthetic device and the remaining nerves in the patient's **stump**. While its implementation has the drawback of requiring a still experimental surgical procedure, the results are very promising [Kuiken et al. 2009].

The expertise and research involved in the development of an artificial limb is vast, and so is the range of knowledge already available in the field. In this chapter, Section 3.1 exhibits the state-of-the-art in upper-limb prosthetics, addressing both commercial and **open source** developments. The design and functionalities of these devices is discussed in Section 3.2. Based upon this discussion, a table of requirements for a novel prototype is defined in Section 3.3. This table will then serve as guideline for the evolution of this work in subsequent chapters.

3.1 State-of-the-art

In order to gain some insight on the current market and research in the field of prosthetics, this section presents the state-of-the-art in upper-limb prosthetic devices. Initially, three of the market-leading artificial hands are exhibited, together with their most relevant measurements and performance informations. In the sequel, the origin and characteristics of the **open source** movement in prosthetics are discussed. Two relevant **open source** projects are then presented.

3.1.1 Commercial products

3.1.1.1 Bebionic3

Advertised as "the world's most lifelike bionic hand", the *bebionic* is the flagship product of the UK-based company, *RSLSteeper*. With over 90 years of experience in prosthetics, *RSLSteeper* also manufactures custom and off-the-shelf silicone cosmetic covers for artificial limbs.

Currently on its third generation, the *bebionic* (shown in Figure 3.1) has 6 active **DOFs** and comes with 14 preset **grip** patterns. Among these are the **precision** and **power** grasps defined in the previous chapter, as well as a finger-adduction-based **grip**, and a few others. The device comes in large and medium sizes (mainly for males), and also features a small size, tailored for the female public.

According to the specification sheet [RSLSteeper 2015], the *bebionic* weights an average of 620g across its available configurations. Custom-designed linear actuators mounted in the palm apply force directly on the hand's **MCP** joints, granting robust actuation. It can exert up to 140N in a cylindrical grip, 36.6N in palmar prehension and 26.5N in lateral prehension (though these **grasps** are correspondingly called *power*, *tripod* and *key grip* by the manufacturer). Opening/closing times for those **grips** are respectively rated as 1.0s, 0.5s and 1.0s. Additionally, the device is capable of supporting 45kg while in a hook prehension, and can safely withstand loads of 90kg taken across the knuckles.



Figure 3.1: *Bebionic3* from manufacturer *RSLSteeper*¹.

The *bebionic* interfaces with the user through one or two custom **EMG** electrodes. Bundled software allows the hand to be calibrated to suit the patient's needs and characteristics, also enabling him to practice the device's operation in a simulated environment. Sporting battery packs of up to 2200mAh, *RSLSteeper* claims that a fully charged battery will last throughout a regular day of usage. The previous generation of the *bebionic* could execute about 3200 cycles (opening/closing) of **palmar prehensions** before running out of charge. This information, however, is not directly provided for the current version.

Depending on the chosen configuration (size, interface, software, etc.) the *bebionic* cost may range from U\$25,000 to U\$35,000. The total price tag may be even higher if other fitting and adjustments are required (which is normally the case).

3.1.1.2 i-Limb

Scottish manufacturer *Touch Bionics* launched the first model of the *i-limb* in 2007, the first commercial prosthetic hand to incorporate individually driven, articulating fingers. Almost a decade later, the *i-limb* is currently available in three different models: the *ultra*, the *revolution* and the *quantum*. Though visually similar, each model offers increasingly more features — specially in terms of software. The *ultra* (Figure 3.2), the most basic model, comes with 14 preset grips. The *revolution* and the *quantum* provide 24 ones, and the latter model still enables the user to store another 12 custom patterns.

The whole product line also features bluetooth capabilities, allowing the device to communicate to a smartphone. The user can access a specific application on the phone to quickly alternate between **grip** patterns. *Touch Bionics* also provides software for a various of activities related to the prosthesis, such as calibration, training, diagnostics and others.

¹Source: http://bebionic.com/downloads/product_literature



Figure 3.2: *Touch Bionics' iLimb Ultra*. Source: [Touch Bionics 2014]

The mechanism of each finger in the *i-limb* line employs a motor contained in the **proximal phalanx**, which drives a worm-gear around a stationary gear in the knuckles — the whole mechanism fits approximately in the volume of one finger. The high gear ratio in this coupling allows even the *i-limb ultra* to attain 136N in a **cylindrical prehension** and 35N in a **lateral prehension**. Opening/closing times for the cylindrical prehension are rated at 1.2s. The devices weight an average of 540g [Touch Bionics 2014].

Another product from *Touch Bionics*, the *i-limb digits*, takes advantage of the reduced size of the finger-driving-mechanism to enable fitting of custom-built devices for partial amputee, i.e., patients with some remaining portion of the hand. Such device is depicted in Figure 3.3², and is currently the only of its kind available in the market.

Detailed information on battery life is not available, but the specification sheets of the *i-limb* series state that a fully charged device should operate during a complete day. Depending on how far up the arm the prosthesis has to extend, which fitting procedures are necessary and which model is picked, the total cost of an *i-limb* device can range anywhere from U\$30,000 to U\$120,000 — it is, however, hard to find any officially disclosed information on the units' prices.

3.1.1.3 Michelangelo Hand

Founded by the german prosthetist Otto Bock in 1919, *Ottobock* is one of the largest companies in the orthopedic industry, and the developer of the first **myoelectric** prosthesis in the 1960s. *Ottobock* currently sports a large portfolio of upper-limb prosthetic devices, ranging from solely cosmetic devices and body-powered grippers to its most advanced unit, the *Michelangelo* hand, shown in Figure 3.4.

The *Michelangelo* is a 2 **DOF** device. One large motor in the center of the palm simultaneously controls flexion and extension of the **digits**. Another one, also in the palm, sets the thumb's position — enabling both palmar and lateral prehensions. Due to the non-articulated nature of the fingers however, **power grips** are not available. According to its specifications [Ottobock 2013], the device can produce up to 70N in palmar prehensions and

²Image source: <http://www.o-pspecialists.com/technology/ilimb.html>



Figure 3.3: *Touch Bionics' iLimb Digits*, an alternative for patients with partial hand amputation.

60N in lateral ones, and can be operated by one or two **EMG** electrodes. The 500g hand uses a separate, 140g battery pack, and is allegedly capable of running up to 20 hours on a single charge.

The adaptive nature of its grasps is probably one of *Michelangelo's* main traits. Though the **digits** present only the **MCP** joint, they feature a flexible coupling to the hand's main motor. This allows grips to better adapt to the shape of the object being handled. The device can also be equipped with a passively flexible wrist (with a manual lock, when rigidity is needed). In 2013, a *Michelangelo* unit cost around U\$74,000.

3.1.2 Open source developments

Each person's body shape and habits make an artificial limb as personal as a biological one. When reading stories of amputees and their prosthetics, one can often encounter reports on **Do-it-yourself (DIY)** prosthesis modification. Besides recurrent visits to the local prosthetist³, homemade attempts on making the artificial limb more comfortable, functional or simply aesthetically pleasing are often undertaken by many patients. Trimming and filing, sawing, coloring and trying new attachments are a few of the customizations experimented on the devices [Motherboard 2015].

Amputees try such modifications on their own mainly due to the severe costs associated with the majority of products and services in the realm of prosthetics: upgrading one's artificial limb, purchasing a new one or even getting a custom-built solution can set one back dozens of thousands of dollars. Some amputees, facing those costs, choose not to wear a device at all.

³As defined by *The American Board for Certification in Orthotics*, a prosthetist is a "person who measures, designs, fabricates, fits, or services a prosthesis as prescribed by a licensed physician, and who assists in the formulation of the prosthesis prescription [...]".

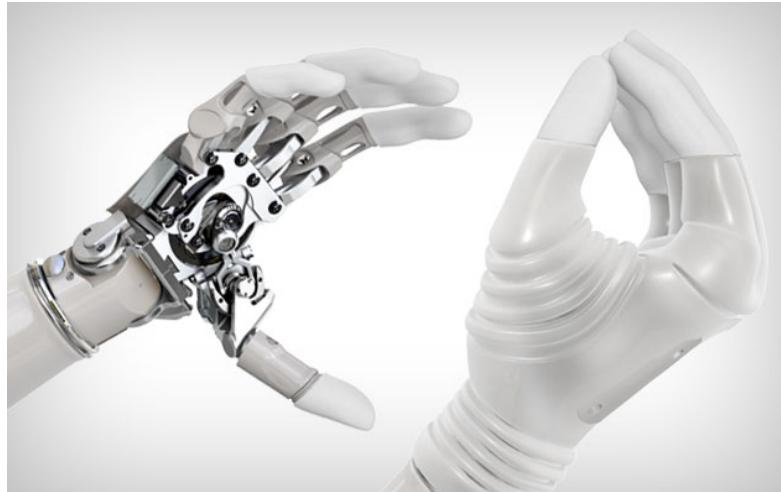


Figure 3.4: *Michelangelo* hand from *Ottobock*. Hand without the rubber cover on the left; covered on the right.

The last five years have seen a massive expansion of the household 3D-printing market. Home grade 3D-printers mostly make use of **FDM** techniques: a plastic filament is molten as it gets pushed through and extruder. According to computer's instructions, the printer's mechanism moves this extruder around, stacking layers of plastic and creating the desired volumetric object. This technique is safe, cost-effective, uses commonly available plastic (generally **Acrylonitrile Butadiene Styrene (ABS)**) and can easily generate objects with complex geometries. With its diffusion, 3D-printing was quickly deemed as a powerful tool for the development of better prosthetics.

3D-printing applications in the field of prosthetics began to be explored not only by large manufacturers, but by the same **DIY** public that was striving for customization possibilities. Soon enough, non-amputees started to join the effort. In a scenario that resembles the early days of the Internet, each user usually shares their⁴ latest developments, gets feedback and allows modifications to their designs. This practice of supplying the blueprints of one's work is a clear example of **open source** collaboration, and has spawned the creation of dozens of interesting and useful prosthesis models in the last couple years.

In the following, two of the most relevant **open source** bionic hands shall be briefly discussed. There are, nonetheless, numerous other great projects available. Worthy of mention are Richard van As' *Robohand*⁵ and the *Enabling the Future* initiative⁶. Closely related projects, both work on supplying 3D-printed, cheap, mechanical-only hand prosthesis for children and adults with hand malformations or missing **digits**. Unfortunately, a more comprehensive compendium of such works lies beyond the scope of this document.

⁴This document often employs the gender-neutral, third person singular pronoun *they*, as adopted in the *Oxford Dictionary*.

⁵<http://www.robohand.net/>

⁶<http://enablingthefuture.org/>



Figure 3.5: Boy wearing a *Robohand*.

3.1.2.1 InMoov

Gael Langevin is a french sculptor and modelmaker. In the beginning of 2012, he used his desktop 3D-printer to print a life-sized right hand and forearm. Langevin added some **servos** motors (devices generally used in radio-controlled planes) in the forearm to drive the hand's fingers through cables (setup shown in Figure 3.6). Inspired by the result, he decided to print a left hand version. Satisfied with the pair, he designed a set of biceps and a minimal torso. Three years later, born out of this incremental endeavor, the *InMoov* project is currently a fully 3D-printable life-sized humanoid robot, as seen in Figure 3.7. The robot now features complete arms, a torso, an on-board battery and electronics, a head equipped with two cameras, microphones and speakers, and can execute several gestures, voice recognition, face tracking, and much more.

Since the beginning of the initiative, Langevin openly shared his design files. He designed the robot in a way that every part can be printed even in the smallest desktop printer, which led to a popularization of his project in the **maker** community. *InMoov*'s hand design was the first of its kind, and quickly drew the attention of french **maker** Nicolas Huchet, a right hand **transradial** amputee. Huchet contacted Langevin, inquiring about the possibility of using *InMoov*'s hand as a prosthetic device. This quest became a parallel project on its own, and in mid-2014 the first version of the *InMoov Prosthetic Hand* was finished (Figure 3.8).

Completely 3D-printable, the *InMoov Prosthetic Hand* uses off-the-shelf motors and electronics. It also embeds a tiny, commercially-available, **open hardware** EMG interface board, which permits the use of one **myoelectric** sensor as control input from the user. All those components are housed inside the hand, making it a possible fit for upper-limb amputees with long **residual limbs**. However, due to the size of the available components, Langevin decided to design a hand with only four **digits**: three single-**DOF** fingers and a 2-**DOF** thumb. This kept the hand from being too wide, and enabled future adaptations of the design for both women and children [Langevin 2015]. The use of readily-available products and desktop 3D-printing keeps the cost of an *InMoov Prosthetic Hand* under the U\$350 mark.



Figure 3.6: *InMoov*'s forearm and hand. The motors are housed in the forearm, and cables running up to the hand actuate on the fingers and thumb.



Figure 3.7: Fully assembled *InMoov* robot.

3.1.2.2 Open Bionics

In 2012, british **maker** Joel Gibbard quit his day-job to pursue his vision of designing 3D-printable, affordable artificial hand suited for everyday activities. Almost a year later, he finished a first version of the *Dextrus Hand*, shown in Figure 3.9. The *Dextrus* uses five **Direct Current (DC)** motors in the hand's palm to move each **digit** via a cable mechanism, and one **servo** to reposition the thumb.

While developing the hand, Gibbard resorted to a crowdfunding⁷ platform to make the project viable. This attracted the media's attention, and in turn, other **makers'**. The initiative grew and, in 2014, Gibbard founded UK-based *Touch Bionics*. He assembled a small team of designers and engineers, and began upgrading his original design.

In early 2015, *Touch Bionics* unveiled their *Dextrus Hand V2* — exhibited in Figure 3.10. The device is 3D-printed almost entirely in one single piece, using commercial-grade flexible plastic filaments. This renders the **digits** flexible enough to be actuated by a cable-driven mechanism similarly to the first *Dextrus*, while keeping the overall assemblage at a minimum. The *Dextrus V2* also relies on two **open hardware** EMG sensors as control inputs from its user. This design won *Open Bionics* a U\$200,000 prize in Intel®'s 2015 *Make it Wearable* contest [Open Bionics 2015].

Open Bionics states that the cost to build a *Dextrus V2* is inferior to U\$1,000.

⁷As defined in the *Oxford Dictionary*, crowdfunding is the practice of funding a project or venture by raising monetary contributions from a large number of people, typically via the internet.



Figure 3.8: Assembled *InMoov Prosthetic Hand*.



Figure 3.9: *Dextrus hand*, first version.



Figure 3.10: *Dextrus hand V2*.

3.2 Discussion on current designs

The devices presented in the last section are the result of research, effort and development undertaken by various individuals and teams. Much knowledge can be obtained by analyzing design choices and features in each of these prostheses. With that in mind, this section begins with a discussion on the weights and sizes of the mentioned devices. Afterwards, the various mechanisms they employ to implement the **digits** are addressed. Lastly, other functional aspects of these devices are debated.

3.2.1 Weights and sizes

The weight of the discussed commercial hand units, together with batteries and sensors, tends to exceed the biological hand's 400g average. Even when using devices with the correct average mass, amputees report the artificial limbs as too heavy [Pylatiuk et al. 2007]. The

socket around an amputee's **stump** distributes the prosthesis' loads on the soft tissue, instead of the skeleton. This in turn increases the perceived weight, and leads to fatigue and discomfort. A weight limit below the 400g mark is thus suggested [Belter et al. 2013].

Such medical-grade devices generally use aluminum or steel for the main structural parts. While this ensures great strength, the average densities of such materials (2.7g.cm^{-3} and 7.8g.cm^{-3} , respectively) together with the mass of the embedded actuators, are the primary causes of the aforementioned overweight. The weights of the *Dextrus Hand V2* and the *In-Moov Prosthetic Hand* were not officially unveiled, but the density of **ABS** plastic — roughly at 1.1g.cm^{-3} — opens possibilities for lighter devices. Though **ABS** is less resistant, comfort levels can be greatly improved.

After the end of the U.S. Civil War in 1865, hundreds of wounded soldiers with limb losses were fitted with primitive prosthetics. Those were mostly simple hooks or peglegs, but the overwhelming dominance of male patients shaped the industry until current days. Up to early 2015, when *RSLSteeper* and *Touch Bionics* announced the small versions of their prosthesis, there were no microcontrolled hands sized for women. Representing 30% of all the amputees in the U.S., the majority of them still has to resort to custom designs fitted by prosthetists. Female amputees are considered a slim profit margin by many manufacturers, so finding products tailored for this public is an arduous task [Motherboard 2015].

The lack of flexibility in traditional manufacturing processes (e.g. milling and injection molding) plays a role in biasing the industry. A smaller prosthesis is a product on its own, and may require various production steps that differ from those of its larger counterpart. Even for a large company, setting up a production line for such a new device can be very expensive and risky. Conversely, 3D-printing decentralizes the manufacturing effort — and thus its risk. Different from large-scale fabrication processes generally employed in the industry, a 3D-printer generates one part at a time, imposing little limits to its geometry and customization. This is a vast improvement towards personalized prosthetics, and is a trend in its early days — as the *Dextrus Hand V2* already points out.

3.2.2 Finger mechanisms and actuation

As analyzed in the previous chapter, each human finger has at least three active **DOFs**. Independent operation of each **DOF** would require several actuators, which is extremely difficult to achieve within the dimensional constraints imposed by the application. Hence, all of the devices presented implement some sort of coupling mechanism between joints, reducing the amount of actuators needed.

The *i-limb* and the *Bebionic* feature an **MCP** joint and a single distal articulation similar to a human **PIP** joint. On the fingers, a distinctive extrusion on the distal segment represents a static **DIP** joint, fixed at 20° . Both hands couple the motion of this **PIP** articulation to the **MCP**'s rotation using some kind of 4-bar linkage⁸. The *i-limb* uses a tendon system, where a cable connects the base of the finger to the distal phalanx (Figure 3.11, **a**) In the *Bebionic*, a plastic rod connects the distal phalanx to the finger's base (Figure 3.11, **b**). Slightly different, the *Michelangelo hand* employs a single-segment finger, actuated only at the **MCP** joint (Fig-

⁸This mechanism shall be further detailed in the upcoming section.

ure 3.11, c)). The *InMoov Prosthetic Hand*'s finger implements all three joints, using two 4-bar linkages — one coupling the **MCP** to the **PIP** joint, and other coupling the **PIP** to the **DIP**, as seen in Figure 3.13. It is noteworthy that none of these prostheses implements an adduction/abduction **DOF**. Seeking further mechanical simplicity, all of them use fixed abduction angles for each finger.

