Politecnico di Torino

MSc Degree in Mechatronic Engineering
A.Y. 2022/2023

Master's Thesis

Study and mechatronic design of a powered lower limb exoskeleton

---

Candidate Luca Bizzotto
ID: S268911

Mentor Prof. Alessandro Rizzo
DET - Dipartimento di elettronica e telecomunicazioni
Politecnico di Torino

Supervisors Prof. Emanuele Menegatti
Dr. Stefano Tortora
DEI - Dipartimento di ingegneria dell'informazione
Università degli Studi di Padova

> *Success is not final, failure is not fatal, it is the courage to continue that counts.*
>
> — **Sir Winston Churchill**
> Statesman

Abstract

LOWER limb exoskeletons are today an excellent rehabilitation device in patients who, due to an injury (e.g., stroke or spinal cord) or due to age, have reduced gait mobility. It cannot be ruled out that in the near future, in addition to rehabilitation, they may be used in everyday life to return to regular walking.

Up to now, there are various models on market that can be used for the latter purpose, or for the purposes of research and rehabilitation, however they are generally expensive and therefore, not very accessible. This thesis aims to develop an active lower limb exoskeleton, taking inspiration from an existing low-cost and open source exoskeleton called ALICE.

After analyzing the characteristics of such an exoskeleton, we will then move on to the selection of components to improve its performance, always with the aim of keeping costs low and maintaining its constructive simplicity. To do this, solutions will be considered that are attentive to widely used fields of application, different from the one under consideration, adapting them to their own needs.

The low-level control of the individual joint is being implemented using the designated components. Afterwards, the performance of these products will be validated to assess their suitability for use in the exoskeleton under development. To do so, the knee gait profile of a healthy subject will be replicated with such a prototype robotic joint, with an experimental setup created for the purpose. The performance will then be discussed, leaving room for possible future development of the project.

Contents

List of Figures

List of Listings

List of Tables

1 Introduction

> *We are survival machines – robot vehicles blindly programmed to preserve the selfish molecules known as genes.*
>
> — **Richard Dawkins**
> Biologist

An exoskeleton (EXO) is a particular type of mobile robots in the category of the wearable robots (WR), that is externally worn from a human user in order to improve the mobility. It can be described as a complex mechatronic system composed of structural mechanical parts moved through actuators, governed by electronic circuits that implement complex control algorithms through sensors' measurement. These electronic (or electromechanical) devices are connected to each other through a dense network of cables (or pipes) that allow the supply and transport of information between them.

The aim of motor improvement can come from a workload need (industrial or military EXO) and therefore to simplify and help the person who wears it, or in healthcare (medical EXO) that is to overcome motor impairments resulting from traumas, diseases or physical aging. An application example of these devices in the described areas is shown in figure 1.1.

Supporting argumentative on the application of exoskeletons in these two macro fields, as reported in [1] which state that over 25% of Europeans experience back injury can be linked to work, and workplace injuries cost European countries up to 4% of their gross national product. The German Federal Institute for Occupational Safety and Health states that those type of injuries account for 23% of the sick days in Germany and lead to an estimated loss in production of € 10 billions and an annual gross loss of € 17 billions.

In this moment in which this thesis is being written, it is recent news that here in Italy, Esselunga (a retail food chain) is experimenting with the use of exoskeletons produced by Comau (a Turin-based robotics company) on its warehouse workers to improve the well-being of operators and prevent injuries [2].

Outside of the workplace, the ratio of people over 65 years touch 17% in the European Union in 2011 and is predicted to rise approximately at 30% in 2060 [3]. This trend is empathized by the low birth rate and the high life expectancy in these countries. This brings to an increasing need to assist elderly people in their daily lives, particularly about their mobility and autonomy [4].

In addition, people affected by stroke are about 13 million every year and it is one of the main cause of serious long-term disability in the world [5]. Most victims suffer from neurological and sensory-motor deficits, so they need a period of rehabilitation in order to achieve functional independence [6]. In this context, hemiparesis is a manifestation of stroke that affects only a side of the body, and commonly impacts gait that is one of the most important activities for daily independence.

Another critical condition that leads to motor difficulties (and in particular the gait) is the spinal cord Injury (SCI) which often leads to permanent paraplegia, use of wheelchair, and several secondary clinical complications. It is estimated that every year, around the world, between 250 and 500 thousand people suffer a SCI [7]. Two-thirds of SCI patients are estimated to be paraplegic; most patients with SCI are young men in their thirties, who need to work to support their families [8].

From the above statements, it can be understood the importance that these exoskeletons have and will have more in the future for the mobility support and the motor rehabilitation of the subjects just described before. It also shows the importance of the participation of governments and companies that have economic and ethical interests in the continuous improvement of existing technology and the reduction of costs in order to make this technology appealing to more and more people.

Unfortunately, due to the different entities that leads to motor deficiency and the subjectivity of each single case, the exoskeleton's functions change in order to face with those specific problems, which translates into the difficulty of finding universal solutions suitable to be industrialized.

1.1 Biomechanics of human locomotion

In this specific application, the human and the WR coexist and must submit to each other's mutually imposed constraints. The human body has several joints and therefore enjoys many degrees of freedom, which the WR tries to replicate, within the limits of technical implementations. From the human point of view (especially in the case of people with residual mobility) the WR is seen as an impediment to the natural walking process. On the other hand, the WR control considers the human as a source of disturbance (especially in the case of predefined gaits) that tries to hinder the target that it pursues.

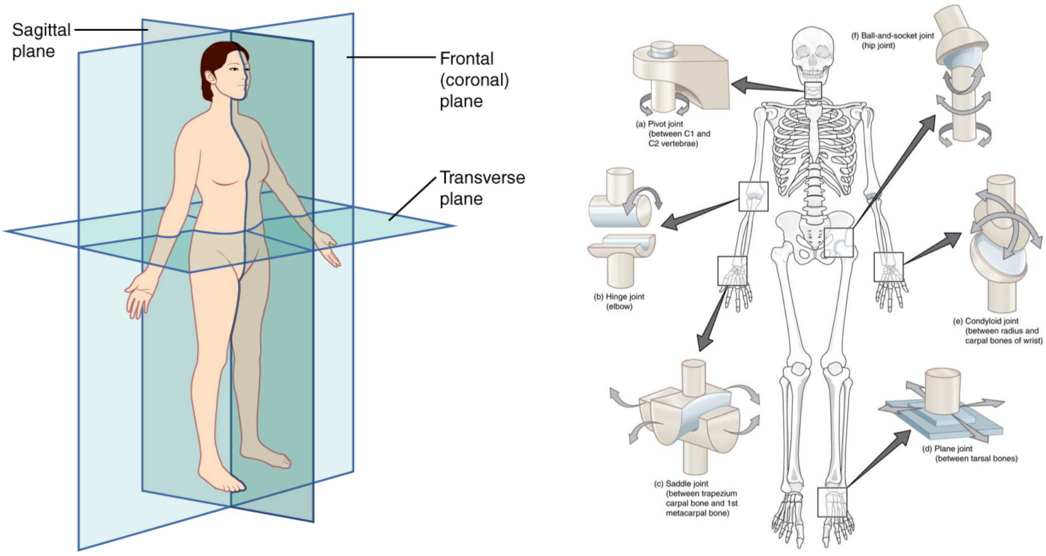
This mutual collaboration is one of the main focuses of research today in this area, and translates into various aspects, such as wearability, communication of human

(a) *Comau Mate*

(b) *Berkeley (Ekso bionics) HULC*

(c) *Ekso bionics eLegs*

Figure 1.1: Examples of exoskeleton [9, 10].