In both the *Bebionic* and the *i-limb*, the angle change ratio between the **PIP** and the **MCP** joint is approximately 1.1. This change ratio controls how the fingers wrap around objects of different size. In human hands, this proportion differs depending on the **grip** that is being executed. Artificial hands that couple the motion of its joints are sometimes referred to as *rigid* or *synchronous* grippers [Belter et al. 2013]. Using a different approach, both versions of the *Dextrus Hand* implement an underactuated finger mechanism. In it, the three free-moving joints (3 **DOFs**) of the finger are actuated by a single cable running inside them (single control **DOF**), as shown in 3.12. This scheme, often named *adaptive* or *asynchronous* gripping, allows fingers to more easily wrap around objects during a grasp, better distributing the contact force (depicted in Figure 3.14).

While this idea seems appealing, the work in [Bergman et al. 1992] demonstrated that conventional *synchronous* prosthetics showed “significantly better results” than their *asynchronous* counterparts when performing a standardized hand function test. This is partially due to the fact that, in an underactuated setting, the fingers cannot be stably held in a stationary position if not completely supported against an object.

Nevertheless, *adaptive* grippers are intrinsically **compliant**, meaning that they are able to move when an external force (e.g. collision, user's manipulation) is applied. This helps preventing damage when the device hits or bumps into something, forcing the fingers to close. Since underactuation is not present in *rigid* grippers, there is no **compliance** inherent to the mechanism. Thus, both the *i-limb* and the *Bebionic* use some mechanical element that allows a certain level of it. In the *i-limb*, the tendon-driven **digits** can be pushed inwards regardless of the **MCP** joint's position — a spring-loaded **PIP** joint returns the phalanges into place. In the *Bebionic*, the proximal phalanx of each **digit** is connected to its actuator via a pin inserted in a elongated slot. When an external force is applied, the pin runs up in the slot, permitting both **MCP** and **PIP** flexion. The *InMoov Prosthetic Hand*, also a *rigid* gripper, does not implement any mechanism of such kind.

3.2.3 General usability

Executing **ADLs** poses a broad range of requirements for a hand. While many **ADLs** require fast speeds and low grip forces (typing, gesturing), a few others demand high grasping forces (opening a door, lifting oneself). In the human hand, those heterogeneous goals are achieved by different sets of muscles, some running along the entire forearm⁹. With a much more constrained space, the choice of actuators in an artificial limb seeks an appropriate trade-off between speed and strength that keeps the device minimally functional.

According to the data available, the *Bebionic*, *i-limb* and *Michelangelo* devices can produce (and even exceed) the maximum **grip** forces needed to perform **ADLs** — approximately

⁹Consult section 2.1.2 if needed.

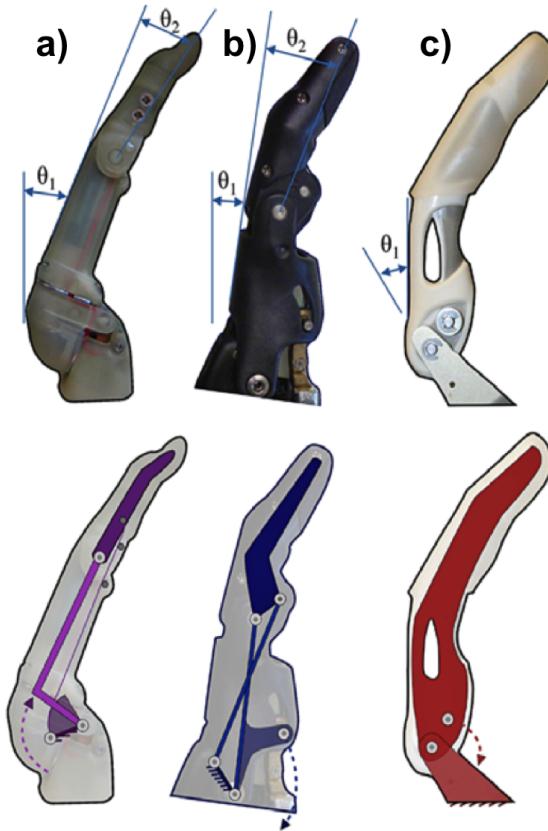


Figure 3.11: Finger mechanisms for: **a)** the *i-limb*, **b)** the *Bebionic*, **c)** the *Michelangelo hand*. θ_1 and θ_2 are the angles of the **MCP** and **PIP** joints, respectively. Source: [Belter et al. 2013]

67N. Fulfilling that requirement comes at the price of reduced **digit** speed: in all three devices, maximum speed at the **MCP** joint is inferior to 100°/s. This represents almost half of the average 172°/s executed by the human hand during similar activities — and contrasts greatly with a biological finger's 2290°/s maximal velocity. Although closing times of up to 1.5s are suggested in the literature [Weir 2003], most amputees describe their **myoelectric** prostheses as "too slow" [Pylatiuk et al. 2007].

Another challenge in conceiving a prosthetic hand is designing the thumb. Effectively implementing all the thumb's 4 **DOFs** would, once again, be extremely difficult and costly — too many moving parts could also reduce **digit**'s robustness. While simplifying this mechanism, it is relevant to remember that the thumb accounts for 40% of the hands functionality, largely defining which **grip** patterns are achievable [Heckathorne 1992]. The majority of the discussed devices employs a 2 **DOF** thumb mechanism. In both the *Bebionic* and the *i-limb*, the spherical motion of the human **TCM** joint is simplified to rotation around a single axis, known as the **circumduction** axis (as seen in Figure 3.15). An additional active **TIP** joint enables thumb flexion/extension. The *Bebionic* and most versions of the *i-limb* require the user to manually position the thumb — mechanically snapping it into position. On the *InMoov*

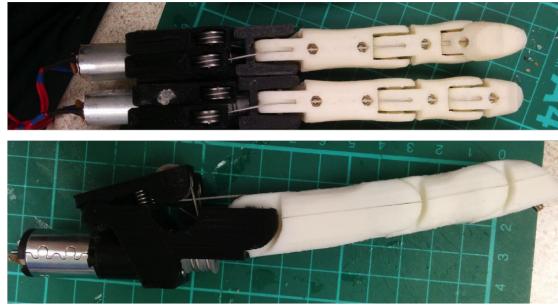


Figure 3.12: *Dextrus*' cable-driven finger mechanism, frontal (top) and lateral (bottom) views.

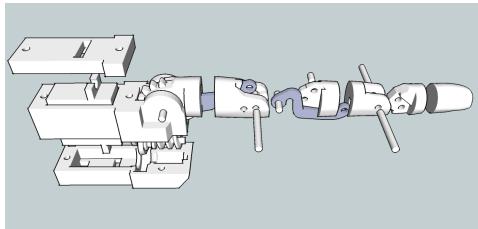


Figure 3.13: Exploded view of the *InMoov Prosthetic Hand*'s finger. Source: [Langevin 2015]

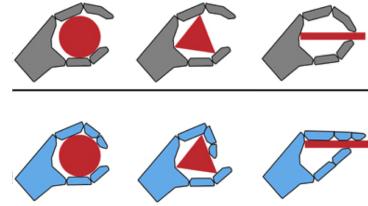


Figure 3.14: Behavior of a *rigid* (top) and an *adaptive* (bottom) **grasp**. Source: [Open Bionics 2015]

Prosthetic Hand however, the thumb **circumduction** is controlled by a motor in the palm. Contrarily to those designs, the *Dextrus Hand V2* uses a **1 DOF** thumb, mounted diagonally at approximately 45° to the hand (Figure 3.16). This configuration optimizes the functionality levels for single **DOF** thumbs [Weir 2003].

Lastly, there is a complex trade-off between cost and robustness. The average **myoelectric** prosthetic hand user will wear their device for at least 8 hours a day, typically executing 120 grasping motions. Such devices should thus ideally endure at least 500,000 grasping cycles before routine servicing [Belter et al. 2013]. The degree of robustness in a prosthesis is heavily related to the materials used in its fabrication. Specialized metal alloys are very suitable choices, but also very costly ones. Moreover, they have to undergo long, expensive machining processes during manufacturing. When those finished devices hit the market, their extremely high prices reflect this complexity and underlying cost.

For the vast majority of amputees, these prices are the biggest obstacle when obtaining such devices, also meaning that repairs or replacements are often not available or prohibitively expensive. Considering that prosthesis wear out usually in the course of three to five years, these price tags quickly turn into a major problem [HSS 2015]. The discussed **open source** developments explore a different perspective on the same issue. While 3D-printing still produces artificial limbs less robust than commercial ones, the ease and the tremendous cost reduction to obtain and repair 3D-printed devices opens up a whole new paradigm. In it, prosthesis are less of a long-lasting solution, to be more of a personal and dynamically changing piece of equipment.



Figure 3.15: Bottom view of the *Bebionic*'s thumb. The dashed line originates on the thumb's circumduction axis and highlights its plane of motion. Source: [Belter et al. 2013]

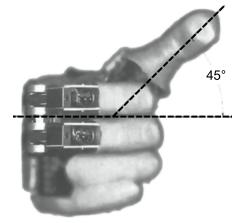


Figure 3.16: Preferred operating plane for a single **DOF** thumb. Source: [Weir 2003]

3.3 Technical requirements

As previously discussed, the main goal of this work is to design and implement a low-cost **open source** prosthetic hand. It is now clear that there is a lot of knowledge and expertise currently available in the field of prosthetics, and that its **open source** initiative has a great potential in shifting the structure of this industry in the future. In order to shape a relevant collaboration to this panorama, the current section aims at devising and presenting a fairly simplified requirement table for the proposed prototype.

From an engineering standpoint, technical requirements specify the desired properties and behavior of a system, and can be mainly divided into two groups: *functional* and *non-functional* requirements. *Functional* requirements define a system's services or functions — what the said system *does*. *Non-functional* requirements outline the system's characteristics and constraints to its development process — laying out what it should *be*. Requirements capture one's understanding of a problem and the proposed solution at a given point on time, and as such, may evolve and change [Sommerville 1995]. Nonetheless, they serve as an important guideline during the development of any technical system. When completed, a system can undergo a formal evaluation, effectively verifying if its planned characteristics and functionalities were fulfilled.

Table 3.1 contains the high-level requirements defined for this project. Many requirements are posed in a rather loose form (i.e. with no definite target values) since much of the design is experimental. Rather, the majority of the requirements are formulated relative to the previously discussed prosthetic devices. The table's first column subdivides the entries in two groups: *function and usability* and *fabrication and accessibility* — respectively *functional* and *non-functional* requirements. The second column attributes them a sequential number and the third one, its priority during implementation. A 0 priority is the highest, 1 is intermediate and 2 is the lowest. This usage of priority markers aims at defining a certain degree of hierarchy within tasks in the time of development.

In the *function and usability* row in Table 3.1, the requirements 1.1 through 1.5 capture the minimum functionality for the prototype, based on similar characteristics in all the previously discussed devices. Since the literature hardly provides some standard on an average human hand size, the constraint in 1.1 mimics the size of the largest *Bebionic* model. In 1.4, the

Type	#	P	Requirement
1. Function and usability	1.1	0	Have a human shaped form and size (maximum palm width of 92mm), with 4 or 5 digits .
	1.2	0	Perform all the 4 listed precision grips and the 3 power grips .
	1.3	0	Execute grasps opening/closing according to user's input.
	1.4	1	Produce at least 16N in <i>palmar</i> prehensions. ADLs .
	1.5	1	Permit the usage of a portable power supply.
	1.6	0	Be suitable for patients with wrist disarticulation and transradial amputation.
	1.7	0	Endure indoor usage for at least 4 hours.
	1.8	1	Switch between grip patters without manual digit repositioning.
	1.9	1	Have a digit speed similar or superior to the ones in commercial devices.
	1.10	1	Have a weight similar or inferior to the ones of commercial devices.
	1.11	2	Allow some degree of control of the digits ' position and force applied during a grip .
2. Fabrication and accessibility	2.1	0	Keep all files needed for its replication publicly available for download and unrestricted modification.
	2.2	0	Have a total cost similar or inferior to the one of other open source devices (inferior to U\$400).
	2.3	1	Use only off-the-shelf parts that can be easily obtained from different suppliers.
	2.4	1	Require only accessible, consumer-grade tools during its manufacturing.
	2.5	2	Reduce the amount of needed tools and build complexity as far as possible.
	2.6	1	Use only cost-free software tools, preferably open source ones.

Table 3.1: Requirements for the prototype.

aforementioned minimum force of 67N is not required, but a minimum force of 16N is defined based on the work presented in [Smaby et al. 2004]. Requirement 1.6 specifies patients with longer residual limbs, which poses the need for a compact device. As stated in requirement 1.7, only indoor usage is foreseen. Outdoor environments are harsher, and preparing the first generation of the prototype to endure in them would introduce undesired complexity into an untested design. Requirements 1.8 to 1.11 define slight improvements over a few of the discussed devices. Item 1.8 addresses the manual repositioning of the thumb in both the *i-Limb* and the *Bebionic*. Lastly, requirement 1.11 attempts to incorporate a feature to control the **digits**' position and force during a **grip**, which is missing in most of the previously presented prosthetics.

The *fabrication and accessibility* row presents requirements oriented towards the **open source** nature of the project. The maximum price tag established in 2.2 was defined based on

the average cost of the *InMoov Prosthetic Hand*. Requirement 2.6 aims at facilitating third-party development: by ensuring that the software tools used can be obtained with no cost, other users or **makers** can modify the design to suit their needs more easily.

With these definitions and with the information discussed up to this point, the following chapters of this work shall present in detail the development process of the proposed prototype.

4

Prototype Design

The prototype was designed based on the information available and the requirements established in the previous chapters. The following sections in this chapter shall present this design process in detail. Section 4.1 outlines the prototype's shape, power source and manufacturing specifications. In Section 4.2 the mechanical design of the prototype is discussed step by step. Subsequently, in Section 4.3, the design of the embedded electronics is addressed. Lastly, Section 4.4 presents the software elements of the prototype, employed tools and development environment.

While the coming sections present the developed work in a very linear fashion, in actuality many separate steps occurred in parallel, since many decisions affecting one aspect of the prototype directly influence others.

4.1 General specifications

The loose nature of the requirements posed in Table 3.1 captures only the essential characteristics in solving a very open-ended problem. Countless approaches to designing artificial hands were already explored — from porcelain covers, pneumatic and hydraulic designs, up to modern models —, so much that only a small fraction of those could be discussed in this document. It is usually not feasible to formally assess the viability of each of those approaches when starting a new project. Often, many of the initial design decisions — specially those regarding tools and techniques — are defined by the designer's (or team's) familiarity, expertise and previous experience.

As a one man endeavor, this project faced a similar process. A few initial design specifications for the prototype were taken out of the designer's own preferences and experience, and — mostly due to time constraints — did not undergo a more rigorous evaluation. Much of this work is built upon these specifications, which will be briefly presented in the following subsections. Initially, the rationale for the choice of a human-shaped robotic hand is addressed. In the following, the manufacturing technique chosen for the prototype (namely, 3D-printing) is discussed. At the end of this section, the choice of power source for the device is justified.

4.1.1 A robotic hand

Aside from hand-shaped upper-limb prostheses, there are various other approaches, such as hooks and claws. Not only are they usually more robust (due to reduced mechanical complexity) but can also emulate many, if not all, prehensions needed to accomplish **ADLs** [Weir 2003]. Those devices are generally driven by a body-harness (cable-driven) and require no batteries.

In spite of such remarks, this work still pursues the implementation of an electrically driven, human-shaped (bionic) hand. Beyond having a better cosmetic aspect, robotic hands feature many active **DOFs** that can provide capabilities beyond any single **DOF** gripper. The interest in uncovering novel functionalities in such multi-**DOF** devices was the deciding point in opting for a human-like shape. On top of that, the performances of both the *i-limb* and *bionic*, served as an inspiration for a design with individually actuated fingers.

4.1.2 Manufacturing

Much of the interest on this project came after realizing that 3D-printing could be a great tool for customizing and making prosthesis. After coming across both the *Dextrus Hands* and the *InMoov Prosthetic Hand*, it became clear that not only was the idea possible, but also there was rapidly growing **maker** community working around the concept.

After some research on the state-of-art in low-cost desktop **CNC** mills, 3D-printing was chosen as the default fabrication tool for this work. With good consumer grade 3D-printers costing as low as *U\$400*, it is currently the most affordable way of rapidly creating objects with complex geometries. Lacking good resellers, a custom 3D-printer model was entirely designed and built for this project, as detailed in Appendix A.

4.1.3 Power source

Designing a prosthesis poses various size and functionality constraints, as previously discussed. Clearly, the device needs to be used untethered from a power source, and batteries are required. As presented in [Weir 2003], standard electrical power sources in prosthetics range from $4.8V$ to $12V$. Congruent to that observation, a **2S Lithium-Polymer (LiPo)** battery — with a nominal voltage of $7.4V$ — was chosen as the power source for the prototype. Such batteries are used in *radio controlled* models (e.g., planes, cars), making them easy to find and very affordable. Designed for high current draws, most **2S** batteries can easily handle discharge currents of more than $20A$. This current and voltage characteristics make them suitable for the majority of the actuators in the size range required in this project (i.e., small enough to fit inside a prosthetic hand's volume).

4.2 Mechanical design

This section presents in detail the main steps involved in the design of the mechanical components of the prototype, according to the requirements defined in the last chapter and the general specifications discussed above. Initially, the choice of a proper actuator is addressed.

Type	Name (Reseller)	Speed	Force/ Torque	V (A)	Size (mm)	Weight (g)	Price (U\$)
DC Motor	Micro Metal Gearmotor 298:1 (<i>Pololu</i>)	100 rpm	0.49 Nm	6 (1.6)	10 × 12 × 26	9.5g	15.95
	Metal Gearmotor 73:1 (<i>Pololu</i>)	180 rpm	0.42 Nm	6 (3.3)	20 × 20 × 42	43.1g	19.95
	Metal Gearmotor 154:1 (<i>Pololu</i>)	90 rpm	0.94 Nm	6 (3.3)	20 × 20 × 42	43.1g	19.95
RC Servo	Standard Servo (All)	0.14 s/60°	1.08 Nm	6 (1.2)	20 × 40 × 41	43g	12.95
	Micro Servo (All)	0.08 s/60°	0.15 Nm	6 (1.0)	11.5 × 20 × 21.3	6.5g	9.95
Linear actuator	Firgelli L12 50mm (<i>Robot Shop</i>)	5 mm/s	45 N	6 (0.5)	15 × 18 × 49	43g	70.00

Table 4.1: Commercial actuators initially considered for the design. Motor shaft lengths are ignored in the *Size* column. The *V(A)* column displays the actuator's rated voltage and, in parenthesis, its stall current.