(a) Anatomical body planes

(b) Synovial joints of the body

Figure 1.2: Kinematic point of view of the human body [15].

intentions to the WR (via HMI) and the prediction by the WR of human intentions [11, 12] (via, for example, vision systems [13]) that are increasingly translated into gait cycles that do not follow only a predefined pattern but that adapt dynamically depending on the context [14, 13]. It is therefore of fundamental importance to understand the anatomy of the human body and also the structure of the WR and to do so, in this paragraph and the following we will try to see the basic concepts of them.

1.1.1 Body movements

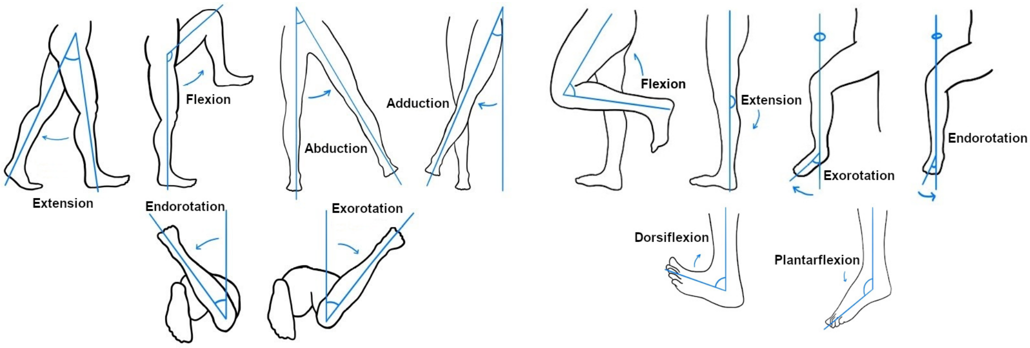
As a first analysis of the anatomy of the human body, from the figure 1.2a we can outline three planes on which the main movements occur, which are [15] :

Sagittal Plane divides the body vertically into right and left sides.

Frontal Plane divides the body vertically into an anterior and a posterior portion.

Transverse Plane divides the body horizontally into an upper and a lower portion.

The nomenclature of these planes comes from the medical field to indicate the sections useful for understanding the anatomy of the body. In this application, they assume the utility of describing the motion of the body in the space. This description occurs through the use of positions, velocities and linear and angular accelerations of each limb, which can be reported to the local reference system of the body (for



(a) Hip movements (b) Knee and ankle movements

Figure 1.3: Movements of the lower limb human body [16].

example the one just seen in the figure 1.2a) and then to an absolute and global reference system (Useful, as will be seen, in describing the gait).

Often in this application, it is useful to represent the body with a projection on the sagittal plane, thus simplifying the kinematics of the body in an articulated motion only on this plane, having as schematization rotoidal joints and rigid links for the human body.

The complete description of the joints of the human body is shown in figure 1.2b where we can outline six different types of joints, specifically called synovial joints. These joints are [15]:

Pivot joints allow for rotation around an axis, such as between the first and second cervical vertebrae, which allows for side-to-side rotation of the head.

Hinge joint it works like a door hinge, the elbow is such joint and also the knee can be simplified as this type of joint.

Saddle joint articulation between the trapezium carpal bone and the first metacarpal bone at the base of the thumb.

Plane joints such as those between the tarsal bones of the foot, allow for limited gliding movements between bones.

Condyloid joint one example is the radiocarpal joint of the wrist.

Ball-and-socket joint hip and shoulder joints are the only ones joints of this type.

In the specific case of lower limb EXO the joints of interest are therefore the hinge joint for the ankle as well as the knee and ball-and-socket joint for the hip. As we will see, most of the existing exoskeletons try to replicate these joints through actuators that actually produce a mono-axial rotation. This simplification may be acceptable in the case of the ankle and knee, but is more limiting for the hip, as this simplification affects the reproduction of the gait, which is different from the natural one.

For completeness, we can see in figure 1.3 the nomenclature used for the main movements that the joints just mentioned before allow. The movements of main interest are hip and knee flexion/extension (Rotations in the sagittal plane).

Secondly, if the EXO is equipped with such degrees of freedom, Plantaflexion/dorsiflexion (Rotations in the sagittal plane) and abduction/adduction (Rotations in the frontal plane). We will then see as explained in the following chapter, how an EXO can be rigid/flexible and therefore can allow or not the movements of exorotation and endorotation (Rotations in the transverse plane) in a passive way.

1.1.2 Gait cycle

For gait cycle is meant the natural sequence of events that occur between the touch of a foot (right or left) on the ground and the next touch, of the same foot, always on the ground. Specifically, this event of the beginning of the cycle is called initial contact or heel strike. The gait cycle results in the movement of the subject with minimal expenditure of energy. This cycle is influenced by the anatomical shape of the subject and is therefore, subjective.

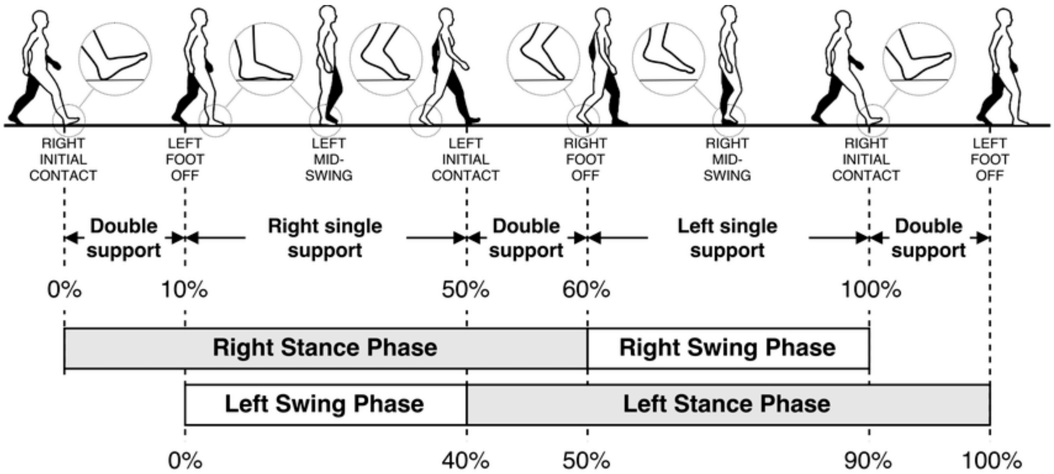
The intermediate phases of the gait cycle are shown in figure 1.4a where for each foot it is possible to distinguish two macro phases [17, 18]:

Stance phase starts with the first touch of a foot and ends when the same foot is lifted off the ground. This phase lasts about 60% of the gait cycle

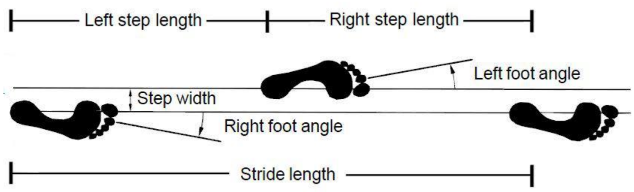
Swing phase it is the phase in which such a foot is in mid air, it begins when the foot is lifted from the ground and ends once the same foot touches the ground again. This phase lasts approximately 40% of the gait cycle.