In the following, a detailed synthesis of the mechanism employed in the fingers is presented. Coupling this mechanism to the chosen actuator proves to be problematic, and these issues are discussed. As a result, a design for a new actuator is proposed and evaluated. The mechanism for the thumb and a palm are then devised. Lastly, the completed model of the system is briefly presented.

4.2.1 Actuator selection

One of the first bottlenecks addressed during the project was a proper choice of actuators. To be suitable, the chosen actuator would have to be compact, affordable and deliver at least 16 N of force. However, since the fingers will employ some sort of leverage mechanism, a much higher force than that is actually desired. Furthermore, during the early stages of mechanical and electrical outlining, a nominal operating voltage of 7.4 V was chosen. To avoid unneeded complexity in the electronics, having actuators capable of running off this voltage is crucial.

Starting with the affordability constraint, a search was performed among some of the leading low-cost consumer electronic resellers: *Pololu*, *Adafruit*, *Sparkfun* and *Robot Shop*¹. Table 4.1 presents some of the available actuators in an acceptable size, voltage and force (or torque) range. While the rated voltage of them all falls below the aforesaid nominal 7.4 V, inductive actuators (with coils) can generally be safely actuated with reasonably higher tension levels. In doing so, there is an increase in force and speed, but the actuator's lifetime may be slightly reduced.

¹www.pololu.com, www.adafruit.com, www.sparkfun.com, www.robotshop.com

The choice of the actuator not only shapes the development of mechanical elements around it, but also places requirements on the electronics. As seen from Table 4.1, though all actuators operate at the same nominal voltage, the stall current (i.e., the current drawn when the actuator is mechanically blocked while driving) varies largely.

The *Firgelli* linear actuator was a very suitable choice, at first sight. However, with at least one unit per finger, the total cost only for the actuators would reach U\$350, almost at the established price limit. Another issue with these actuators is their availability: those models are supplied by only one manufacturer, so finding resellers is not trivial. On top of that, a product with a single manufacturer is more prone to being discontinued, breaking designs based on it.

A much more available and cost-effective product are Standard **RC servos**². Commonly used in radio-controlled cars, planes and boats, these actuators embed all their driving electronics, and are standardized across various manufacturers. Though they deliver considerable torque, their form factor was deemed unsuitable for the application. *Micro servos*, also very commonplace products, are too weak to operate the hand's **digits**.

At the end, *Pololu's Metal Gearmotor (model 154:1)* was chosen (Figure 4.1). With a 154 : 1 gear reduction, the motor delivered high torque at low voltages and small packaging. Its high current requirements, however, (in comparison to other alternatives) placed a challenge to the electronics design. Having defined an actuator to drive each finger, the next step involved designing the fingers themselves.



Figure 4.1: *Pololu's Metal Gearmotor*³.

4.2.2 Finger mechanism design

Based on the conclusions presented in Section 3.2.2, a *synchronous (non-adaptive)* finger design was chosen. The mechanism used in the *bebionic*'s fingers (displayed in Figure 3.2.2) served as a guideline. As previously pointed out, the *bebionic* (and almost all of the discussed prosthesis) uses a 4-bar linkage to couple the motions of the **MCP** and **PIP** joints. Similarly, a finger with a single fixed distal segment (fixed **DIP** joint) was chosen.

A linkage (or *kinematic chain*) is an assembly of *joints* and *links* that provides a desired output motion in response to a given input motion. A *link* is a rigid body with at least two *nodes*; these, in turn, are the points through which one *link* connects to another via a *joint*. *Joints* allow motion to occur between *links*. For planar mechanisms (i.e., constrained to a single plane of motion), *joints* may be prismatic or revolute, providing one **DOF** of linear or rotational motion, respectively. The synthesis and analysis of such mechanisms is a large

²Model specifications for RC Servos may vary slightly across manufacturers.

³Source: <https://www.pololu.com/category/51/pololu-metal-gearmotors>

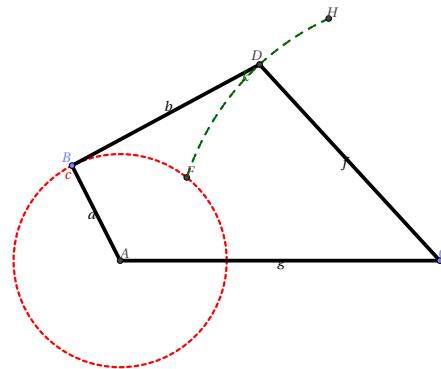


Figure 4.2: Example of simple 4-bar mechanism.

field of research, and many advanced graphical and mathematical techniques are applicable during their design [Slocum 2013].

One of the most common types of linkages is the 4-bar linkage, used in bicycle suspensions, pliers, sliding cranks, and more. Such linkages are composed by 4 links and 4 joints. A *ground* link acts as a reference for the motions of all other links, also referencing the power input to the mechanism, usually a motor or an external link. The input link is called the *rocker* — or *crank* if its motion is continuous/unrestricted — and it is connected via the *coupler* link to the *follower*. The latter is the mechanism's output link. An example of a typical 4-bar linkage is shown in Figure 4.2. In this figure, the segment \overline{AC} defines the ground link. The node A is the mechanism's input — where, for example, a motor could be attached. In this case, \overline{AB} defines a crank, since it can move (rotate) continuously. Its full range of motion is represented by the dashed red line. \overline{BD} is the coupler and \overline{CD} the follower, which generates the \overline{FDH} output arc shown in the dashed green line.

For any planar mechanism, the number of **DOFs** is given by the simple *Gruebler's Equation*: $F = 3(n - 1) - 2f$, where n is the total number of links, f the total number of joints⁴ and F the number of **DOFs** [Slocum 2013]. One quickly verifies that the mechanism in Figure 4.2 ($n = 4$ and $f = 4$) has indeed a single **DOF**. This conclusion is applicable for all 4-bar linkages.

The types of motion in a 4-bar linkage can also be anticipated using the *Grashof* criteria. Let S, L be the lengths of the shortest and longest links respectively, and P, Q the lengths of the remaining two links. If $L + S < P + Q$, then the criteria states that continuous (unrestricted) relative motion between the links is possible [Slocum 2013]. In the previous example, $AB = 2$, $AC = 6$, and $BD = 4$, $CD = 5$, thus enabling the \overline{AB} crank to rotate continuously⁵.

When designing a linkage, one wishes to control the trajectory along which a chosen output node moves. The task of synthesizing a linkage that follows a desired path is called *kinematic synthesis*, and a detailed explanation on the topic is far beyond the scope of this work. For a simple 4-bar linkage as the one shown in Figure 4.2, the output node is point D — the

⁴Some joints count as $\frac{1}{2}$, 1, 2 or 3. Revolute and prismatic joints count as 1.

⁵For any two points X, Y , the notation \overline{XY} refers to the line segment connecting both, and XY refers to this segment's length.

joint between the *follower* and the *coupler*. Since this node can only rotate about point C , its motion is bound to be circular or arc-shaped. An arc or circle can be uniquely specified with a minimum of three points, known as *precision points*. For a 4-bar linkage, *precision points* may be defined beforehand to represent the points of interest which the mechanism should reach.

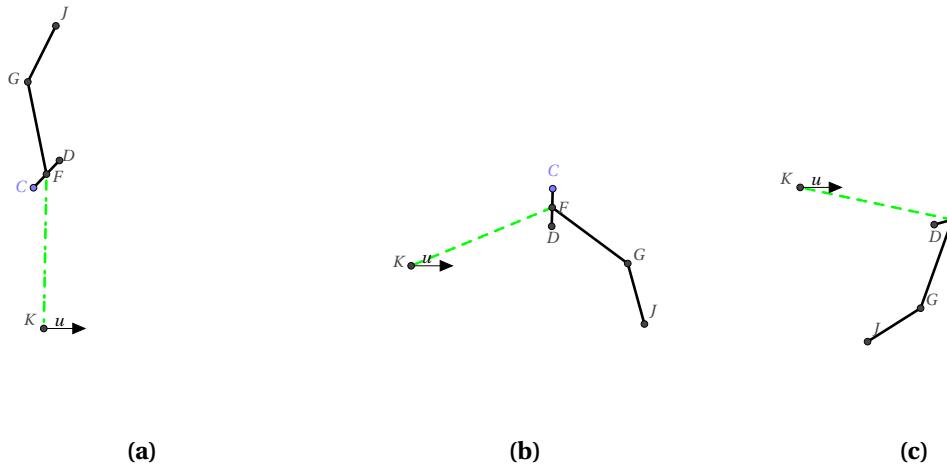


Figure 4.3: Finger poses used to define the fingers precision points. (a) is a pointing position, (b) a finger hook, and (c) represents a fully flexed finger (the **distal phalanx** touching the palm). \vec{u} is normal to the hand's palm.

Based on this concept, three desired finger poses were sketched, as shown in Figure 4.3. Point K represents a pivoting point, close to the **MCP** joint. For the purposes of this sketch only, segment \overline{KF} represents the **proximal phalanx** and point F , the **PIP** joint. Segments \overline{FG} and \overline{GI} illustrate the distal segment of the finger. Line \overline{CD} delineates the base of the finger's distal segment. Lastly, \vec{u} is normal to the hand's palm. Pose (a) depicts a pointing finger, (b) a hook shape (e.g., during a cylindrical **grip**) and (c) the full flexed finger (e.g., when clenching a fist).

Naming point D as the output node, poses (a), (b) and (c) define three *precision points* of its trajectory, specifying an arc $d = \widehat{D_a D_b D_c}$ with center B . Likewise, an arc $c = \widehat{C_a C_b C_c}$ with center A can be designated with the positions of point C in each pose. Segment \overline{AB} defines the mechanism's *ground link*, \overline{AC} , the *rocker*, and \overline{BD} , the *follower*. In the finished mechanism (depicted in Figure 4.4), the point A now defines the **MCP** joint, and the \overline{AC} segment is the **proximal phalanx**. For reference purposes, \vec{u} is again normal to the hand's palm. The ratio between the **PIP** to the **MCP** joints is of 1.49, allowing wider prehensions than the *i-limb* or the *bebionic*. As a result of the larger wrapping ratio, the **distal phalanx** segment \overline{GI} is tilted inwards at 30°.

To apply force on the mechanism's *rocker*, the actuator's shaft is coupled with a short cable to a lever \overline{AN} , as shown in Figure 4.5. When holding an object in an cylindrical **grip**, a loading force f_G is applied to the finger. This force is considered normal to \vec{u} (the vector

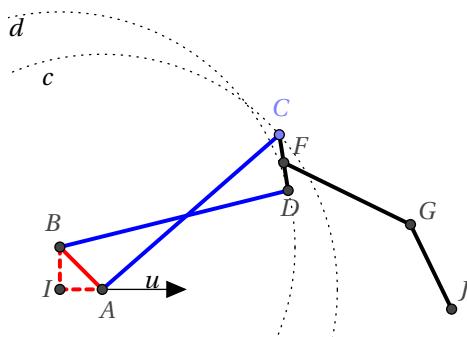


Figure 4.4: Finger's mechanism. Arc c describes the output trajectory.

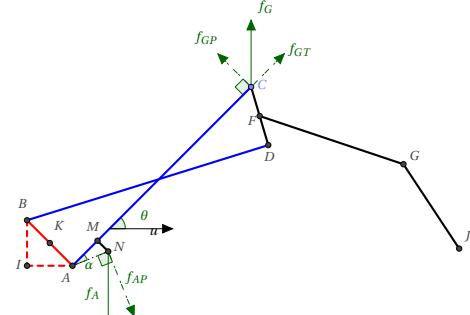


Figure 4.5: \overline{MN} lever and simplified forces in cylindrical **grasp**.

normal to the palm). With the **MCP** at an angle θ in relation to \vec{u} , a simple balance of torques can be written as

$$f_A = f_G \frac{AC}{AN} \frac{\cos\theta}{\cos(\theta - \alpha)} \quad (4.1)$$

where f_A is the total force applied by the actuator (also considered normal to \vec{u}) and α is the angle \overrightarrow{CAN} . In a real situation, many other forces and contact points would exist, but this simple analysis enables a coarse calculation of the forces applied (required) by the actuator. In the actual mechanism $AC = 50mm$, $AN = 7.6mm$ and $\alpha = 23.2^\circ$. The example illustrated in Figure 4.5 has $\theta = 40^\circ$. In this situation, for f_G to be the minimum of $16N$ defined by requirement 1.4⁶, the actuator needs to exert approximately $85N$. Such force can be produced, for example, by coupling the selected motor (with a stall torque of $0.94N.m$) to a pulley with a radius of $1cm$, so the requirement and the mechanism are feasible. Lastly, knowing also that $BD = 50mm$, $AB = 12.73mm$ and $CD = 12mm$, Grashof's criteria yields $AC + CD < BD + AB$. No continuous rotational could be generated with this mechanism — specially because, on a real setting, other constraints restrain the finger's motion.

4.2.3 Issues with backdrivability

The actuation method discussed in the last section proposed the use of a motor coupled to a pulley, pulling the finger's **proximal phalanx** through a cable. This arrangement is called *backdrivable*, since a force applied to the finger in the direction of its extension can directly drive the motor's shaft. While this configuration is theoretically viable, two main issues arise with its usage. The first relates to the finger's extension motion — a cable driven finger can only be flexed, and some sort of spring or elastic element would be needed to extend it back (similar to the *Dextrus Hand V2*).

⁶The requirement poses a minimum force of $16N$ in a **grasp**, i.e., divided between two or more **digits**. The usage of the $16N$ in this example is merely illustrative.

The second and most relevant issue is battery consumption. During a **grasp**, the actuator needs to constantly exert force. Else, the fingers will be extended by any external force or load. Exploring the example presented in the end of the last section, suppose one finger is carrying a 16N load. The actuator's 3.3A stall current would deplete a regular 1000mAh **LiPo** battery in less than 20 minutes. This short battery life is a enormous drawback for any user, and conflicts with requirement 1.6.

This is a common problem when designing artificial hands, and the solution adopted by most of manufacturers is to employ some sort **Non-Backdrivable Mechanism (NBDM)** for the device's actuation. **NBDMs** — sometimes referred to as *non-reversible* mechanisms — prevent external forces from directly driving the actuator or moving the fingers. While this approach has the drawback of requiring an extra mechanism to provide compliance to the finger (as discussed in Section 3.2.2), it permits a **grip** to be maintained without constant actuation. This reduces battery usage and enables some grip patterns (such as a *hook* or a *cylindrical grip*) to be passively support with greater loads than the actuator could bear — e.g., placing the device in a *hook grasp* and then using it to lift a bag.

After this analysis, the chosen DC motor was deemed unsuitable, and it was opted to replace it with some non-backdrivable alternative. The only **NBDM** listed in Table 4.1, *Firgelli's* linear actuator, is not usable in this project due to cost constrains. The solution was thus to design a custom non-backdrivable actuator.

4.2.4 Designing a linear actuator

With no affordable, non-backdrivable actuator readily available, the solution was to design something to fill that gap. Clearly, completely designing a new motor model was not an acceptable alternative, so adapting a commercial product was chosen as strategy. The focus turned once again to standard **RC servos**, shown in Figure 4.6. These inexpensive actuators embed various components in a small enclosure. Figure 4.7 shows a dismantled **servo** and its parts: a **DC motor**, four metal gears (may be plastic in very cheap models), driving and control electronics, a position feedback potentiometer, bearings and metal dowels. Though the materials used in some components may change across manufacturers, **servos** are a commoditized product for radio-controlled devices, so all major brands use very similar (if not identical) components — making all standard **servos** essentially interchangeable.



Figure 4.6: *Left:* a standard **RC servo**. *Right:* a **servo** with its top lid removed, exposing the gearing.

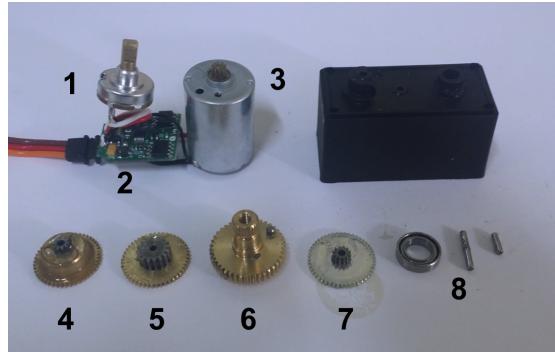


Figure 4.7: A disassembled **RC servo**. **1)** Position feedback potentiometer. **2)** Drive and control electronics. **3)** DC motor. **4,5,6,7)** Reduction gears. **8)** Bearings and metal dowels.

After disassembling a couple of those devices, it seemed viable to build a non-backdrivable **linear actuator** off of its parts. As visible in the images, **servos** use a set of four gears to reduce the high-speed, low-torque rotation of the **DC motor** to a slower but stronger output. Even though the ratios of the gears may change slightly among manufacturers, in general:

1. The **DC motor** drives gear **7** (the *primary gear*) with a $5 : 1$ ratio,
2. Gear **7** drives **4** (the *secondary gear*) with a $4 : 1$ ratio,
3. Gear **4** drives **5** with a $5 : 1$ ratio,
4. Gear **5** drives the output shaft **6** through a $4.5 : 1$ relation

These four levels yield a total reduction of $450 : 1$. The output shaft provides considerable torque, but can usually be manually backdriven when the servo is not powered. This backdrivability could be avoided by replacing the output shaft with a leadscrew. Such screws are widely used in converting rotational motion into linear motion: the displacement of an element coupled to a screw (usually a *nut*) with every turn of the screw is called *pitch*. For small pitches, friction renders most leadscrews non-reversible.

As it turned out, the *secondary gear* (numbered **4** in Figure 4.7) can be disassembled. During fabrication, the smaller gear on top is pressed into the bottom gear, and can be easily removed (Figure 4.8, left). The remaining bore adequately fits a *M3* screw⁷, which can be firmly fixed with a standard *M3* nut (Figure 4.8, right). Allowing another nut to freely translate along the screw while constraining this nut's rotation converts the screw's rotation to linear motion. Albeit not being intended for motion transmission, regular brass screws are very cheap and accessible, and can easily endure the forces foreseen for this application.

The *secondary gear* has a diameter of 16mm , and a standard *M3* screw has a 1.25mm pitch. This produces a $40 : 1$ reduction during linear to rotational motion conversion. Attaching the **servo's DC motor** to the *primary gear* and the modified *secondary gear*, one obtains a total reduction of $800 : 1$ at the leadscrew. Converting the **servo's** rated speed of $0.14\text{s}/60^\circ$ into the new mechanism yields approximately 8.4mm/s . The next step is to calculate the

⁷Metric screw size notation: M3 screws have a nominal outer diameter of 3mm .



Figure 4.8: *Top left:* Second gear in a **servo**'s reduction. *Bottom left:* Removing the small top gear. *Right:* Attaching a M3 screw.

maximum force applicable in this configuration. In a screw, the torque τ_{raise} needed to raise a load is given by the following equation [Bhandari 2007]:

$$\tau_{raise} = \frac{Fd_m}{2} \left(\frac{l + \pi\mu d_m}{\pi d_m - \mu l} \right) \quad (4.2)$$

where F is the load on the screw, d_m the screw's mean diameter, l its pitch (also named *lead*) and μ the coefficient of friction. At the **servo**'s secondary gear, the torque is $1.08/(4.5 \cdot 5) = 0.048Nm$. Rearranging (4.2), one gets:

$$F = 2 \frac{\tau_{raise}}{d_m} \left(\frac{\pi d_m - \mu l}{l + \pi\mu d_m} \right) \quad (4.3)$$

Substituting $\tau_{raise} = 0.048Nm$, $\mu = 0.15$ (given coefficient for brass against steel with machine oil) and $d_m = 0.003m$ yields $F \approx 110N$. This value surpasses the previously chosen actuator and suits the analysis made at the end of Section 4.2.2.

Completing the design of the **linear actuator**, 3D-printable parts were designed to hold all gears and the motor in place, as displayed in 4.9. The depicted model is the final version, which has attachments for a **proximal phalanx** — those details will be discussed in the following subsection. A support bearing was added to the M3 leadscrew, enabling it to rotate freely even when subjected to an external load. An exploded view of the mechanism is visible in Figure 4.10.

Lastly, requirement 1.6 (in Table 3.1) states that the device must be suitable for patients with **wrist disarticulation**. In such case, all the actuators must be contained in the hand's volume, since there is no available space between the prosthesis and the **residual limb** to house them. With this in mind, the main body of the linear actuator (which holds the gears) was designed to have a tiling shape, allowing two or more to be compactly placed side-by-side (Figure 4.11). Two adjacent linear actuators can have threads spaced at a minimum of 17mm, against the 20mm spacing required by the previously chosen DC motor.

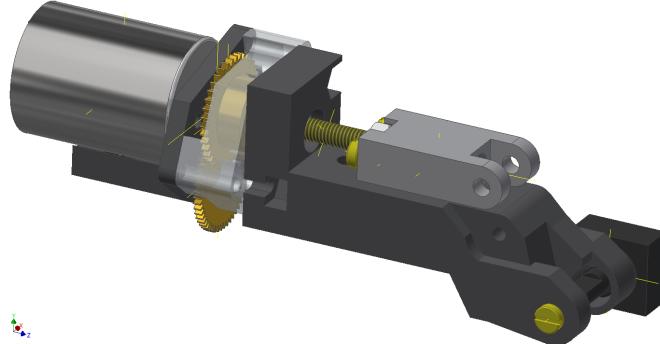


Figure 4.9: **Computer-Aided Design (CAD)** model of the custom linear actuator. One of the parts was set transparent to make the internal gear arrangement visible.

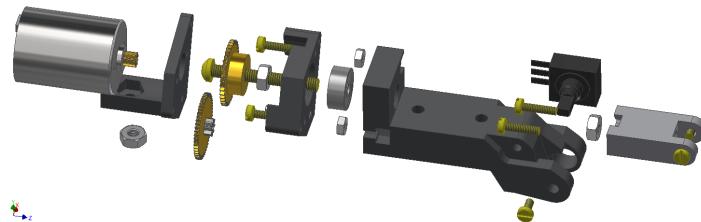


Figure 4.10: Exploded view of the designed linear actuator. On the far right, a part can be seen, that holds a M3 nut and gets displaced according to the screw's rotation.

4.2.5 Modeling the fingers and the palm

Having concluded the design of both the actuator and the fingers' mechanism, everything was modeled and integrated in a **CAD** program. The choice of a suitable **CAD** software was difficult one. In trying to meet requirement 2.6 (in Table 3.1), many different **open source** alternatives were explored. Unfortunately, all softwares tested either lacked too many features or were too unstable to be used reliably. *FreeCAD*⁸ is currently the only fully **open source** parametric **CAD** software available, and the modeling of the fingers was possible up to a certain point (Figure 4.12). However, the absence of a proper assembly module and various bugs made further development on the platform unpractical. *Autodesk Inventor®* was then chosen as **CAD** platform for this project. Though not **open source**, non-commercial licenses allow cost-free use of the software's professional edition for three years.⁹

The first step during modeling was to implement the fingers and the designed mechanism. From Figure 4.5, segment \overline{AC} is implemented with a hollow **proximal phalanx**. Through it, runs a rod (the **back link**) that acts as the \overline{BD} link. Both connect to a single rigid distal segment whose bores at the bottom represent \overline{CD} . The ground link \overline{AB} is implemented by the

⁸<http://www.freecadweb.org/>

⁹The chosen version, *Autodesk Inventor®* 2014 is now outdated, and a two cloud-based, cost-free **CAD** platforms have been unveiled by other companies during the first half of 2015.



Figure 4.11: Tiling shape of the linear actuators allow compact side-by-side positioning (a few parts were suppressed for better visualization).

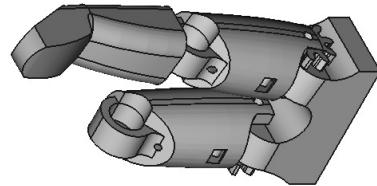


Figure 4.12: Initial model of the index finger and knuckles made with the **open source** *FreeCAD*.

elongated part connected to the **linear actuator** (already seen in Figure 4.10), referred to as the **knuckle** part. The complete designed finger is exhibited in Figure 4.13.

In the diagram of Figure 4.5, the actuator is connected to the \overline{AC} segment via the \overline{MN} lever. In the model, point N corresponds to a bore through all the width of the **proximal phalanx**. However, N describes a circular motion around point A (the **MCP joint**), and cannot be directly coupled to the linear actuator. To solve this, a intermediary linkage was placed between both, highlighted red in Figure 4.14. Due to the chosen position of point N relative to A , this linkage stays mostly parallel to the **linear actuator** during flexion, only tilting slightly during the last stages of finger extension (when large forces are usually not required).

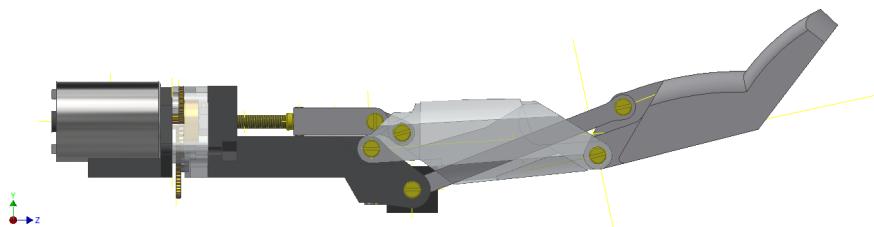


Figure 4.13: Model of index finger, coupled to the **linear actuator**. The **proximal phalanx** was set transparent for better visualization of the finger's mechanism.

The extended finger is already displayed in Figure 4.13. The other two required finger poses defined in Figure 4.3 — a finger hook and fully flexed — are accessible, and can be seen in Figure 4.15. Additionally, to move the fully extended finger to a completely flexed position,

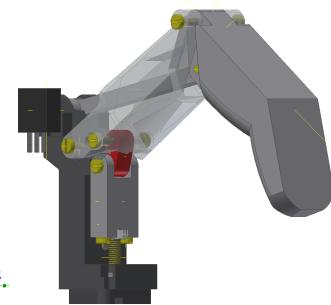


Figure 4.14: Linkage between the **linear actuator** and the **proximal phalanx** highlighted in red.



Figure 4.15: Finger reaching the poses defined during the mechanism design phase. *Left:* Finger in hook position. *Right:* Finger fully flexed.

the **linear actuator** has to travel 10.8mm. Considering the calculated speed of 8.4mm/s, fully flexing an unloaded finger takes approximately 1.3s, a time comparable to the *bebionic* and *i-limb*.

Also seen in Figure 4.14 is a square potentiometer attached (glued) to the **knuckle** part. This is a position feedback potentiometer from a standard **RC servo** that is connected to the finger's **back link**. It enables measurement of the finger's position (since it has a single **DOF**), as defined in requirement 1.11 (in Table 3.1).

It should be pointed out that all parts in both the finger and the **linear actuator** were designed to be 3D-printed in a **FDM** printer. All parts can be printed laying on their sides, so that the main forces to which they are subjected get distributed along the printing planes. This renders the parts considerably stronger. This topic will be discussed more detailedly in the next chapter. Furthermore, with requirement 2.5 in mind, the whole finger design requires only two different sets of screws — *M2* and *M3* — each one in two different lengths, simplifying the assembly process: *M3* screws hold the digits in place and serve as leadscrews, while *M2s* are used in the **linear actuator** and as pivots in the **digits**'s joints.

With the index finger completed, the other fingers were created as its scaled versions: the ring finger is the same size as the index', the middle finger is 10% longer, and the pinky, 20% shorter.

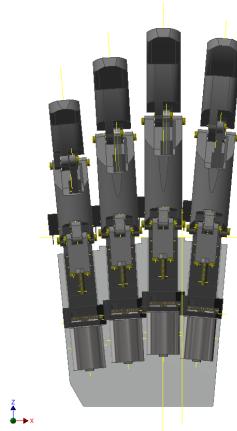


Figure 4.16: Palm's back plate with attached fingers and **linear actuators**. The fixed abduction angle provides a more natural look, and enables thin objects to be held when transitioning from fully extended to fully flexed fingers. The palm is 90mm at its tallest, and 90mm at its widest, dimensions similar to the *bebionic*'s large model.

The subsequent step was to devise a palm to which the fingers would attach. The palm has to withstand all the loads in the fingers, relaying them to the wrist. To achieve enough robustness, it was conceived in two parts: a strong *dorsal* plate that holds all **digits** and a aesthetic *volar* cover, that covers the mechanism and provides a more lifelike appearance. A first iteration of the palm's back plate and the attached fingers is depicted in Figure 4.16. Since the fingers do not provide abduction/adduction motions, a fixed abduction angle was set — a usual choice in commercial devices. The middle finger was chosen as horizontal reference; the index is tilted at 5° from it, the ring finger at 3.5° and the pinky at 5.5°. These angles were chosen empirically to improve appearance and permit a more compact arrangement of actuators in the palm. Such abduction angles, absent in both the *Dextrus Hand V1* and the *InMoov Prosthetic Hand*, enable thin objects to be held between the fingers, such as bank cards and cutlery. Lastly, slots in the palm fit the protruding gears of the **linear actuator**.

As of the writing of this document, the front cover had not yet been designed.

4.2.6 Thumb design and modeling

After devising a first iteration of the palm and the fingers, the thumb remained to be designed. As pointed out in previous sections, the thumb plays a major role in the hand's performance, being responsible for up to 40% of its functionality. In Section 2.2.1, the thumb was modeled with four active **DOFs**. Following the discussion in Section 3.2.3, artificially implementing all these **DOFs** would be too costly and complex. To work around this, most manufacturers implement a 2-**DOF** thumb, that can flex/extend and rotate around a circumduction axis. Being a well tested solution, such approach was also employed in the prototype.

The thumb circumduction axis was chosen based on the observations in [Belter et al. 2013] and on the position of *bebionic*'s thumb, as previously seen in Figure 3.15. In the bottom

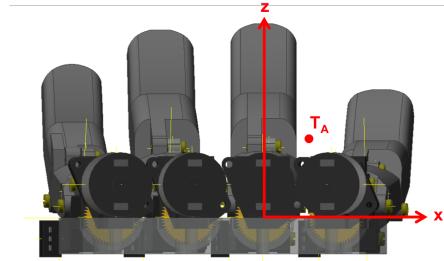


Figure 4.17: Hand's bottom view. The highlighted z axis is normal to the palm and centered on the middle finger, and the x axis is parallel to the internal palm surface. The thumb's circumduction axis runs through point T_A and is perpendicular to the zx -plane.

view of the hand in Figure 4.17, the z axis is normal to the palm and centered on the middle finger, and the x axis is parallel to the internal palm surface. The circumduction axis runs through point T_A at $z = 20\text{mm}$ and $x = 10\text{mm}$, and is perpendicular to the zx -plane.

As a function of the small offset and the type of motion between the circumduction axis and the palm's surface, the designed **linear actuator** could not be used. The forces during thumb circumduction are generally low — only for the thumb's repositioning — and thus the **micro servo** presented in Table 4.1 was chosen as the actuator for this motion. The thumb itself was designed with active motion around the **TIP** joint, using the custom **linear actuator** for flexion/extension — the actuator is positioned as the thumb's **metacarpal** bone. The use of a fixed distal segment (equivalent to the thumb's **proximal** and **distal phalanx**) avoids the need of another 4-bar linkage, turning the thumb into a simple lever. A potentiometer (also from a **RC servo**) is mounted to the side of this distal segment at the **TIP** joint — permitting the thumb's position to be measured. The modeled thumb can be seen in Figure 4.18, with the aforementioned elements highlighted.

As shown in Figure 4.18, the **linear actuator** is attached directly to the base of the thumb — not to the palm. This arrangement would leave the **linear actuator**'s gears would be exposed, which could harm the user. To counter this problem, a variation of the **linear actuator** was created, which fully encloses all the gears, leaving only the leadscrew exposed.¹⁰.

4.2.7 Completing the model

Lastly, the palm was updated to allow the thumb's attachment. With the thumb in place, it needed to be verified whether or not all hand **grips** defined in Figure 2.7 could be reached. Using the **CAD** software, the prototype's **digits** were positioned in order to achieve all the predefined **grasps**, as depicted in Figure 4.19. Both palmar prehensions, the lateral, power and hook **grips** were executed as defined. Tip prehension can be done, but the lacking **DIP** joint in the fingers slightly changes the overall aspect of this **grasp**. Lastly, a spherical **grip** can also be emulated, but the absence of active abduction/adduction in the fingers **MCP** joint also poses a minor modification to the prehension.

¹⁰The leadscrew may harm the user or get tangled in fabric, and an enclosure for it is also planned.

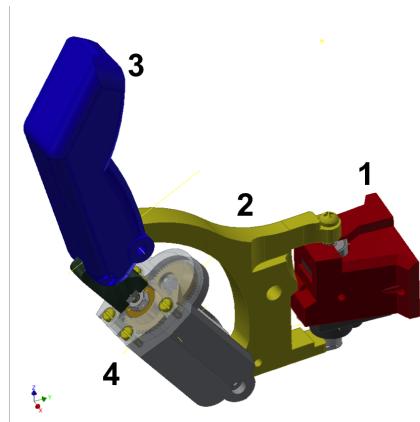


Figure 4.18: Designed thumb. 1) part that attaches the whole thumb to the palm and encloses the **micro servo**. 2) part that holds the **linear actuator** and the finger segment. 3) single fixed distal finger segment. 4) enclosed variation of the **linear actuator** — its top cover was set transparent for visualization of the internal gearing.

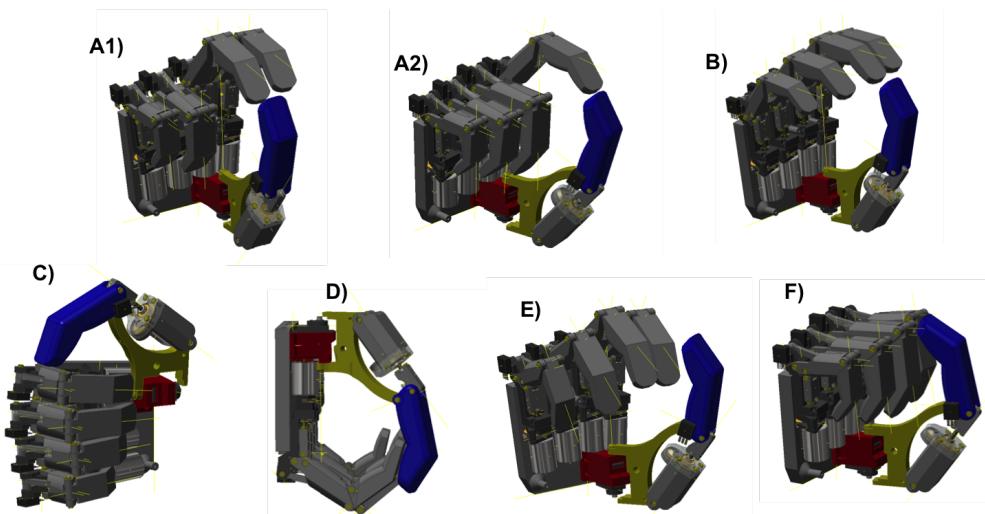


Figure 4.19: Modeled prototype in all the poses previously defined in Figure 2.7. **A1)** Palmar (tripod) prehension. **A2)** Palmar (pinch) prehension. **B)** Tip prehension. **C)** Lateral grip. **D)** Hook grip. **E)** Spherical grip (partially achieved due to the lack of active finger abduction). **F)** Cylindrical (power) grip.

4.3 Electrical design

With the mechanical design completed, the following step is to be able to drive and control the prototype. This requires a electronic control unit to be devised. Such unit needs to drive the **DC** motors in the **linear actuators**, read the prototype's sensors and provide an interface with the user.

Up to this point, the prototype features 6 actuators, and each **digit** is fitted with a potentiometer for position feedback. If any control electronics were to be placed outside the hand, at least 19 independent wires would need to run between them. This would not only hinder the hand's usability, but would reduce its robustness — wires constantly bending between the control unit and the device would break more quickly, eventually causing short-circuits. Thus, housing the electronics in the hand is a suitable option, adopted in all of the devices discussed in Section 3.1. Clearly, embedding all needed control electronics in the hand's volume poses considerable size constraints, and no off-the-shelf compact board existed, that met the needs for the prototype.