In addition, we can further divide these two phases into **double support** (both feet are in contact with the ground) and **single support** (only one foot is in contact with the ground). Note that the double support phase in a complete cycle occupies only 20% of the total time and therefore in terms of control of the WR is a major challenge to replicate the gait and try to balance a system that is inherently unstable. As a supplement to the divisions seen, other subdivisions can be found in the literature that are intended to refine the cycle analysis [18] but are not reported here.

To conclude the analysis, the figure 1.4b shows the projection on the transverse plane of the complete gait cycle, in which we can denote the following main parameters:



(a) Gait cycle phases projected on the sagittal plane



(b) Gait cycle parameters projected on the transverse plane

Figure 1.4: Gait cycle description [19, 20].

Stride length is the distance between the two initial contact points of the same foot consecutive or equivalently, the sum of two step length of two different feet, always using as reference the initial contact point.

Step length is the distance between the initial contact points of two opposite feet.

Step width is the distance measured between the center of the heel of each foot during the support phase.

Walking rate is measured as the number of steps per minutes.

Velocity is the product between the step length and the walking rate, and it is commonly measured as the distance covered in one minute.

1.2 The wearable robot

1.2.1 Classification

Exoskeletons in recent decades have experienced multiple developments, given also the ever-increasing demand and consequent scientific research. Given the countless works that have been presented in the literature, in [21] it is intended to give a universal classification that can group such works together, so that the differences among them can be studied in a systematic way.

Such a procedure could also come in handy for eventual standardization or at any rate lead to some guidelines that researchers and companies can follow, reducing the scattering of works already present and consequently starting from an established basis for eventual improved development.

In [21], 5244 scientific articles, patents, and commercial products were evaluated over a time frame from 2016 to 2020 from which 75 devices were selected after screening.

The resulting general classification proposes several categories of possible exoskeletons: by structure, by body part focus, by action, by power technology, for purpose and finally for application area.

The schematic result of the review is shown in figure 1.5.

Body part focused exoskeletons can be designed for specific body parts, such as one hand, one leg, or the entire body. The most common body parts for exoskeletons are: full-body, which assist all or most of the body; upper body, which involve the chest, head, back, and/or shoulders; and lower body, which involve the thighs, lower legs, and/or hips. There are also classes for specific limbs and joints, such as the knee, ankle, hand, arm, foot, etc. Additionally, there is a special class for any

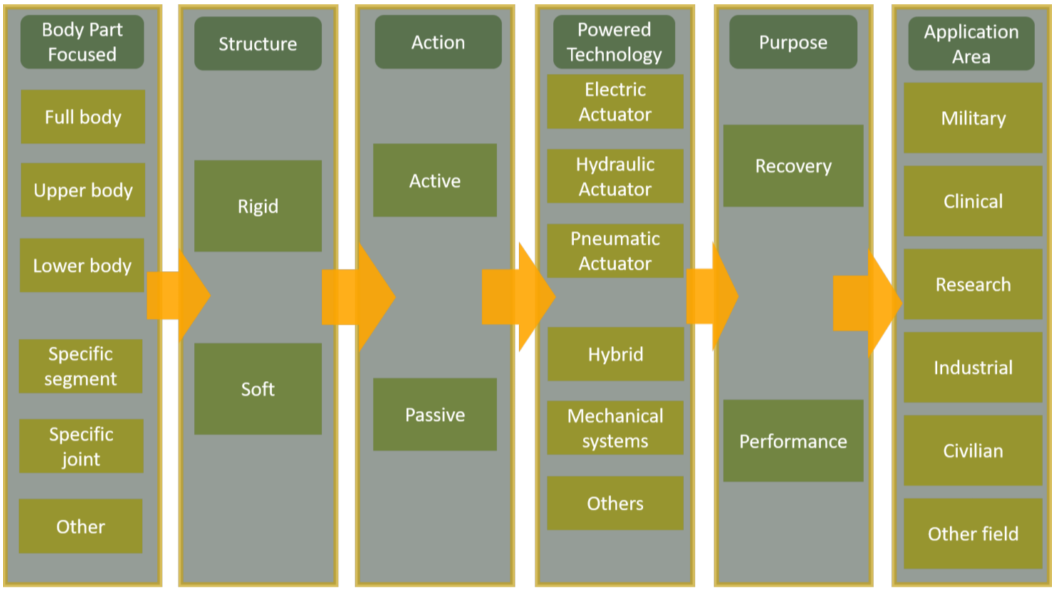


Figure 1.5: Exoskeletons classifications model from [21].

other EXO that is not included in the previous classes. The most common body part focused class is the lower body, which holds 56% of the total reviewed material in the cited article.

Structure exoskeletons can be differentiated by their structure, resulting in two classes: rigid and soft. Rigid EXOs are made with hard materials such as metals, plastics, and fibers, while soft exoskeletons, also known as exo-suits, are made with materials that allow for free movement, often textiles. The majority of EXOs reviewed are rigid, with 89% of the total.

Action exoskeletons can also be categorized by the type of assistance they provide to the user, into active and passive categories. Active EXOs are equipped with actuators that move the user's body parts without the need for the user to apply energy. Passive EXOs, on the other hand, do not have actuators and rely on the user to perform the movement, but they facilitate the user's movement through passive elements such as springs or cables. The most common category is active EXOs, with 83% of the total market share.

Powered Technology powered exoskeletons are separated into four main classes: electric, hydraulic, pneumatic, and mechanical. A separate class is designated for hybrid EXOs that use a combination of two or more of the main classes. Electric actuators are powered by any kind of electric motor, hydraulic and pneumatic actuators use pistons and soft actuators, and mechanical systems store or transmit mechanical energy through devices such as springs, dampers, pulleys, or gears. The most common hybrids are those that combine electrical actuators with mechanical systems, such as series elastic actuators (SEAs) which use a combination of electric actuators and sprung mechanical systems.

Purpose exoskeletons can be categorized by their intended use into recovery and performance classes. Recovery EXOs are used for rehabilitation, while performance EXOs are used for assistance. Almost all exoskeleton applications that are not specifically for rehabilitation are considered performance EXOs. In this case, the distribution is more balanced, with 44% in the performance class and 23% in the recovery class.

Application area the final category of exoskeleton classification is the application area for which the EXO was made. Each exoskeleton may belong to one or more class. The military class includes EXOs used for activities involving the army, navy, airforce, or any other military branch. The medical class includes EXOs used in clinical or medical settings, including recovery EXOs. The research class includes exoskeletons that are currently in the research and development phase. The industrial class includes exoskeletons made specifically for industrial activities, designed to prevent long-term physical damage for people without any pathology. The civilian class includes recovery or performance EXOs intended for use in homes or public spaces, to aid in activities of daily living. The most frequent class is research, holding 33% of the entire market share, indicating that the development of EXOs is heavily dependent on the research field.

After looking at the general classification of an exoskeleton, the configuration of the specific classes that will denote the category of EXO considered in this thesis, as the basis for the designing of an improved version of it, will now be defined. This EXO will be for the lower body, rigid, active, rehabilitation purpose and as the main application target that of research or possibly clinical.

Therefore, the structure of such a category of EXO will be analyzed, going into the specifics of powered technologies, which is necessary for an improvement analysis and a subsequent correct design choice for the new version of exoskeleton and the control techniques of such technologies.

To support this dissection, similarly to what was seen earlier for the general classification model [21], the figure 1.6 is used and the various parts of it are explained below.