A custom control board needed to be created, and this section is devoted to present the steps of its design process. The custom board was designed within *KiCAD*¹¹, a fully open-source and cost-free **Electronics Design Automation (EDA)** software. As already mentioned, the electronics were designed with the usage of a 7.4V power supply in mind. This section begins with a discussion about driving circuitry for **DC** motors, presenting the choice for a commercial motor driver **Integrated Circuit (IC)** to handle the task. In the following, the chosen microcontroller and its setup are also presented, together with the schematics and renderings of the finalized control board.

4.3.1 Motor drivers

The hand contains 5 **DC** motors (apart from the motor enclosed in the *micro servo*), and being able to drive them is essential. In order to drive a motor in both directions, a very common arrangement of switches, known as *H-brige*, is used. As depicted in Figure 4.20, activating only switches S1 and S4 makes current flow from the power source V_{in} through the motor M (the load) from the left to the right, making it rotate. Conversely, activating only switches S2 and S3 makes the current flow from the right to the left, and the M rotates in the opposite direction. Most importantly, pairs S1 and S2 or S3 and S4 should never be activated together — which would short-circuit V_{in} . Nowadays *H-bridges* are rarely implemented with physical switches. In the majority of the applications, S1 to S4 are transistors triggered by some sort of microcontroller.

The switching behavior of *H-bridges* means that the load is either fully on or fully off. Given a power source with a constant voltage, changing the speed with which the motor rotates requires a technique known as **Pulse-Width Modulation (PWM)**. By rapidly turning a set of switches (e.g., S1 and S4S), a series of on/off voltage pulses are applied to the motor. The inductive nature of motors (and other actuators with coils) keeps an approximately constant current flowing through them. The intensity of this current is proportional to the voltage of

¹¹www.kicad-pcb.org

the pulses and the ratio between the on/off time (know as the *duty cycle*), as shown in Figure 4.21.¹²

RC servos, on the other hand, house all the electronics needed to drive the **DC** motor contained in them. However, they are also controlled by a **PWM** signal. By supplying a standard (or a *micro*) **servo** with a modulated pulse, the embedded electronics control its position based on the pulse's duty cycle, as depicted in Figure 4.22.¹³ Generally, the signal has a $20ms$ period, and pulses during $1ms$ to $2ms$ respectively set the servo between its 0° and 180° movement range.

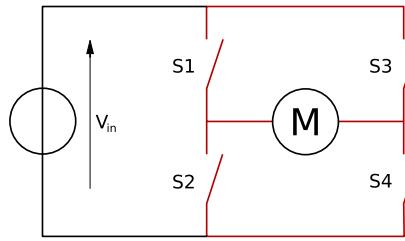


Figure 4.20: Theoretical H-bridge arrangement. Source: [Wikipedia 2015b]

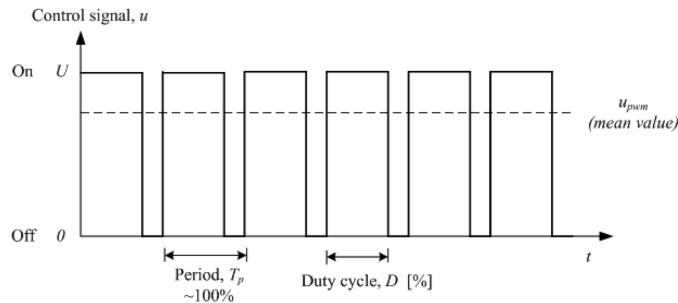


Figure 4.21: A signal with **pulse-width modulation**. The signal is on during D percent of a fixed T_p time period (D is know as the *duty cycle*). As a result, an inductive load "sees" an mean voltage u_{pwm} .

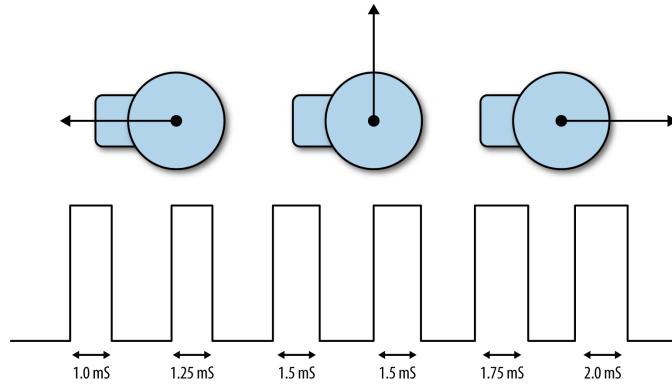
There is a myriad of available, affordable **ICs** that implement robust *H-bridges* and accept **PWM** signals as control inputs, so it would not make sense to build a *H-bridge* out of individual transistors. Some of those **ICs** offer a *sense* pin — an output that allows direct measurement of the current flowing in the circuit. The torque exerted by a **DC** motor is proportional to the current flowing through it, so this feature is desirable in the prototype, according to requirement 1.11. A brief research was conducted in the database of *DigiKey*¹⁴, one of the world's largest suppliers of electronic components. Suitable motor driver **ICs** — with the *sense* feature, and appropriate voltage and current ratings — are listed in Table 4.2.

According to the specifications presented in Table 4.1, each of the prototype's actuators

¹²Image source: www.techteach.no/simview/pwm_control

¹³Image source: <http://razzpisampler.oreilly.com/ch05.html>

¹⁴www.digikey.com

Figure 4.22: Controlling a **servo** with a **PWM** signal.

Name (Reseller)	Voltage	Packaging	Max. A	Channels	Unit Price (U\$)
DRV8801 <i>(Texas Instruments)</i>	5.5 to 40V	QFN HTSSOP	2A (2.8A)	1	3.15
DRV8833 <i>(Texas Instruments)</i>	2.7 to 10.8V	QFN HTSSOP	1.5A (2A)	2	2.73
MC33926 <i>(Freescale Semiconductor)</i>	5 to 28V	PQFN	3A (5A)	1	3.72

Table 4.2: Motor driver **ICs** suitable for the prototype. The *Max. A* column shows the device's maximum continuous current output (per channel), and, in parentheses, maximum peak current. The *Channels* column informs how many individual motors the **IC** can drive. The prices listed are for single units.

can draw over 1A when stalled, so all drivers listed can supply at least 1.5A. The price tags of the listed **ICs** are very similar, so at first glance the MC33926 could be deemed suitable for the application. However, this device is only available in an *PQFN* packaging (picture 4.23). This square packaging is intended for reflow soldering¹⁵, and not for manual soldering. Its pads (the solderable part of an **IC**) face down, and cannot be accessed with a regular soldering iron. According to requirement 2.5 and the **DIY** nature of this project, the usage of the MC33926 was rejected.

Both the DRV8801 and DRV8833 come in a HTSSOP packaging, easier to solder and more compact. During the research of the available **ICs**, the *Metal Gearmotor* was still the decided actuator. Its 3.3A stall current made the DRV8801 the most compatible choice — short of supplying all the current needed, yet stronger than the DRV8833. For that reason, the DRV8801

¹⁵Reflow soldering consists in placing an **IC** on solder paste and heating the setup (both board and atmosphere) until the solder melts. It is a common process in large scale manufacturing, since it allows many components to be soldered simultaneously.

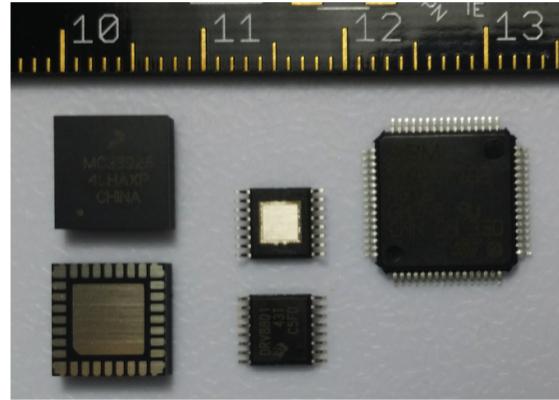


Figure 4.23: A few IC packaging types. *Left:* PQFN package (top and bottom views). *Center:* HTSSOP package (top and bottom views). *Right:* 64-pin LQFP package.

was chosen, and a few units were purchased. Later, with the development of the custom **linear actuator**, the DRV8801 continued to be used due to its immediate availability. In the near future, however, switching to the DRV8833 is planned, due to its reduced price tag and the capability of driving two motors independently.

Proceeding with the design of the board, a minimal circuit was established around the DRV8801, according to the information in its datasheet. Depicted in Figure 4.24, this circuit drives a single **DC** motor, permitting variation in the direction and speed of rotation and outputting an analog signal proportional to the current flowing through the motor's windings. After implementing and testing this minimal circuit (presented in the next chapter), the design was simply replicated 5 times, for each **DC** motor on the prototype.

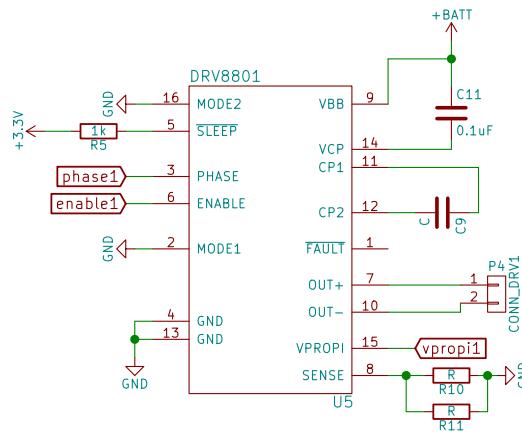


Figure 4.24: Minimal schematic for the DRV8801. The *enable1* input receives a **PWM** signal, varying the motor's rotation speed. *phase1* defines the direction of rotation. *vprop1* outputs a voltage proportional to the current in the IC's internal *H-bridge*.

4.3.2 Microcontroller

To orchestrate the operation of all motor drivers, the **RC servo**, sensors and interface with the user, some sort of microcontroller is required. A microcontroller is an **IC** that embeds a processor, all peripherals it needs to run, and other useful functionalities. Many models include timers, communication modules, **Analog-to-Digital Converters (ADCs)** units, and more.

The well known *Arduino*¹⁶ development boards could have been used with this prototype. However, two main concerns drove the development away from them. Firstly, most *Arduino* boards run on 8-bit microcontrollers. While they are sufficiently powerful to run some complex algorithms, further software development could quickly become restricted due to hardware constraints.¹⁷ Secondly, proper software *debugging* tools hardly exist in the *Arduino* ecosystem — a major drawback when writing embedded software.

After some research, the *STM32F103* from *STMicroelectronics* was chosen. It is a 32-bit **ARM**-based processor from the **Cortex-M3** series, with which the prototype's designer had worked before — more information on it will be provided in the next section. This device is available in an 64-pin LQFP packaging (Figure 4.23, right), which can be soldered manually. Following its datasheet, a minimal working circuit was designed, including a **Single-Wire Debug (SWD)**¹⁸ interface — a hardware and software interface that permits the processor to be programmed and debugged — and a *LM1117*, a voltage regulator that reduces the batteries' 7.4V to 3.3V (the microcontroller's nominal operating).

This minimal circuit was assembled and tested in a prototyping board. In *KiCAD*, the design was then expanded: the *STM32F103* was connected to the 5 *DRV8801*, to the **micro RC servo** and to the potentiometers in each **digit** — using some of the microcontroller's available **PWM**, **ADC** and **General-Purpose Input/Output (GPIO)** lines. An interface to a low-cost **Inertial Measurement Unit (IMU)** was also added, following the designer's interest in researching its possible applications in the prototype. While there is no direct connection between this board and the user (e.g., support for **EMG** sensors), two independent serial communication buses were made available. They allow the board to be connected to an external control source — a computer, smartphone or a microcontroller that interfaces with the user. This grants more flexibility to the design, enabling various user interfaces to be tested. Figure 4.25 shows the schematic of the microcontroller's connections (the 5 motor drivers were suppressed).

With a completed schematic, the board was then routed (i.e., the physical position of the components and their positions were laid out). When finished routing, circuit boards can be sent off for manufacturing — or manufactured with simpler **DIY** techniques. The routed board is shown in Figure 4.26, and has a size of 42.5mm × 58mm. It fits on the prototype's palm, and the symmetry of the connectors allows its use for both left and right hand versions.

¹⁶www.arduino.cc

¹⁷In special, most *Arduino* boards (and their processors) are unsupported by *FreeRTOS*, the real-time embedded operating system chosen for this project. This issue shall be discussed in the next section.

¹⁸Despite the name, **SWD** actually requires at least 5 wires: a data and a clock line, a microcontroller-reset line, a shared ground and a connection to the microcontroller's power source.

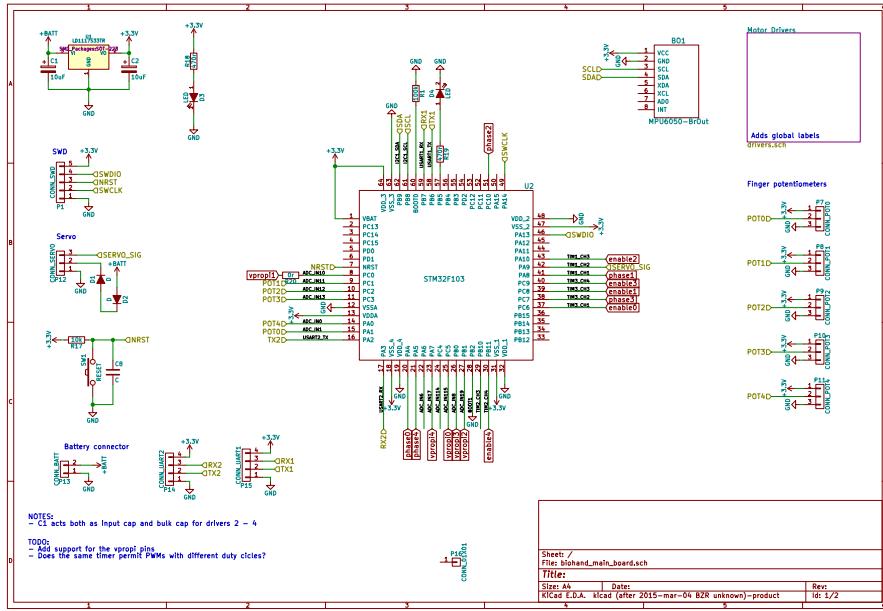


Figure 4.25: Schematic of the STM32F103 and its connections.

4.4 Software outline

This section will provide a more qualitative discussion about the software aspects of this project. At the time of this writing, only a template of the software had been created, compiled and tested. This section will begin addressing further details about the choice of a microcontroller. It will proceed to discuss the tools that are required by it, and how a completely **open source** work environment was set up. Lastly, the above-mentioned software template will be outlined.

4.4.1 On threads and microcontrollers

Choosing a suitable microcontroller to operate a device is intimately related to the end-use intended for that device. In systems that make large use of software — e.g., MP3 players, smartphones, drones —, the overall functionality is very dependent of the capacity and performance of the chosen processor/microcontroller. As such, a first step in choosing an appropriate microcontroller for the proposed prototype was to compose a very simple list of the activities it shall execute during a normal day of use, that are implemented or depend on software:

- Receive and interpret user inputs from a given interface (e.g., **EMG** sensor).
- Perform **grasps** by simultaneously controlling the speed and force at each **digit**.
- Change between grasps.

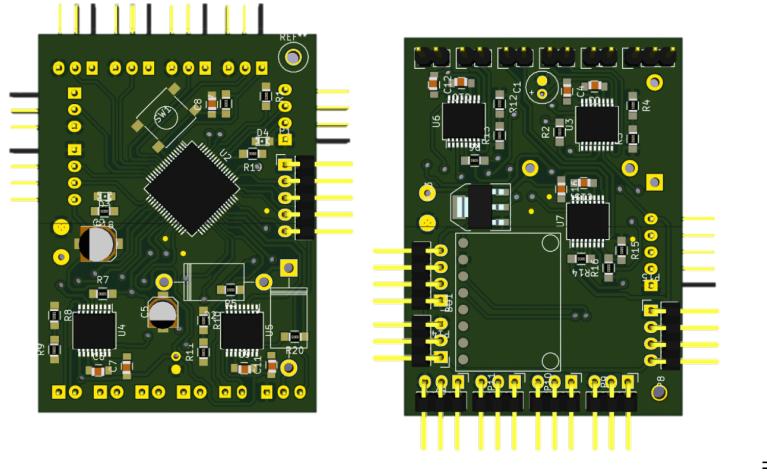


Figure 4.26: The board after routing. *Left:* Top view of the board, showing the processor and two DRV8801. *Right:* Bottom view of the board; the remaining three DRV8801 can be seen.

- Read sensors (potentiometers, sense pins of motor drivers, built-in IMU).
- Use contextual information (battery level, orientation, torques) to adjust grip strength and speed.

This list provides a set of relevant features, but various others are planned. Many of the listed activities are fairly complex and need various actions to occur essentially simultaneously. For example, during a **grip**, potentiometers and motor driver **ICs** of each **digit** have to be measured, and correct **PWM** signals must be generated to ensure sufficient force is applied. This has to happen without loosing track of the user's inputs.

Using a single threaded¹⁹ program to tackle that scenario would be barely feasible, and adding new features later on would be highly difficult, if not impracticable. A microcontroller with a software setup capable of handling multiple threads is thus highly desirable, allowing each of the listed activities to be implemented as one (or more) independent sequence of instructions. Relevant to know, is the fact that this thread handling is done by some sort of *operating system*, that manages the execution of each software thread in a process called *scheduling*. In the designed application, it is crucial that the threads act in a synchronized manner with each other and with the external world — avoiding, for instance, that something gets dropped or crushed due to fingers moving out of sync or sensors being read at wrong moments. To ensure such synchrony, the chosen *operating system* needs to cope with deadlines imposed by the real world, and is thus classified as an **Real-Time Operating System (RTOS)**.

For embedded applications, the *FreeRTOS*²⁰ is very popular, tested and reliable **open source RTOS**. It provides a range of thread scheduling functionalities for a range of microcontrollers and other processors. The massive community and support around it, and the

¹⁹A thread is, simply put, a sequence of programmed instructions.

²⁰www.freertos.org

designer's previous experiences with it, made it the *operating system* of choice for this project. *FreeRTOS* has no official support for *Arduino*-based boards, but strongly supports ARM processors, weighting in for the choice of the *STM32F103*.