Rotary or linear actuators this component is the core of an active exoskeleton. It comprises an electric motor coupled with a mechanical transmission. The latter, has the purpose of changing the ratio of input/output torque (or input/output speed) and in some cases transforms the type of motion, for example from rotary to

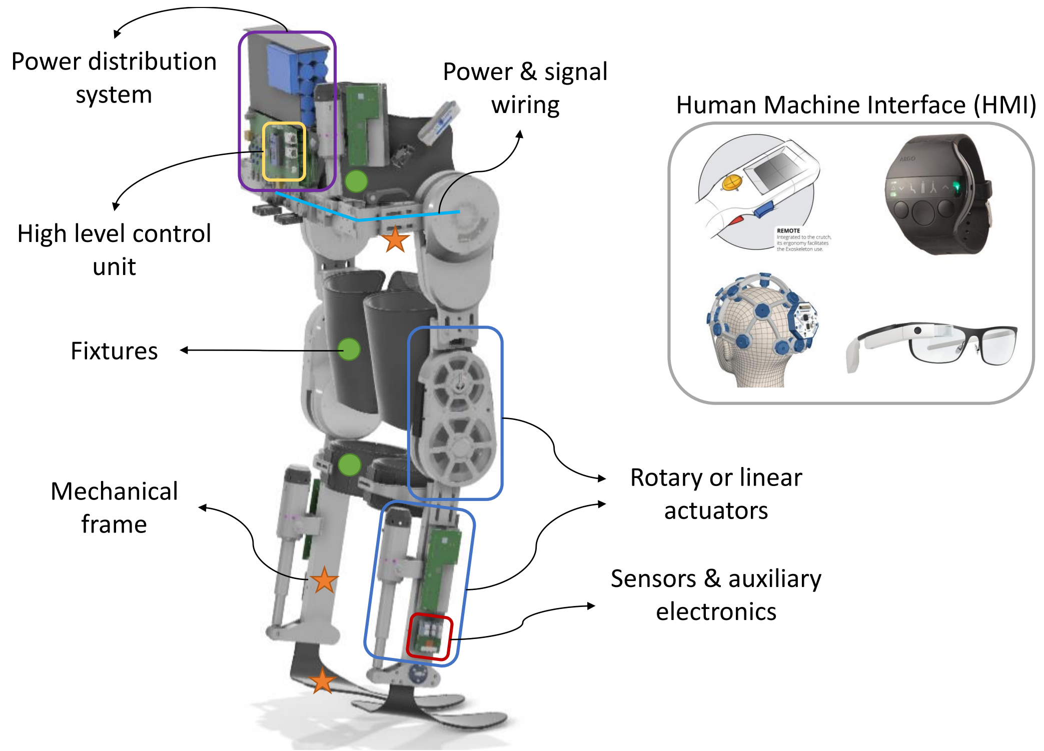


Figure 1.6: Block diagram of the components of a active lower limb exoskeleton, modified from [22] with permission of Project MARCH team.

translatory. For example, with a rotary electric motor, it can be achieved a rotary or translatory final motion depending on the type of gearbox chosen. To simplify the discussion, sensors and other circuits that are intended to support low-level control of the actuator are also incorporated within this definition. Examples are position sensors placed on the motor shaft (or inside the motor) and/or after the mechanical transmission.

Mechanical frame the mechanical frame has the function of supporting both the human's and the robot's body. An exoskeleton can weigh from 10 kg to 60 kg and more, and this weight is mostly accounted to the actuators, mechanical frame and back pack. In order to save weight and thus increase the performance of the system, one of the variables that can be manipulated upon the most is the mechanical frame. At the same time, however, the design of the mechanical frame of the robot is one of the most important challenges in the design of an exoskeleton. To simplify the calculation of the above problem, FEM and worst-case analyses are used, often with safety factors two or three. The materials used are generally metal (especially aluminum) and carbon fiber.

Fixtures they allow the frame of the exoskeleton to be firmly anchored to the body of the subject wearing it; the design must be careful with the aim of achieving the greatest possible comfort (following the anatomy of the subject's body as closely as possible) while at the same time having excellent structural strength. To achieve these characteristics, they are usually made of composite materials, for example, with an aluminum, plastic, and/or carbon fiber structure and foam rubber for the trim part. Other parameters to take into account in the design are speed and ease of application/removal.

High-level control unit generally, lower limb exoskeletons have two types of controls: one at a low-level (the motor driver) and another at a high level (usually an embedded PC). Specifically, the high-level control unit is responsible for making complex calculations of the overall kinematics (and often dynamics) of the robot, based on various feedbacks coming from the devices that make up the EXO; then these calculations are translated into torque and/or velocity and/or position references, which are distributed to the individual low-level controls.

Power distribution system in this category are found the elements that are used for the delivery, distribution and control of the energy required to move the actuators. In the case of electromechanical actuators, the battery will then be found as the main component. In these applications, due to strict requirements such as high capacity and low weight, the predominant technology is lithium-ion. Given the unstable nature of these batteries, there are often controlled distribution and monitoring boards that enhance the safety of the exoskeleton and aid troubleshooting in cases of anomalies.

Power & signal wiring although wires have no special technological peculiarities, the careful design of them is necessary for the proper operation of the entire EXO. They can be divided into two types of wires: signal wires and power wires. The first ones deal with connecting sensors or transmission/reception modules arranged in the various elements that make up the robot and thus handle signals with information content; the second ones, on the other hand, carry the power coming from the power distribution system to the various utilities. To decrease the number of wires (especially signal wires), what are called Fieldbuses [23, 24] are arranged, which allow with a reduced number of wires (usually two or four) to carry all the information to the various nodes of the system.

Sensors & auxiliary electronics fall into this category, all sensors and electronic parts that play a secondary or optional role in the operation of the exoskeleton. An example are inertial measurement units (IMUs) [25, 26] and pressure sole sensors [27], signal conditioning circuits, filters and so forth.

Human machine interface (HMI) it represents the bridge that connects the human with the exoskeleton; they can be an integral part of the exoskeleton or a separate extension of it. An example of an HMI integrated into the exoskeleton may be a beeper or diagnostic LEDs placed behind on the back pack; while a crutch with a hand control, a smart watch, or a BCI may be an HMI external to the exoskeleton.

1.2.2 Actuators

It's possible to define an actuator as an element that is capable of performing mechanical movements (with a certain torque or force) by transforming the energy supplied to it, which is generally carried by electrical, pneumatic or hydraulic means. Actuators are present in most of the machines that surround us in our daily lives. To make an analogue to the human body, the actuator performs a function similar to the muscle, which converts biochemical energy into mechanical movement.

The actuator (but also the muscle) is actuated by a control signal, which is controlled by a central unit. Often by the term servomotor, we refer to a particular actuator that assembles within it such a central unit and other components supporting it, for example, position or current sensors. In the following discussion, as mentioned in the previous section in the description of a lower limb exoskeleton, the generic term actuator will be used to include within it the motor, the gearbox, the driver and its auxiliary sensors.

Let us therefore briefly describe the main components that make up the actuator, which will be the element to which I have given the most space in the design choices of this thesis work. This description is intended to prepare the reader for a subsequent chapter on the selection and analysis of products on the market, and therefore, is not intended as an in-depth explanation of these components.

Brushed and brushless DC motors

One of the most important parts that constitute an actuator is definitely the motor. This element, beyond its type, is composed of the parts seen in the figure 1.7 which are mainly two: a stationary part called stator and a moving part called rotor [28]. The air gap between the stator and the rotor is necessary for the motor to turn, and the length of the air gap can vary depending on the type of motor. The stator and rotor have an electrical circuit and a magnetic circuit. These components are constructed with a ferromagnetic core, as shown in figure 1.7, through which the magnetic flux created by winding currents flows. These windings are arranged inside the slots, which play the role of supporting the winding conductors.

The conductors, inserted into the slots of the iron core, form the electrical circuit. When current flows through these conductors, a magnetic field is created across the iron core and the rotor and stator assembly composes an electromagnet. This fundamental principle underlies each electric motor and the constructive implementation

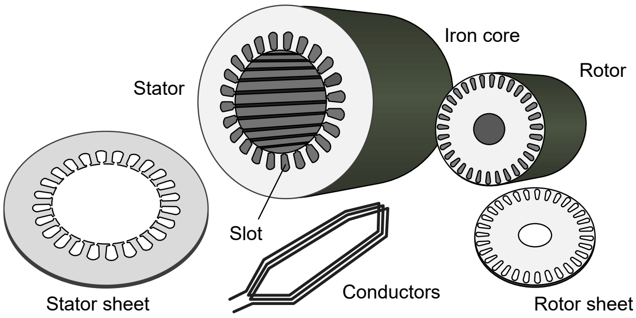


Figure 1.7: A rotary electric motor [28].

of it, determines the type of motor and in cascade, the control circuit and associated techniques.

To achieve higher magnetic flux for a given current in the conductors, the iron core is usually made of ferromagnetic material with high permeability, such as silicon steel [28]. In some cases, the stator or rotor creates magnetic flux using a permanent magnet.

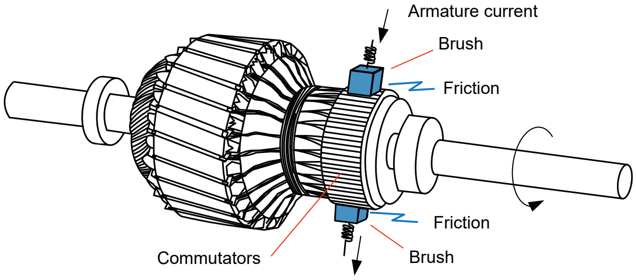
Torque production (and as an effect, speed) occurs by means of the interaction between these two magnetic fields (stator and rotor). The control techniques of such motors, have among their various purposes, the main one of keeping the torque as constant and unchanging as possible, being of fundamental importance to keep the motor moving.

Historically, a type of motor widely used in the past but still in use today is the brushed DC motor (BDC). In this type of motor, the power supply is continuous, and therefore the magnetic fields generated on the stator and rotor, are stationary. To produce a useful torque to turn the motor, we would therefore need these two magnetic fields to be at 90° to each other. However, since the rotor will be in relative motion with respect to the stator, it would therefore not be possible to maintain this ideal condition.

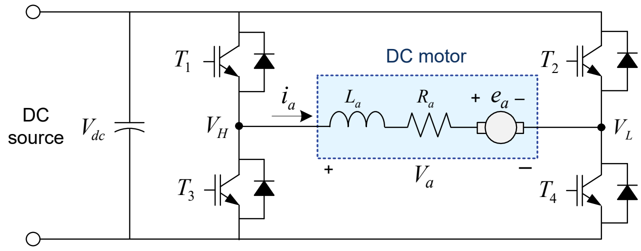
In order to actually produce a stationary rotoric magnetic field, a special electromechanical construction is implemented in this type of motor, consisting of brushes, conductors and a commutation ring, as depicted in figure 1.8a.

It has the peculiarity of greatly simplifying the control of such motors, in which the torque is proportional to the current by means of a constant, which is a design feature of the motor. The control circuit is also simplified as a result, as can be appreciated in figure 1.8b in which there is a simple H-bridge in which we can control the direction of rotation depending on whether T_1 and T_4 or T_2 and T_3 are activated.

Such a construction, however, also leads to non-negligible disadvantages, such as friction between the brushes and the commutation ring and short circuits in the



(a) Rotor assembly of a brushed DC motor (BDC)



(b) Typical control scheme for this motor (H-bridge)

Figure 1.8: Brushed DC motor (BDC) [28].

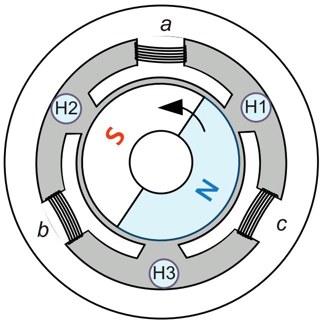
transition from one winding to the other under dynamic conditions (especially troublesome during startup). These non-idealities greatly decrease the total efficiency of the motor. In addition, as a result of this construction, the power-to-size ratio is modest.

In recent years, due to the development of electronics, the BDC has begun to be less and less present in certain applications, at the expense of the BLDC [29, 28].

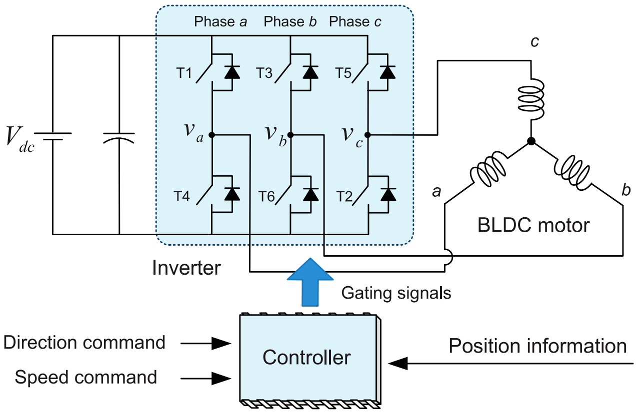
In figure 1.9a we can observe the construction of such a motor, which has three coils, necessary for the construction of the stator magnetic field and the permanent magnets to create the rotor magnetic field. We immediately notice the simplification in construction that allows it to have a high power to size ratio, as well as eliminating the friction and non-idealities described for the brush motor [29].

In this motor, since there is no electromechanical circuit as in the brushed motor, in order to achieve sustained torque, it is necessary to implement externally a more complex control, in which commutation is done electronically. We also note that in the electromechanical construction of such motors, it is necessary to integrate (usually) three hall-effect sensors, which are needed to estimate the position of the rotor and subsequently correctly recreate a stator magnetic field at 90° to the position of the rotor magnetic field [28].

Figure 1.9b shows the block diagram of the typical control circuit for such motors. We can immediately see the greater complexity compared to the H-bridge seen in figure 1.8b. In this case we have six MOSFETs (or IGBTs depending on the power involved) instead of four and a microcontroller unit (MCU) that reads the rotor



(a) Section view of a brushless DC motor (BLDC) motor



(b) Typical control scheme for this motor (inverter + controller)

Figure 1.9: Brushless DC motor (BLDC) [28].

position by means of the hall sensors placed inside the motor and consequently, controls the MOSFETs, by means of specific modulations.

Without going into detail about the various existing modulations, it is useful to know that one of the best performing is the field oriented control (FOC), at the expense, however, of the computational load at the side of the MCU. Note that in this case (and in other modulation techniques) the current is sinusoidal. Therefore, sometimes the term BLDC can be misleading because the motor is not supplied by a DC voltage (or current), but thanks to the inverter, it is possible to reproduce a sinusoidal pattern from a DC supply voltage input rail [28].

Mechanical transmissions

Actuators can be divided into two categories: reduced actuators and direct actuators [30]. By reduced actuator, we mean an actuator that internally, has a gearbox cascaded to the motor, while by direct actuator, we mean an actuator that has the output shaft directly coupled to the motor, without any reduction.