ARM stands for *Advanced RISC (Reduced Instruction Set Computer) Machine*, and is a 32-bit processor architecture which is massively employed in the market of embedded devices — being at the core of most modern smartphones. The *Cortex-M* series (available in the M0, M0+, M1, M3, M4 and M7 versions) is the subset of ARM processors tailored for use in microcontrollers. The architecture is licensed for different companies, and the *STM32F103* is a *Cortex-M3* processor manufactured by *STMicroelectronics*, the largest European IC manufacturer.

4.4.2 Development environment

Professional tools for development of embedded software, specially with ARM processors, can be very expensive. Usually intended for product deployment or industrial applications, these tools usually bundle programming and debugging hardware, dedicated libraries and special **Integrated Development Environments (IDEs)**, all under a closed-source regimen. Purchasing one of such tools was not a viable option, due to their restrictive prices and the direct conflict with requirement 2.6. A search thus began for viable **open source** ARM development tools.

When writing code, one needs to convert it into instructions that a processor can execute — the so called machine code. This is carried out by a set of utilities (a compiler, linker, libraries, etc) known as *toolchain*. The act of using one processor (namely, a personal computer) to compile code for another is called *cross-compiling*. For the *STM32F103*, the *GCC ARM Embedded*²¹ was chosen, for being the main **open source** ARM toolchain currently available. The development occurs using the C programming language, inside *Eclipse*²² — a very established, fully **open source IDE**. The *toolchain* and the **IDE** are binded through a set custom plugins, supplied by the *GNU ARM Eclipse*²³ project. These plugins are optional, but provide useful debug functionality for the microcontroller and *Eclipse*.

Once of the code is compiled into a *binary* (a file containing the machine code for the program), it has to be uploaded to the physical processor. On the processor side, interfaces such as **SWD** or **Joint Test Action Group (JTAG)** allow it to receive the program and to be debugged during a program's execution. In order to connect such interfaces to a regular computer, however, a probe is necessary — usually converting **USB** to the chosen protocol.²⁴ Professional probes generally embed all the software to communicate with the target processor, and can be connected seamlessly with their bundled **IDEs**.

This direct connectivity is not available in the chosen software tools. The queries between the computer and the microcontroller have to be run through a generic probe and be orchestrated by another dedicated piece of software. Currently one of the only **open source** options

²¹<https://launchpad.net/gcc-arm-embedded>

²²www.eclipse.org

²³<http://gnuarmeclipse.livius.net/>

²⁴**SWD** and **JTAG** are both hardware and software specifications, defining physical interfaces as well as logical protocols.

available for that task is *OpenOCD*²⁵. It provides both socket and command-line interfaces that enable a user (or another application) to connect to a target processor via a probe, upload software to it, debug its execution, poll the processor's status, halt and resume a program's execution, etc. The aforementioned set plugins (provided by the *GNU ARM Eclipse* project) integrate the *Eclipse IDE* to *OpenOCD*'s functionality. This setup is schematically depicted in Figure 4.27.

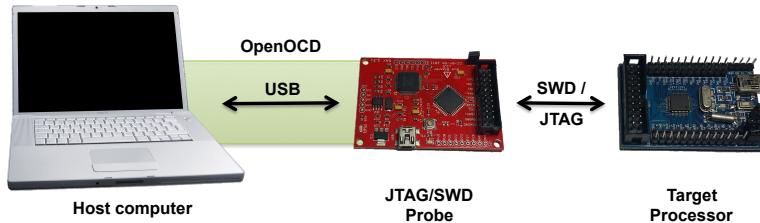


Figure 4.27: Setup for programming and debugging of the processor. The *host computer* uses a **USB probe** that connects it via **SWD** or **JTAG** to the *target processor*. All queries executed by the probe are coordinated by *OpenOCD*, a software being executed on the *host computer*.

Those elements lay out the tools for code development, briefly addressed in the next section. For the majority of the tests and configurations, a development board was employed: the *STM32 Nucleo F103RB*, shown in Figure 4.28. The board features the same 64-pin version of the processor chosen for the prototype's control board, and embeds a **SWD/USB** interface. This interface can be used to program and debug the on-board processor, as well as an external one. The *STM32 Nucleo* board — costing approximately U\$10 — and the outlined software tools supply a very satisfactory low-cost tool set the development of embedded software.

4.4.3 Template Structure

After consolidating the development tools, all necessary libraries and configurations needed for proper coding were condensed into a software template. This template serves as a starting point for any further developments or tests with the chosen microcontroller, and its constitution will be briefly explained.

All *ARM Cortex-M* microcontrollers essentially embed an *ARM Cortex-M* processor core and manufacturer-dependent peripherals (e.g., communication modules, **GPIO** pins, **ADC** units, etc). This common core can be programmed with the aid of a standard software library, the *CMSIS (Cortex Microcontroller Software Interface Standard)*. Each manufacturer then usually supplies *CMSIS*-compatible libraries for their own microcontrollers' peripherals. Figure 4.29 displays a conceptual diagram of the template's structure. At the *Core* layer, *CMSIS ARM* represents the standard processor libraries, and *CMSIS Device*, the libraries for the *STM32F103* peripherals.

On top of the *Core* layer, the *Base* layer provides more advanced libraries and functionalities. The aforementioned *FreeRTOS* is also included in the template, and only depends

²⁵www.openocd.org

on the core *CMSIS* libraries. A custom *Hardware Abstraction Level (HAL)* is being developed, which further abstracts and simplifies the usage of the *CMSIS Device* libraries. Using functionalities provided by *FreeRTOS*, a *Finite State Machine (FSM)* module was also developed and tested. It contains a configurable *finite state automaton (FSA)* and is executed parallel to the main multi-threaded program. The *FSM* keeps track of notifications emitted by other threads and updates the internal *FSA* — which, in turn, modifies the main program's behavior accordingly. Lastly, a set of custom *Project Libraries* bundles commonly used functions for communications, math, data structures, and more.

Being supported by all those elements is the *Application* layer itself, still in heavy development. It contains threads that tackle the activities described in Section 4.4.1 above, and is the part of the software that effectively drives the prototype's behavior. To simplify the interlocking mechanisms between those threads, a physical peripheral may only be accessed by a single thread during the program's execution. Threads interchange information by asynchronous messaging, implemented using basic *FreeRTOS* functions.

Lastly, though the software is comprised of many different elements and threads, all of them are compiled into a single *binary* file that is executed in the *STM32F103*.

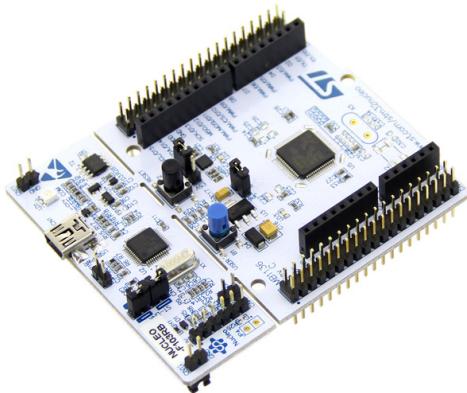


Figure 4.28: The *STM32 Nucleo F103RB*, a development board for the *STM32F103* microcontroller²⁶.

²⁶Source: www.st.com/stm32nucleo

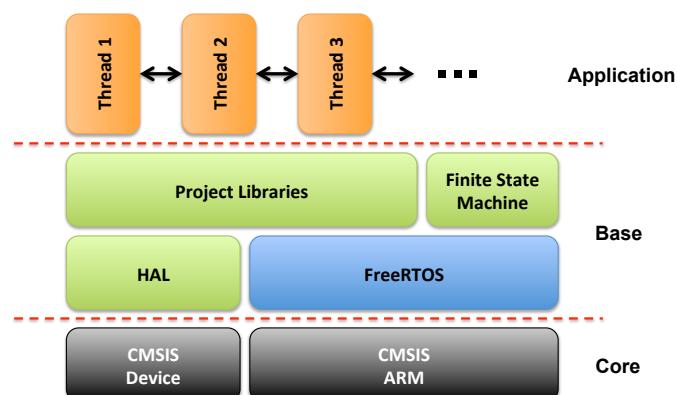


Figure 4.29: Structure of the developed software template.

5

The Prototype and Overall Results

This chapter presents the overall printing and assembly process of the prototype, further discussing its functionality, performance, and results obtained. In Section 5.1, a few aspects about 3D-printing are addressed, and how of its characteristics influenced some design choices. The section continues presenting the key steps of the prototype's assembly. In Section 5.2, the assembled device is tested in terms of weight, speed and strength. The results are compared to some of the commercial devices shown in section 3.1.1, and general discussion on the prototype's functionality closes the chapter.

5.1 Printing and assembling the prototype

This section begins with a brief discussion on the 3D-printing process and the characteristics of the parts it spawns. In the following, the main steps of the prototype's assembly are described and shown in pictures. Lastly, part of the electronics designed in the last chapter are implemented and tested physically.

5.1.1 3D-printing and part orientation

In a very basic sense, traditional milling processes start with a piece of some raw material (e.g., wood, steel, aluminum, etc) and carve it until it acquires a desired shape. Because of this workflow, those processes are referred to as *subtractive* manufacturing. 3D-printing, on the other hand, is classified as *additive* manufacturing. Since this technology emerged in the early 1980s, many different techniques have been developed. However, all of them share a common working principle, in that they create an object by gradually adding together some substrate.

Since major patents related to 3D-printing technology expired less than a decade ago, the market of consumer-grade 3D-printers has seen an enormous expansion. The majority of such desktop devices uses a technique called **fused material deposition (FDM)**. In this approach, a **Computerized Numeric Control (CNC)** machine moves a printing head around, according to a predefined program. This printing head, called extruder, pushes a plastic filament (fed from a spool) into a hot nozzle, that melts and squirts plastic in diameters ranging

from 0.1 mm to 0.5 mm — the plastic re-solidifies very quickly right after being extruded. This extrusion process, controlled by the same program, gradually deposits plastic in various layers, composing the desired object. A commercial 3D-printer in action is shown in Figure 5.1.

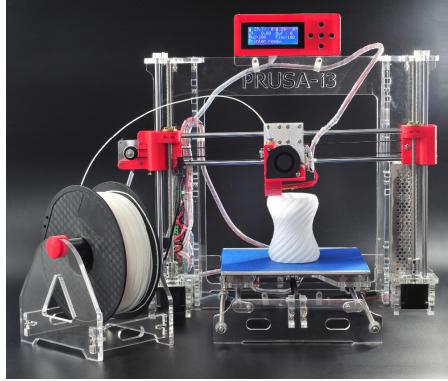


Figure 5.1: Commercial **FDM** printer in action. The filament in the spool is pushed by the extruder into a hot nozzle. The extruded plastic is deposited to form the desired part.

This layered nature of **FDM** printing is a double-edged sword. On the one hand, it allows a mechanically simple device (usually with only 3 **DOFs**) to produce parts with almost arbitrarily complex geometries. Intricate and complicated parts often require regular milling **CNCs** to have several **DOFs**, and sometimes cannot be fabricated by subtractive manufacturing at all. On the other hand, forming an object layer by layer usually means the resulting material properties of the part are not homogeneous, i.e., are not the same in every region of the object. Namely, bonds within a layer are much stronger than inter-layer ones. This means that not only does the shape and design of a part account for its strength, but also the orientation in which it is printed. This effect can be so expressive, that it is well advised to take it into account when designing parts for 3D-printing.

The study in [Sparx Engineering 2015] brings a quantitative measure to the influence that the orientation has on the strength of 3D-printed parts. As depicted in Figure 5.2, test parts were printed in six different orientations, being subsequently subjected to a destructive stress test. Desktop 3D-printers allow one to choose the quantity of material used to fill a part (the *infill* parameter, ranging from hollow to solid), so the tests were executed with different infill settings. For the common choice of 15% infill, print orientations 2 and 4 were at least three times stronger than orientations 1 and 3 — and increasing the infill also increased this difference even further. This confirms that a 3D-printed part resists greater loading when the main shear stress is not directed along the inter-layer interfaces.

Essentially all parts of the prototype were designed with this effect in mind. All phalanges have flat sections on their sides, to be printed "laying on their side" as depicted in Figure 5.3. In that way, the main forces applied on the phalanges at the joints get distributed along the layers. Additionally, parts have a 15% rectilinear infill, as visible in Figure 5.4. This type of infill further aids the distribution of loads in a part.

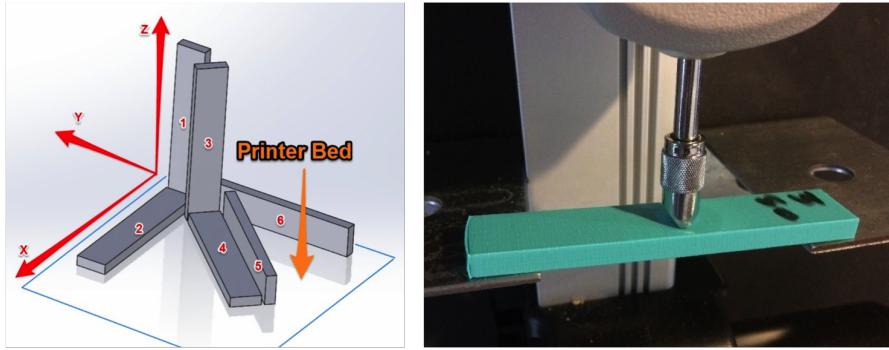


Figure 5.2: *Left:* Test parts were printed in six different orientations. *Right:* Each part was then subjected to a destructive stress test. Source: [Sparx Engineering 2015]

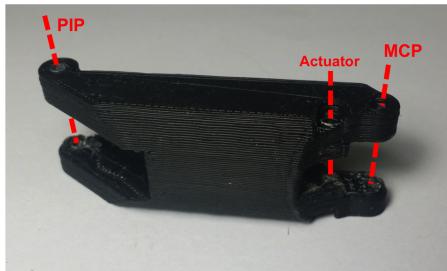


Figure 5.3: The **proximal phalanx** of the index, as well as all phalanges in each **digit** are printed laterally, as in the image. Thus, the forces applied at the **PIP** and **MCP** joints, and at the coupling of the **linear actuator** (all three represented by the dashed lines) get distributed along the print layers.

5.1.2 The build process

3D-printers enable faster development cycles when designing hardware. A new or modified part can be printed and tested within hours at a very reduced cost. These fast iterations were much explored in this project, specially during the modeling phase. Creating accurate parts with **FDM** printers can difficult — dimensions may vary a few tenths of millimeters depending on the nozzle size. To calibrate for, and counter this effect, many parts had to be printed several times, with different clearance and dimensional settings. The majority of those various iterations occurred during the design of the first finger and the **linear actuator** (Figure 5.5, left).

After various adjustments, the first working finger (the index) and a **linear actuator** were assembled, as shown in Figure 5.6. In order to effectively assess its function, a *finger tester* was built for it (Figure 5.5, right). This tester used an *Arduino* connected to a commercial DRV8801 board and to the finger's potentiometer, all mounted on a custom 3D-printed support. Running off a 7.4V **LiPo** battery, this setup was used to verify the functionality, speed and strength of the finger's mechanism — results of these tests will be presented in an upcoming section. After successfully completing some basic tests, the remaining fingers were printed and assembled.

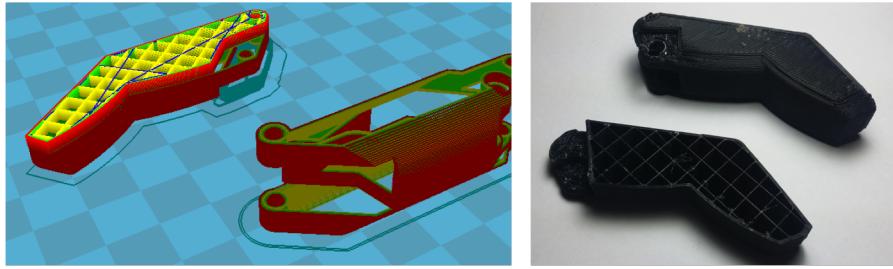


Figure 5.4: *Left:* Distal segment and **proximal phalanx** of the index finger being prepared for 3D-printing in specialized software. Phalanges are printed parallel to the print bed; the infill pattern is purposely exposed for the picture. *Right:* Distal segment of the thumb — a complete and a failed part. The rectilinear infill is visible.

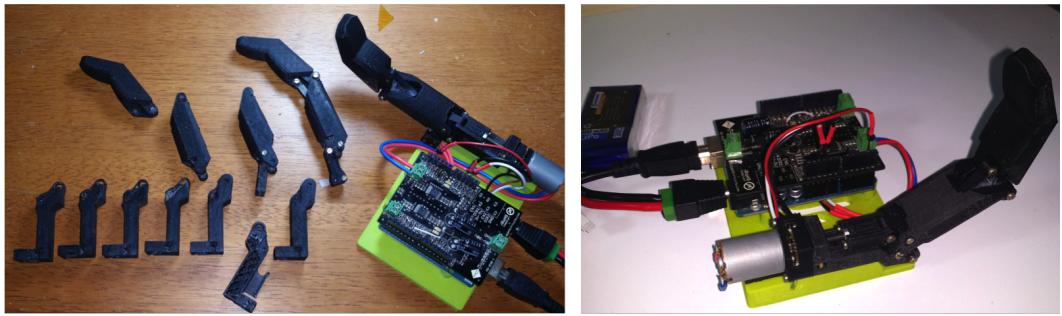


Figure 5.5: *Left:* Various iterations of the **linear actuator's knuckle** parts and the phalanges until the first working finger version, on the far right (mounted on the *finger tester*). *Right:* The *finger tester*, and *Arduino* driving a commercial *DRV8801* board, powered by a 7.4V **LiPo** battery.

Following the assemblage process, the thumb designed in section 4.2.6 was printed and assembled. As once again visible in Figure 5.6, the designed **linear actuator** has a protruding gear on its back. Like previously pointed out, the thumb's linear actuator is housed between the **TCM** and **TIP** joint, with a positioning similar to a **metacarpal** bone. Because of this, the exposed gear of the **linear actuator** would be visible to the user — eventually harming them or getting damaged. The proposed solution was to enclose the gears, as shown in Figure 5.7. The assembled thumb is presented in Figure 5.8.

With all **digits** assembled and working, the palm's back plate was printed. The fingers and the thumb were attached to it, resulting in the arrangement exhibited in Figure 5.9. The presence of an abduction angle is visible. As of the writing, a front cover had not been designed and printed.

5.1.3 Electronics

The process of designing the electronics for the prototype, as described in section 4.3, followed in parallel with various tests. This ensured that schematics drawn into the **EDA** software were indeed functional.