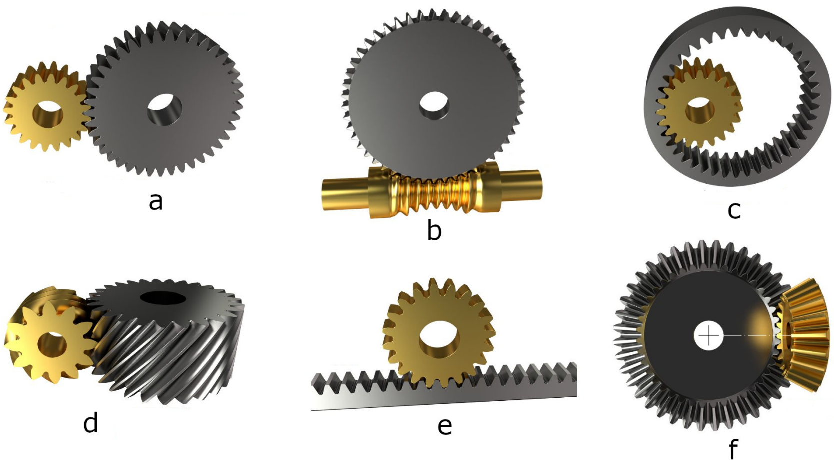


Figure 1.10: Some types of commonly used gears: ⓐ spur gears (external toothing) ⓑ worm gear ⓒ internal toothing ⓓ screw gears (hyperboloid gears) ⓔ rack and pinion ⓕ bevel gears [33].

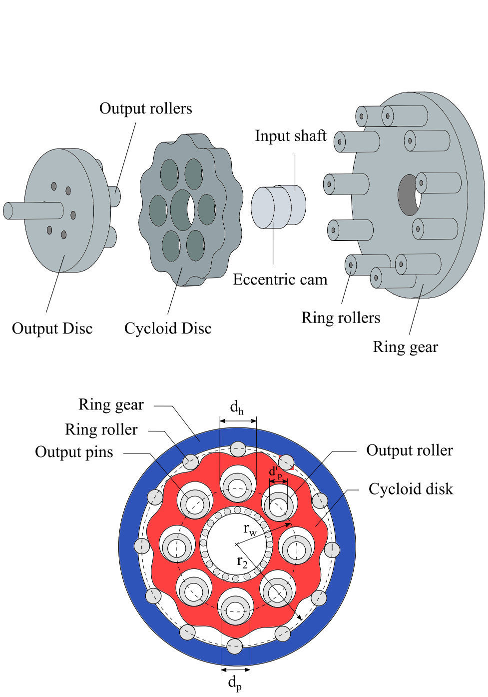
The purpose of a reduction is to increase/decrease the torque/speed of the chosen motor, specifically for the application in which the actuator is used. There are various techniques for reducing motion, which can be divided by type, for example, rigid transmissions (gears, worm-screw, etc.) and flexible transmissions (chains, belts, etc.) [31].

In the application field of exoskeletons, mainly rigid, and usually gear-type transmissions are used, therefore in figure 1.10 the main types of gear-type transmissions are shown.

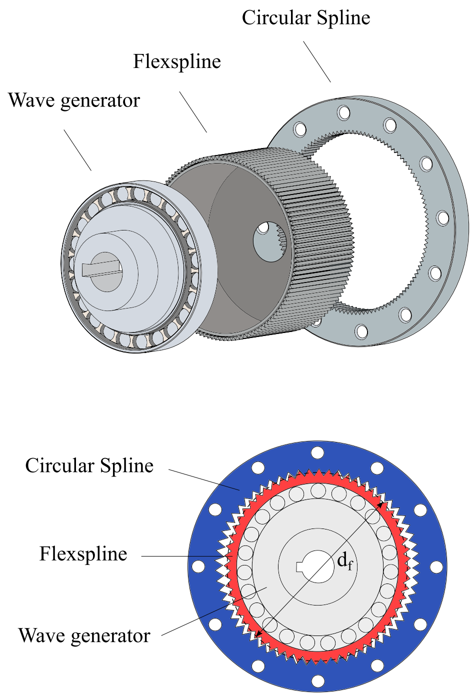
Going into more detail, they can be divided into two other subcategories, namely external gears (detail ⓐ of 1.10) and internal gears (detail ⓒ of 1.10). The latter category is among the most widely used in this field because it allows for high reduction ratios while maintaining a small footprint [32].

Among the implementations of internal gear transmissions, we find mainly three specific types in the market, namely: planetary drives (figure 1.11c), harmonic drives (figure 1.11b) and cycloidal drives (figure 1.11a). These three types are widely used in robotic applications [34].

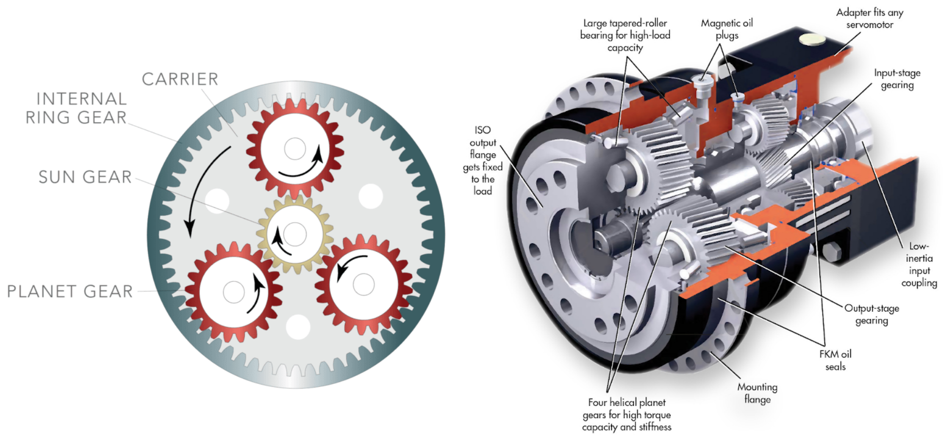
The parameters that distinguish each implementation and determine its design choice are: footprint, performance, life and wear, mechanical backlash, inertia, and many others [35, 36, 34, 37].



(a) Cycloidal drive construction



(b) Harmonic drive construction



(c) Planetary drive construction

Figure 1.11: Common internal gearboxes used in WR applications [35].

1.3 Aim of this thesis

This thesis aims to analyze an existing open source exoskeleton, which was created for pediatric use and later adapted for adults. The analysis is intended to outline the characteristics and limitations of this exoskeleton and propose improvements for a future revision or for the design of a new model.

The analysis of the torques required at the joints, based on the scientific literature of healthy adult persons, will lead to questioning the actuators present on the exoskeleton and in cascade all the other components of the system.

This analysis will then be followed by a major components research and technical evaluation of these parts. Next, a concept design of the joint revision that I aim to make will be presented. As a first stage of development, I will then go on to implement the low-level control of the mechatronic joint. Furthermore, in order to evaluate the correct design choice of the components, a test bench will be setup in which various tests will be performed, including a simulation of the knee gait profile through data collected from the literature on healthy subjects.

1.3.1 Outlines

Chapter 2 the ALICE exoskeleton is presented; the main components are viewed from a mechatronic point of view, using the classifications and knowledge base introduced in the first introductory chapter. Improvement analysis conclude the chapter, which lays the foundation for the next chapter and the entire thesis work.

Chapter 3 commercially available solutions are evaluated; following the selection of these components, a concept design of the single joint is presented. Finally, other design solutions that may be considered in a future advancement of the project are set forward.

Chapter 4 the development of low-level control is exposed; a classical approach leads to step response analysis for the three control loops: torque, velocity, and position.

Chapter 5 high-level control is simulated using a MATLAB script. For this purpose, a dataset of a knee gait profile is used, following the concepts seen in the first introductory chapter. An experimental setup is prepared in order to overload the actuator and simulate a condition similar to a real usage.

Chapter 6 here there is the general discussion of the entire thesis project and then, the conclusions.

2 ALICE: an open source exoskeleton

> *In open source, we feel strongly that to really do something well, you have to get a lot of people involved.*
>
> — **Linus Torvalds**
> Engineer

In this chapter we will look in detail at the characteristics of the exoskeleton held at the Intelligent Autonomous System Laboratory (IAS-Lab) at the University of Padova shown in figure 2.1a, while in figure 2.1b all the components ready for assembly. ALICE, developed by the French company INDI Engineer and Technology [38], is an EXO developed primarily with open source parts and materials produced by 3D printing, which aims to bring down costs (the total cost of all components is about $1500) to make the technology accessible to as many people as possible. Initially developed in Mexico for pediatric use, it has now evolved to be used on adults and mainly in scientific research.

2.1 Mechanical description

2.1.1 Fixtures

The fixtures of this exoskeleton are ordinary plastic snap-on buckles with adjustable cloth tapes, which pass through the plastic elements (ABS plus carbon fiber). The latter, are attached to the frame of the EXO by through-through M6 screws, also securing the composite bulkheads that are in contact with the user's body. These bulkheads follow the anatomy of the point where they are anchored and are made of PLA with two layers of carbon fiber, completing with a layer of soft material; in figure 2.2 the fixings just described are visible. Similar discussion regarding the fixing for the torso, visible in figure 2.3b and 2.1a.

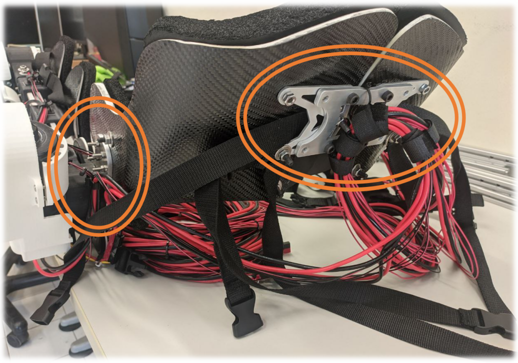
(a) Final assembly of the exoskeleton (without control box) in sit position on the table

(b) Workbench with all components ready for exoskeleton assembly

Figure 2.1: The Alice exoskeleton held by the Intelligent Autonomous Systems Laboratory (IAS-Lab) at the University of Padua

Figure 2.2: Fixtures of a leg of ALICE exoskeleton.

(a) Aluminium profiles of different lengths



(b) Back support and anchor brackets to the exoskeleton frame

Figure 2.3: Key frame parts of the ALICE exoskeleton.

2.1.2 Frame

The mechanical frame is composed mainly of fixed-length links made of commercially available standard 20x20 mm aluminum profiles. There are two models of profiles and they can be distinguished by the presence or absence of the hollow wire pass and by the drill holes on them; one for the femoral section and the other for the tibial section. With this solution, since there is no adjusting system, there is a simplification on the design side (thus structural and assembly) but at the same time a variety of aluminum profiles have to be produced to be able to cover most of the anatomical lengths of the various subjects and also there has to be a frame links replacement phase.

Then the two halves (the frames for each leg) are anchored to the back seat which is also divided into two halves, fixed by a plate - counterplate system which has two possible configurations to be able to adjust the width of the seat for the user's hip. Such plates are circled in figure 2.1a and are made of aluminium with a thickness of 5 mm and 2 mm with M4 and M6 screws.

2.1.3 Actuators

The actuator used for the four joints is an AM equipments 226 and consists of a BDC motor (typology seen in the figure 1.8) and a worm gearbox (detailed b of figure 1.10). It has a starting torque (i.e. at zero speed) of 26 N m and a nominal torque of about 10 N m; the rated voltage is 12 V and the power consumption is 120 W nominal and 260 W peak. Figures 2.4a and 2.4b show the front and back views of the motor, respectively.

The mechanical coupling is made with play on an aluminum plate, with no axial constraint on the motor shaft. In figure 2.4c the hip joint assembly is shown. It is worth noting that the two black side parts lock in place the bracket to which the aluminium profile is anchored. In addition, these parts have the structural function of mechanical stop in case of system failure (e.g. position sensor breakage or control problems).

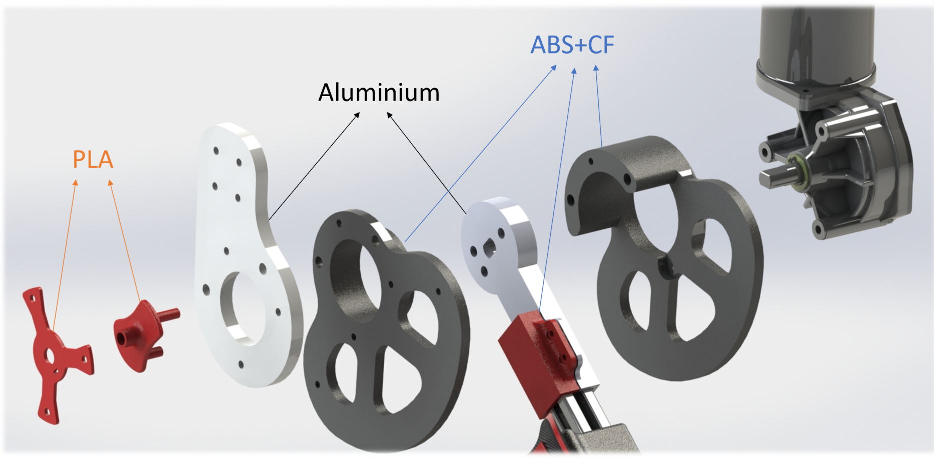
2.2 Electrical and electronic description

2.2.1 Power and cable managements

The electrical system consists mainly of 12 AWG (eight wires in total) silicon wires for the phases of the motors and 26 AWG (twelve wires in total) for connection to the position sensors of each joint. Since there is centralized control, this results in

(a) Front view of the actuator

(b) Back view of the actuator



(c) Exploded view of the motor assembly with the frame; different materials are used depending on the structural role of the component

Figure 2.4: Electromechanical actuator used in the four joints; hip joint assembly is shown.

(a) $LiFePO_4$ 12V battery

(b) Bundle of wires with related connectors to be terminated on the control box

Figure 2.5: Battery and wiring setup of the ALICE exoskeleton.

a bundle of 20 wires flowing from the EXO to the external control box; the length of this backbone is 10 m and it is possible to interchange it with one of a different length (for convenience of the experimental setup).