Figure 5.6: First assembled finger and **linear actuator**. The square potentiometer connected to the finger's **back link** can be seen.



Figure 5.7: Detailed view of the thumb's enclosed **linear actuator**.

The minimal circuit for the microcontroller was the first one to be tested in a breadboard¹ setting (Figure 5.10, left). For this test, a 48-pin LQFP STM32F103 microcontroller was soldered to a breakout board². An *Arduino Nano* board supplied 3.3V off its regulator, and an external **SWD** probe tested the functionality of the arrangement. After successfully uploading the software template (discussed in section 4.4) to the microcontroller and blinking a **Light-Emitting Diode (LED)**, the circuit got transferred into the **EDA** software.

Similarly, a minimal design for the DRV8801 **IC** was tested. However, lacking a readily available breakout board for its packaging, a breadboard test could not be performed. Instead, the minimal circuit was conceived in the **EDA** software and routed into a small ($19 \times 25\text{mm}$) test board. The same 3D-printer that printed the parts for the prototype received an adaptation with a cutting tool, enabling it to mill a circuit board's traces in copper. The adapted device than fabricated this test board (Figure 5.10, right), with which the functionality of the minimal circuit could be verified.

Milling the test board was not a trivial task. The HTSSOP packaging of the DRV8801 has sixteen 0.3mm-wide pins, only 0.35mm apart from each other. Very small cutting tools had to be employed, and the cutting depths in the copper had to be precisely adjusted. Adding even more complexity is the fact that the test board is double-sided, requiring milling in both the top and bottom faces. For this to happen properly, an alignment mechanism was implemented. With it, the board can be flipped and milled on both sides, producing matching traces/layers.

After finishing the design of the prototype's main control board described in section 4.3, various attempts were made to mill it. However, being roughly 6 times larger than the DRV8801

¹ Prototyping board for electronics. Metal plated holes allow components to be mounted and interconnected with no soldering.

² Simple board that permits an **IC** with a small packaging to be fit into a prototyping board.



Figure 5.8: Lateral view of the assembled thumb. The square potentiometer attached to the **TIP** joint is visible.

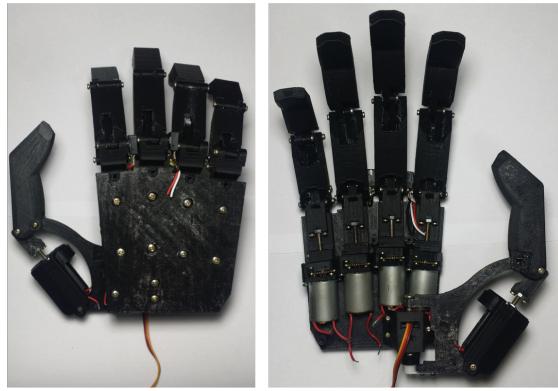


Figure 5.9: **Digits** attached to the palm's back plate. Back and front views.

test board and using even smaller-pitched components (namely, the microcontroller), results were not satisfactory. The design was sent off for manufacturing, and, at the time of this writing, had not been completed.

5.2 Overall results of preliminary tests

After assembling the prototype, it is of crucial to obtain information about its functionality and robustness. Despite the lacking control board, various tests could be performed with the assembled hand, which are exhibited in this section. Initially, considerations on the prototype's weight, speeds and strengths — in comparison to the devices presented in Section 3.1 — are addressed. In the sequel, the reachable **grasps** are tested. Following, the total cost of the prototype is devised. Lastly, the functionality and the results of the prototype are briefly discussed.

5.2.1 Weight, speeds and strengths

The assembled prototype, as shown in Figure 5.9, weights 270g. The control board and the aesthetic front cover are expected to weight a maximum of 35g and 30g, respectively. The

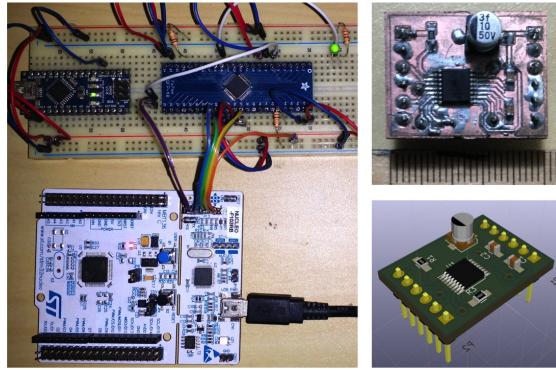


Figure 5.10: *Left:* Breadboard arrangement while testing the microcontroller. An *Arduino* (top left corner) is only used to supply 3.3V to the setup, and a commercial **SWD** probe (white board, bottom) programs and debugs the microcontroller. *Right, bottom:* Routed board with a minimal circuit for the DRV8801. *Right, top:* Functional board with soldered components.

total weight of the prototype, with no batteries, is thus expected to be around the 335g mark. This represents only 54% of the *bebionic*'s weight, and 62% of the *i-limb*'s.

Using the *finger tester* built before, the maximum finger speed could be measured. A minimum of 1.6s were needed to go from full extension to complete flexion (or vice-versa). Over a 90° motion range, this equates to a 56.25°/s speed at the **MCP** joint. According to [Belter et al. 2013], this achieved speed represents 58% of the *bebionic*'s³ maximum speed for the equivalent motion, and 93% of the *i-limb*'s.

Measuring produced **grasp** forces proved to be more difficult. With no main control board to actuate all fingers at once, and no proper load-measuring cell available, an exact value could not be assessed. A simpler force measurement test was executed with a single finger. A regular kitchen scale was placed in the prototype's palm, and the middle finger (10% larger than the index), was flexed until the mechanism stalled with the tip of the finger pressing the scale. Driving the **linear actuator** with only 4V⁴ (instead of the batteries' 7.4V) produced an average force of 6.3N. This value represents 82.5% of the maximum force in a single *i-limb* finger, and 43% of the strongest *bebionic digit* — according to a similar test in [Belter et al. 2013]. In an equivalent setup, the thumb (again running at 4V) exerted 4.8N in a lateral prehension.

As a final robustness test, the prototype was placed in a **power grip** and, with a 3D-printed handle, loaded with 40N, as depicted in Figure 5.11. The setup was manually lifted solely by the base of the palm's back plate during 30 seconds. The handle ensured a more equal force distribution between the fingers. During the experiment, no mechanical instability could be noticed, and the test was completed with no visible sign of damage to the prototype.

³The tests in [Belter et al. 2013] were executed with a *bebionic v2*, a now outdated model. Values may differ for the current *bebionic v3*.

⁴A reduced voltage was used to ensure that the actuators would not be damaged during stall.



Figure 5.11: The prototype is placed in a *power grip*, and statically loaded with 40N. The test was completed with no damages.

5.2.2 Grasps

To further verify the implemented device, different poses were tested for reachability. Without the main control board, each finger was individually powered until the desired configurations were reached. All *power* and *precision grips* defined in Figure 2.7 are indeed reachable — most of them are shown in Figure 5.12. The abduction angles of the fingers were also tested: thin objects can be securely held between them. Lastly, it is worth noting that all the shown poses and the previous static loading test were executed with no power being supplied to the hand. This displays the effectiveness of the **NBDMs** employed in the **digits**.

5.2.3 Final cost

One of the main goals of this project is to devise a prototype that is usable, yet cost-effective. With the informations in tables 4.1 and 4.2, the cost Table 5.1 for the prototype can be elaborated. Prices for ABS plastic were obtained online,⁵ and some estimations were made on the price of remaining hardware elements (screws, glue, etc). Battery prices were consulted at

⁵Source: <http://www.3ders.org/pricecompare/>

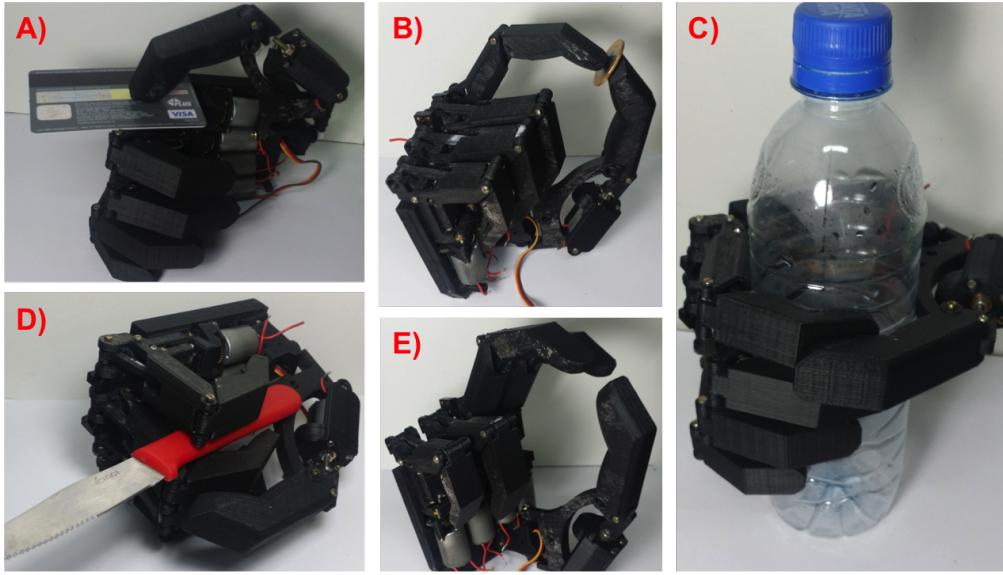


Figure 5.12: Tested **grips**. **A)** Lateral prehension, holding a bank card. **B)** Palmar prehension (*pinch*), holding a coin. **C)** Cylindrical **grip**, holding a water bottle. **D)** Securing an object between the fingers thanks to their abduction angle. **E)** Palmar prehension (*tripod*) pose.

*Amazon*⁶. The price for the control board's fabrication is based on the ratings of the international manufacturer *OSH Park*⁷ (considering an unit cost for a batch of three boards).

A total cost of *U\$132.05* was calculated. This value was devised based on the purchase of individual components (or in very small number). Clearly, the total cost could be further reduced by manufacturing larger batches of the prototype. While this cost is for the materials only (does not foresees the purchase of any tools), it is still at least *200* times lower than any other discussed commercial prosthetic, and half the price of the cheapest **open source** artificial limb.

5.2.4 Overall evaluation of the prototype

The last subsections explored various features of the assembled prototype, drawing relevant conclusions about its functionality. As already mentioned, the prototype still lacks aesthetic covers and a proper **socket** attachment, as both are still being designed. The missing control board — not timely delivered by the supplier — also hindered some tests.

Nonetheless, the prototype showed promising results. As presented in subsection 5.2.1, speeds and strengths achieved are very satisfactory when compared to market-leading prosthetic devices. The prototype successfully implements a minimal set of **grip** motions to execute everyday activities, while being lighter than the average commercial prosthesis. Lastly, the price reduction by a factor of *200* (compared to robotic hands available in the market) is very relevant achievement.

⁶Source: www.amazon.com

⁷www.oshpark.com

Item	Cost per unit (USD)	Quantity	Subtotal
<i>Standard RC servo</i>	12.95	5	64.75
<i>Micro RC servo</i>	9.95	1	9.95
Black ABS plastic	19.19 (1kg)	155g	2.97
Other hardware costs	-	-	5.00
STM32F103	7.14	1	7.14
DRV8801	3.15	5	15.75
Overall board costs	-	-	15.00
2S, 1000mAh LiPo battery	11.49	1	11.49
		Total:	132.05

Table 5.1: Cost table for a complete prototype. The planned final weight was used to calculate the material cost. *Other hardware costs* include the price of screws and glue. *Overall board costs* include fabrication and remaining components.

On the downside, this price reduction comes greatly from the usage of cheaper materials and actuators, which renders the prototype less robust than other commercial prostheses. For instance, while the assembled device endured a 40N loading test, prosthesis like the *bebionic* resist at least ten times this amount in a similar setting. Also, the use of **ABS** plastic as the prototype's main material means that parts will wear out much faster — specially in the joints, where attrition against the hinges will eventually damage the parts. Tests on the prototype's durability are thus needed.

On the other hand, the low-cost aspect of the prototype allows broken and worn out parts to be replaced at a very low cost, with only off-the-shelf tools and products. This, added to the **open source** nature of the design, grants the prototype more flexibility and accessibility than current commercial prostheses.

6

Conclusions

This work described the development process of a prototype low cost hand prosthetic, exploring the use of **open source** tools. This is neither the first nor the last work in this field. Albeit in its infancy, the research of low-cost 3D-printed prosthetics has been showing very promising results, as discussed in previous chapters. While many artificial limbs created in a **maker**-community setting may not be medically regulated or approved at first, there are two main benefits to this initiative. First, many more amputees may try on such **DIY** devices and give their feedback about usability, functionality and cosmetic issues. With this kind of evaluation, and fast development cycles, **open source** prosthetics may quickly become good enough for everyday use, while maintaining a low cost. This effect can trigger a second benefit: a shift in the current prosthetic industry. With an existing user base for those less expensive devices, either leading manufacturers will start providing more affordable solutions, or some new company will.

The prototype presented in this document — friendly codenamed as *biohand* — is a small contribution to this expanding panorama. It is still incomplete, and lacks various features — e.g., aesthetic covers, connection to standard **socket** models, among many others that will be developed in upcoming months. Nonetheless, in its current form, the prototype already fulfills the vast majority of the items defined in the requirement Table 3.1. Lacking are the items 1.3 and 1.7, that couldn't be properly tested or implemented due to the unfinished control board. At the time of this writing, this custom board had not been delivered by its manufacturer. Also, as stated in the requirements, all files necessary to replicate the prototype are hosted online¹ under the *GNU General Public License V3.0*², an **open source** licensing.

Developing a custom **linear actuator** was also crucial for the project. Though it slightly increased the overall build complexity, results obtained during the tests in the above section 5.2 were positive. The prototype was shown to be lighter and (at least) 200 times cheaper than leading prosthetic devices, while maintaining at least half of their speed and strength performances.

Clearly, the massive cost reduction comes at the expense of reduced robustness. The **ABS** plastic used in the prototype's structure and the **servos** are much less durable and resistant

¹<https://github.com/MartinBloedorn/biohand>

²<http://www.gnu.org/licenses/gpl-3.0.en.html>

than the various metal alloys and custom high-end motors employed in commercial devices. The prototype's approach on the cost/durability aspect spells a different philosophy inherent to current **DIY** prosthetics: instead of a single, expensive and all-enduring device, prosthetics become accessible, dynamic and personal as a biological limb. Assembling a new finger for the prototype costs less than U\$15, so replacing broken or worn-out parts becomes accessible.

The prototype's parts were printed on a custom 3D-printer that was built with this project in mind. This printer — detailed in Appendix A — features milling capabilities and a larger form factor, usually not available in commercial-grade machines. However, all parts in the prototype were designed to be 3D-printable even on regular commercial desktop printers. This characteristic, together with the usage of only off-the-shelf tools and components, ensures that this design can be cost-effectively replicated anywhere.

Naturally, in the prototype's current form, fixing or replacing parts is still a fairly technical issue, that requires some minimal knowledge on how to operate a 3D-printer. However, as both prosthetics and 3D-printers evolve, swapping out a broken part may become as simple as installing an ink cartridge in a regular inkjet printer.

Lastly, the prototype's designer believes that, once pending design aspects are completed, the *biohand* will be indeed suited for a less intensive, daily use. It serves as a proof-of-concept that viable prosthetic devices can be created at a fraction of the cost of current commercial and **open source** ones.

6.1 Future works

Much of the main aspects of the prototype's design are complete and were implemented, however, as roughly outlined in the previous section, many features are lacking. While some of them are more aesthetic, most of them are vital for complete functionality. The work on this prototype shall thus be continued.

Firstly and foremost, the control board of the prototype needs to be concluded and assembled. This will not only open up the possibility of driving all **digits** simultaneously (making **grasp** test easier and more effective), but will also permit software development to evolve. With an open path for software to be written, user interfaces may be implemented and tested (initially with some sort of computer-based interface, then with regular **EMG** electrodes).

The palm is also far from being complete. Currently, it has no mounting holes or fixations, and designing an attachment for at least one type of **socket** is a priority in the near future. A proper attachment will permit more precise load-bearing measurements, and will enable amputees to wear it with their existing sockets. Designing and printing a frontal (*volar*) enclosure for the hand is also foreseen at this point.

Other smaller improvements are also planned. Coating the fingers with rubber or a similar material will greatly improve **grip** stability. An alternative design for the thumb will be evaluated, changing the active **DOF** from the **TIP** to the **TCM** joint. Additionally, a future version of the control board shall use the cheaper and more compact DRV8833 motor driver **ICs** (instead of the current DRV8801).

With this most basic steps completed, proper usage tests with amputees are necessary.

Clearly, coordinating a large test with dozens of individuals is not feasible in the circumstances of the project, but having at least a couple of users assess the prototype's functionality is crucial. Their experience with the prototype will provide valuable feedback to direct further developments.

Finally, there are many tests and improvements that lie beyond the intent of a single designer. Like other **open source** initiatives, this project not only aims at creating a more accessible device, but also at empowering its user to make the changes and improvements that they see fit. Prosthetic devices are intended to aid individuals with missing limbs, by making daily tasks less ponderous. However, habits, desires and routines vary largely from one person to another. Amputees wearing similar devices may have vastly different experiences, often facing challenges that not even an experienced designer can anticipate. By giving the user power over what their device is capable of (beyond purely sawing and grinding it), one can expect the field to evolve much faster. After all, prosthetics are about people, and prostheses should better reflect that notion.

Appendix A

Designing and Building a 3D-Printer

A.1 Motivation

The discussions in sections 1.1 and 4.1.2 point out that much of the interest in this project's topic emerged inspired by similar **open source** developments. All of those initiatives explored the use of 3D-printers in fabricating customized low-cost prosthetics, and this work trailed an equivalent path. To do so, having a 3D-printer was an obvious requirement.

The first small steps and research that led up to this work started in late 2013. At that time, no Brazilian reseller supplied 3D-printers at a satisfactory cost/benefit ratio. Simple wooden printers in national territory would cost as much as high-end devices abroad. After brief online investigation, it seemed viable to design and build a much more complete model at a lower price tag.

The personal interest in completely designing and assembling a **CNC** machine also played a major role. Beyond the intent to build only a 3D-printer, the goal quickly became to — if viable — assemble a multi-functional machine. It was also desired to accommodate basic *milling* functionalities, so that the device could fabricate circuit boards and carve soft materials (such some woods and plastics).

This following appendix is thus dedicated at outlining the main steps of development and assembly of a custom **FDM** 3D-printer, subsequently used in the manufacturing of the prototype prosthetic hand described in the previous chapters of this document. If needed, a concise explanation on the **FDM** 3D-printing process is given in section 5.1.1.

A.2 Mechanical design

With those goals in mind, a quick survey took place, on the main characteristics in **CNCs**. In a very broad sense, **CNCs** machines use computerized controls to drive actuators that position some tool in space. Due to their simplicity, *cartesian* setups are often preferred. In such, the tool moves along (usually) three straight axes, all perpendicular to each other. Even in such setups, there are major differences between how 3D-printers and **CNC** mills are arranged — differences that greatly reflect the distinct end-activities of each one.

While 3D-printing a part, the print head is moved around, extruding plastic when needed. It meets very low (if any) opposing forces, and moves as quickly as possible to complete the process. Current desktop 3D-printers usually print at 60mm/s , and may do non-printing moves at twice that speed. The construction of these machines reflects such characteristics: the design is optimized to reduce the mass of the moving elements as much as possible, enabling more aggressive acceleration profiles and higher speeds, while reducing the wobble propagated through the *frame* (the machine's structure). The frame is thus made as light as possible.

On the other end, CNC mills are designed with heavy-duty carving and cutting operations in mind. The moving tool in these machines is a specialized electric motor (usually referred to as *spindle*) fitted with a cutting tool (drill, milling bit, etc). Such elements have direct contact with the materials being cut/carved, and experience high opposing forces. To keep a precise positioning of the cut tool, the frame carrying it may not deform under these forces, so rigidity is crucial. This is achieved by assembling more robust machines, with stronger materials, rendering the overall device much heavier than a regular 3D-printer.

Designing a device capable of doing both tasks (even if only lighter milling operations are planned for) is finding a suitable compromise between its structure's rigidity and a reduced moving inertia. With that in mind, the machine shown in Figure A.1 was created. It was conceived in a **CAD** tool, using off-the-shelf aluminum profiles, linear rails and fixations. Custom parts were designed to be laser-cut out of 3mm -thick aluminum sheets.

As better visible on the **CAD** rendering on the left, the machine is composed of three moving elements. At the base of the structure, is a square plate — the *bed* — that moves in the x direction, upon which parts are printed or milled. Moving up and down in the z direction is the *gantry*, which carries the *y carriage*. A tool of interest (spindle or extruder) is mounted to the latter.

Since 3D-printing is done in successive layers, the z motion is taken in small, sparse steps. The *gantry*'s design takes advantage of this fact; it holds still during any given layer of a print, and only the *bed* and the *carriage* move around. These two moving elements have low masses in comparison to the torque of the chosen actuators (discussed in the next section). Furthermore, all moving elements are mounted onto cylinder-shaped bearings that slide along the device's 12mm -thick linear rails — enabling a low-friction displacement.

This arrangement is not novel. Many well known consumer 3D-printers models — such as the *Prusa I3*, in Figure A.2 — explore the same configuration. This custom built 3D-printer, however, has a working volume (space reachable by the tool) of $35\text{cm} \times 35\text{cm} \times 21\text{cm}$, more than three times larger than the average desktop printer. The metal frame also ensures a rigid structure, theoretically enabling the desired milling operations. The finalized device is shown in Figure A.1 (right).

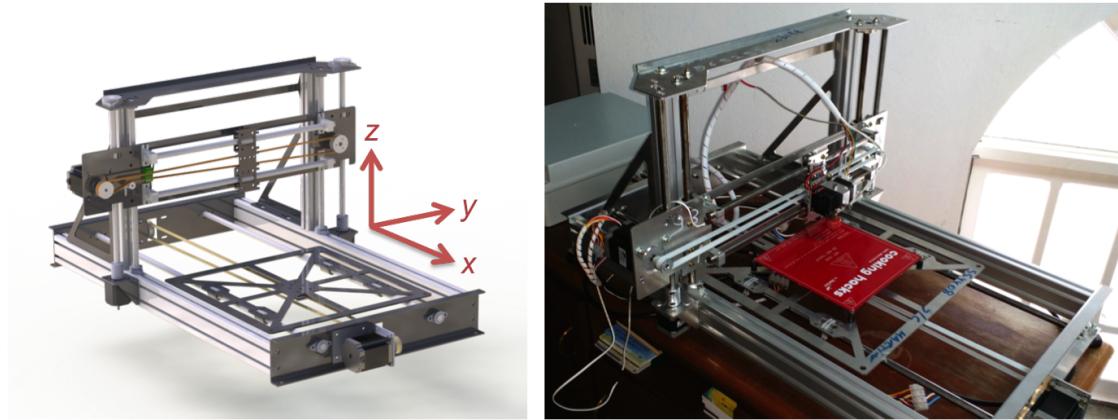


Figure A.1: *Left:* CAD rendering of the custom designed 3D-printer, with highlighted axis directions. *Right:* Assembled and functional printer.



Figure A.2: A commercial 3D-printer, the *Prusa i3*, using the same cartesian arrangement of the custom designed machine.

A.3 Electronics setup

A.3.1 Choice of actuators

For applications where precise positioning is required, *stepper motors* are very often the actuator of choice. As visible in the disassembled unit in Figure A.3 (left), these motors use various coils circularly arranged around a toothed rotating core. When a pair of opposing coils is energized, the teeth in the core align to the generated magnetic field. Rapidly disabling one pair of coils and energizing the adjacent pair makes the core move into the next magnetically stable configuration — hence the *stepping* characteristic of its motion. *Stepper motors* are unable to rotate as fast as the majority of the **DC** motors, and drain a constant current regardless of their loading. Nonetheless, they can produce a great amount of torque at low rotations and move in very precise increments.

The various coils in these motors are usually connected into four groups. Figure A.3 (right) shows a simple schematic of the motor's coils. Various arrangements of these groups exist when connecting them to driving electronics — such as *unipolar*, *bipolar series* or *bipolar parallel* —, but these will not be discussed in this document. The *bipolar parallel* configuration was chosen for all motors in the machine, since it provides the greatest amount of torque possible. In this configuration, coils are connected in two pairs: $A+$ to $A-$ and $B+$ to $B-$. Alternating the direction of the current flow in each of those pairs, following a suiting order, rotates the core. For this current flow alternation, two *H-bridges* (like the ones presented in section 4.3.1) are needed — one for each pair.

Most desktop 3D-printers use small stepper motors in a standardized form factor, called *NEMA 17*.¹ In the custom 3D-printer, two of such *NEMA 17* are used to move the *gantry* with the aid of two *M12* screws² (that act as leadscrews). For the x and y motions, the larger *NEMA 23* stepper motors were chosen. These motors are capable of producing holding torques of $1.9Nm$ (in the *bipolar parallel* configuration), while draining $4.2A$.

In total, the custom 3D-printer uses two *NEMA 23* and three *NEMA 17* motors (two for the *gantry*'s z motion, and one for pushing the plastic filament into the extruder).

A.3.2 Control board and motor drivers

Single *NEMA 17* motors can be driven with dedicated **ICs**, such as the *A498* or the *DRV8825*. These **ICs** embed both *H-bridges* and logic controls required to drive a stepper motor in a *bipolar* configuration. However, the high current requirements of the chosen *NEMA 23* motors required a more powerful solution. For driving these motors, three *Powerlolu*³ boards were purchased. This **open hardware** boards can drive stepper motors up to $50V$ and $10A$, sufficient for the application. The x and y motors were each connected to an individual *Powerlolu*, and both z motors were connected to the remaining third board.

To orchestrate the motion of all motors during 3D-printing, an *Arduino Mega*⁴ board was chosen. With the advent of **open source** desktop 3D-printers, the *Arduino* boards have become almost a *de facto* standard. Various software and hardware tools for 3D-printing are oriented towards these devices, making them easy to work with. In special, the *RAMPS*⁵ board can be directly connected to an *Arduino Mega*, and serves as an hardware interface to the aforementioned motor drive electronics. Beyond that, other elements needed in 3D-printer — such as a heated bed, fans and a extruder — can be directly attached to it. Figure A.4 shows the connections between the *RAMPS* board and the elements in the 3D-printer. The whole system is driven off a $12V$ **DC** power supply, which is connected directly to the mains power. All the electronics were mounted into a proper enclosure, as shown in Figure A.5.

¹ *NEMA* stands for *National Electrical Manufacturers Association*, who standardized the form factor.

² Metric threads with a nominal diameter of $12mm$.

³ <https://github.com/fluidfred/powerlolu>

⁴ <https://www.arduino.cc/en/Main/arduinoBoardMega>

⁵ http://www.reprap.org/wiki/RAMPS_1.4

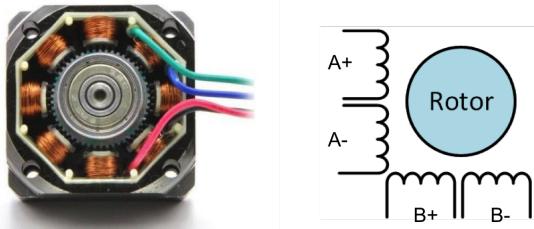


Figure A.3: *Left:* Disassembled stepper motor. The coils and around the spinning core can be easily seen. *Right:* Schematic view of the coils in the motor.

A.4 Software setup and results

For the actual 3D-printing to take place, three independent **open source** softwares were used. The first one is *Marlin*⁶, the firmware programmed into the *Arduino* board. It enables the *Arduino* to receive specialized commands (the *g-code*) via its **USB** connection, that specify how the tool (i.e., the extruder) should be moved. *Marlin* then interprets these commands, and triggers the appropriate signals for the corresponding motions to be performed by the stepper motors. It also triggers and controls the heated bed, fans, extruders, etc.

The instructions describing how to print a part are generated by another program, called *Cura*⁷. This program receives a 3D model of an object as input, slices it into layers and generates a *toolpath* for each layer. All paths are combined in a single *g-code* file. This file then needs to be streamed in parts to the *Arduino* board, while the printing is happening. The third software used in this setup, *Pronterface*⁸, is responsible for this streaming and is executed on a dedicated notebook connected via **USB** to the *Arduino* board.

After completing the assembly of all mechanical and electrical elements, as well as configuring the required software parts, the custom 3D-printer was fully functional. One of the parts produced by the machine is shown in Figure A.6.

⁶<https://github.com/MarlinFirmware/Marlin>

⁷<https://ultimaker.com/en/products/cura-software>

⁸www.pronterface.com

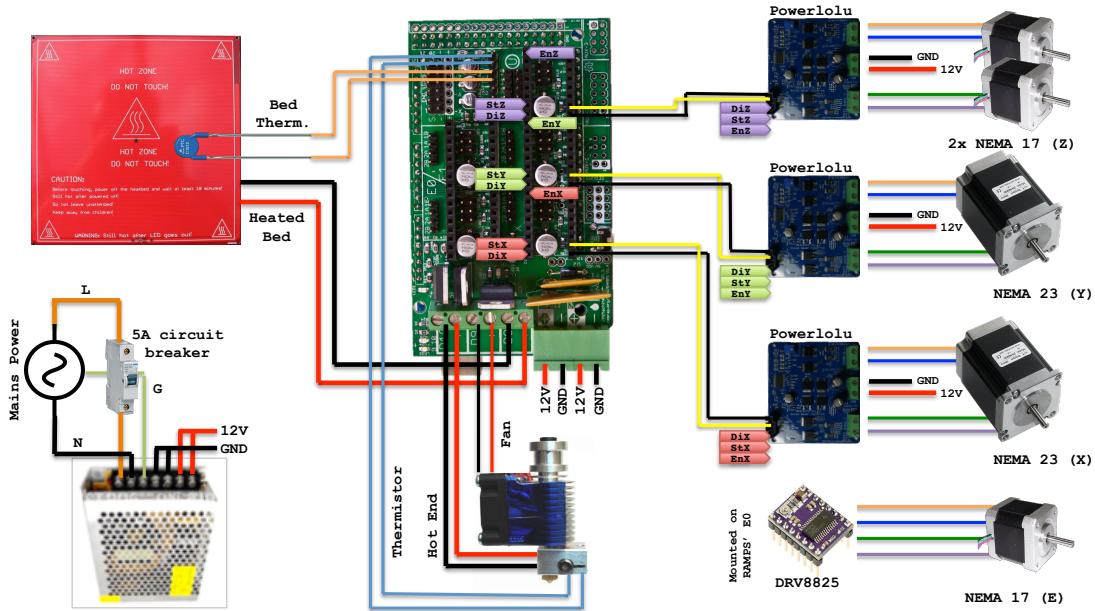


Figure A.4: Electrical connections of the elements in the 3D-printer. In the center, the *RAMPS* board, connected to an *Arduino Mega*. The *RAMPS* drives the three *Powerlolu* boards and a single *DRV8825* (for the extruder's motor), on the right. The extruder's hot end is displayed in the bottom-center of the image. On the top-left corner, the heated bed and a thermistor (that reads the bed's temperature) is depicted. The whole system runs off a 12V power supply, shown in the bottom-left corner of the image.



Figure A.5: Assembled electronics in a wall-mounted enclosure.



Figure A.6: One of the parts produced with the custom 3D-printer.

Bibliography

- ACA. *Amputee Coalition of America. Coverage of Prosthetic Devices*. 2011. URL: <http://www.amputee-coalition.org/advocacy-awareness-docs/documents/state-studies/maryland-full-study.pdf>.
- Belter, J. T. et al. "Mechanical design and performance specifications of anthropomorphic prosthetic hands: A review". In: *Journal of Rehabilitation Research & Development* (2013).
- Bergman, K. et al. "Functional benefit of an adaptive myoelectric prosthetic hand compared to a conventional myoelectric hand". In: *Prosthetics and Orthotics International* (1992).
- Bhandari, V. *Design of Machine Elements*. 2007.
- Bundhoo, V. and E.J. Park. "Design of an Artificial Muscle Actuated Finger towards Biomimetic Prosthetic Hands". In: *12th International Conference on Advanced Robotics* (2005).
- Criswell, E. *Cram's Introduction to Surface Electromyography*. 2011.
- Feix, T. et al. "A comprehensive grasp taxonomy". In: *Robotics, Science And Systems Conference: Workshop On Understanding The Human Hand For Advancing Robotic Manipulation* (2009).
- Heckathorne, C. W. "Upper-Limb Prosthetics: Components for Adult Externally Powered Systems". In: *Atlas of Limb Prosthetics: Surgical, Prosthetic, and Rehabilitation Principles* (1992).
- HSS. *Hospital for Special Surgery. The Cost of a new Limb can add up over a lifetime*. 2015. URL: http://www.hss.edu/newsroom_prosthetic-leg-cost-over-lifetime.asp (visited on 07/21/2015).
- Kang Li, M., C. I-Ming, and S. H. Yeo. "Development of finger-motion capturing device based on optical linear encoder". In: *Journal of Rehabilitation Research & Development* (2011).
- Kuiken, T. A. et al. "Targeted Muscle Reinnervation for Real-time Myoelectric Control of Multifunction Artificial Arms". In: *The Journal of the American Medical Association* (2009).
- Langevin, Gael. *InMoov blog*. 2015. URL: <http://www.inmoov.fr/> (visited on 07/18/2015).
- Lauer, L. T, P. H. Peckham, and K. L. Kilgore. "EEG-based control of a hand grasp neuroprosthesis". In: *NeuroReport* (1999).
- McGimpsey, G. and T. C. Bradford. *Limb Prosthetics Services and Devices*. Worcester Polytechnic Institution, 2009.

- Motherboard. *Man Hands*. 2015. URL: <http://motherboard.vice.com/read/man-hands> (visited on 07/18/2015).
- Open Bionics. *Open Bionics company site*. 2015. URL: <http://www.openbionics.com> (visited on 07/18/2015).
- Ottobock. *Michelangelo Technician Product Brochure*. 2013. URL: http://www.leben-mit-michelangelo.de/fileadmin/downloads/techniker/english/technician_product_brochure.pdf (visited on 07/17/2015).
- Pendleton, H. M. and W. Schultz-Krohn. *Pedretti's Occupational Therapy: Practice Skills for Physical Dysfunction*. 2012.
- Pylatiuk, C., S. Schulz, and L. Döderlein. "Results of an Internet survey of myoelectric prosthetic hand users". In: *Prosthetics and Orthotics International* (2007).
- RSLSteeper. *BeBionic 3 Technical Information*. 2015. URL: http://bebionic.com/distributor/documents/bebionic3_Tech_Manual_web.pdf (visited on 07/16/2015).
- Sanei, S. and J. A. Chambers. *EEG Signal Processing*. 2007.
- Slocum, A. *Fundamentals of Design*. 2013.
- Smaby, N. et al. "Identification of key pinch forces required to complete functional tasks". In: *Journal of Rehabilitation Research & Development* (2004).
- Sobotta, J. and R. Putz. *Sobotta Atlas of Human Anatomy*. 2006.
- Sollerman, C. and A. Ejeskär. "Sollerman Hand Function Test: A Standardised Method and its Use in Tetraplegic Patients". In: *Scand J Plast Reconstr Hand Surg* (1995).
- Sommerville, I. *Software Engineering*. 1995.
- Sparx Engineering. *3D Print Orientation and the effect on part strength*. 2015. URL: <http://www.sparxeng.com/blog/mechanical/3d-print-orientation-effect-part-strength> (visited on 07/29/2015).
- Srinivasan, A. "Doc Ock - Development of Filtration Techniques to Facilitate Accurate Pattern Detection in EEG Signals". In: *Google Science Fair* (2011).
- Touch Bionics. *i-Limb Ultra Datasheet*. 2014. URL: http://www.touchbionics.com/sites/default/files/ma00001a_i-limb_ultra_datasheet.pdf (visited on 07/17/2015).
- Weir, R. F. *Design Of Artificial Arms and Hands for Prosthetic Applications*. 2003.
- Wikipedia. *Activities of daily living*. 2015. URL: https://en.wikipedia.org/wiki/Activities_of_daily_living (visited on 07/15/2015).
- Wikipedia. *Hbridge*. 2015. URL: https://en.wikipedia.org/wiki/H_bridge (visited on 07/29/2015).
- Wikipedia. *Hand*. 2015. URL: <https://en.wikipedia.org/wiki/Hand> (visited on 07/10/2015).
- Wikipedia. *Prosthesis*. 2015. URL: <https://en.wikipedia.org/wiki/Prosthesis> (visited on 08/04/2015).