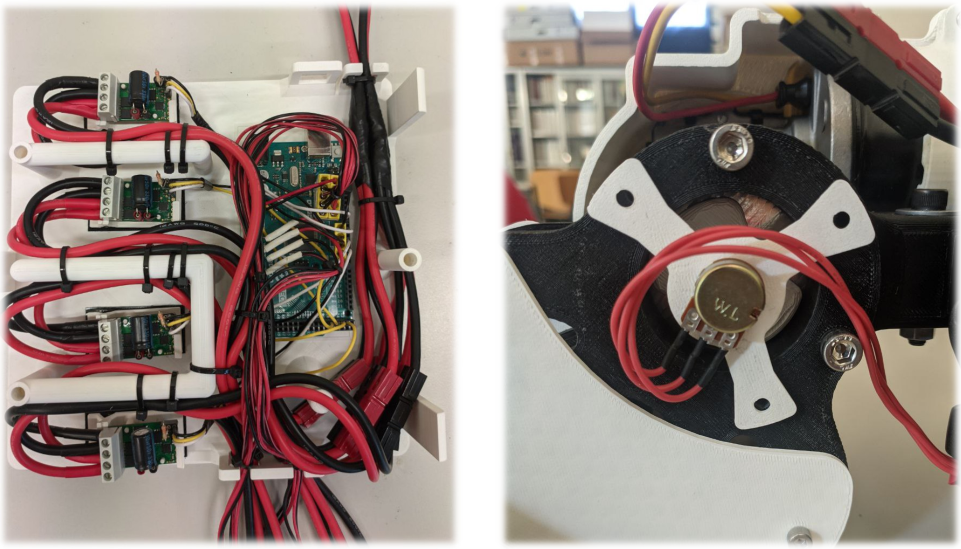
There are two types of connectors; the first, for the power section, are Anderson PowerPole 15-45 connectors that are very common in the automotive industry; the second, for the signal section, are SMP-03V-BC from JST. In figure 2.5b the wire bundle is visible with related connectors.

Regarding the battery, a 12 V $LiFePO_4$ battery was chosen. This type of battery is less performing in capacity-to-weight ratio than Li-NMCs, however, due to their characteristics (lower cost production, high safety, low toxicity and long life cycle) they are increasingly being used in the automotive field. In figure 2.5a it is possible to see such a battery with its cable cord for connection to the system.

Lastly, there are no other control or safety circuits for system power, so the battery is connected directly to the system.

2.2.2 Control box and sensors

The central control unit shown opened in figure 2.6a has four H-bridge drivers for brushed motors, the Pololu 18v25, which can work with a voltage between 6.5 and 30 V and a continuous current of 25 A; it also has additional features such as reverse polarity protection and the ability to estimate the current and be able to limit it. Inside there is also the control board which is an Arduino Mega 2650 R3 which has an 8-bit MCU, 256 kbit flash memory and a 16 MHz working frequency.



(a) Internal wiring of the control box

(b) An ordinary potentiometer acting as an angular position sensor

Figure 2.6: Inside view of the centralized control and coupling of the angular position sensor.

In figure 2.6b is possible to see the potentiometer used for the implementation of the position sensor. The coupling with the link flange and to the motor shaft, is made with three pivots that prevent a improper mounting and ensure a correct zero position.

2.3 Control description

The robot is controlled by means of a graphical interface, which communicates through the serial port of the Arduino Mega. This interface triggers the states of a state machine inside the central control unit.

2.3.1 Gait profile

The control of ALICE has a very basic implementation. Each joint is controlled by means of a PID whose purpose is to reach the desired target position. In figure 2.7 we can see the profile of the angular displacement that ALICE tries to reproduce, which is a trapezoidal composed of two angular position references (45°) with different duration and time interval between them.

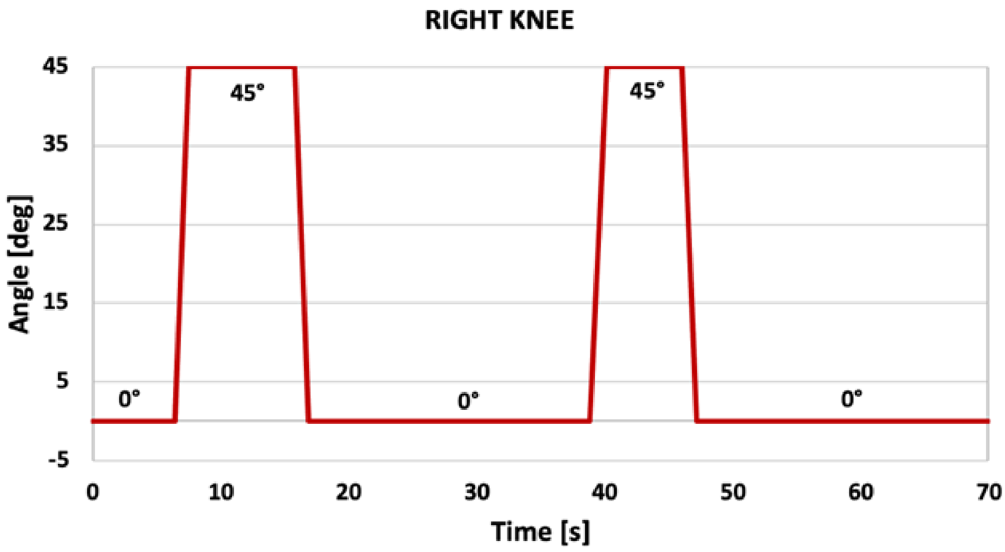


Figure 2.7: Right knee angle profile generated by ALICE exoskeleton.

Table 2.1: Maximum absolute joint torques for a healty 80 kg, 180 cm person. [39]

Task	Hip max. (N m)	Knee max. (N m)	Ankle max. (N m)
level walking	65	40	125
stair ascent	40	90	105
stair descent	50	105	90
sit-to-stand	50	70	45

2.4 Improvements analysis

Having concluded an initial descriptive analysis of the exoskeleton at our disposal, let will now, in this section, go on to enrich its analysis with an improvement point of view. As mentioned above, this EXO was created for pedriatic use and only later, adapted for use on adults. This adaptation is the cause of the main critical issues that emerge in the following analysis.

The first point to be analyzed, therefore, is the torques involved in the new field of use. In table 2.1 the approximate order of magnitude, for specific task, of the torques required for each joint can be consulted. These measurements were made by direct trials on a healthy subject of 80 kg and 180 cm [39].

It is important to specify that these torques are maximum and not average, but for the purpose of a design analysis, a choice based on maximum torques may be a conservative and precautionary decision.

Referring to ALICE, therefore, we have a torque of 26 N m (however, at zero speed) and about 10 N m at nominal speed. Comparing then with the table just mentioned, the main criticality emerges in the case of, for example, a hip joint, level walking, where a torque of 65 $\mathrm{N\,m}$ is needed at least impulsive, but at a sustained nominal speed. It follows, to a practical approximation, that an actuator with at least four times the nominal torque would be needed.

It is clear, therefore, that the electromechanical actuator on ALICE is not sufficient to carry the full load of the human-plus-robot system, and the help it provides to the movement of the adult subject is poor. As seen in 1.2, the mechanical actuator is the central component of the exoskeleton, and the choice of it, affects in cascade all the other components of the system.

In the specific case of ALICE, in order to achieve such a jump in performance, it would be necessary to question the use of BDC motors in favor of BLDC motors (following the considerations made in the section 1.2.2); a consequent evaluation of the driver to be used is equally important. And again, it may be necessary to raise the working voltage by a factor of two or even four, to mitigate the currents involved and keep the wire sections to be used contained.

With such a performance jump, not even the mechanical structure can be exempt from structural evaluation; therefore, it may be necessary to consider increasing the cross-sectional area of the aluminum sections and improving the coupling between the shaft and the joint, and also to ensure that the mechanical stops (figure 2.4c), brackets and anchor screws (figure 2.3b details) can withstand such a power increase.

Further negative note of ALICE is that the mechanical plays of the structure, lead to a non-negligible mechanical hysteresis on the movement of the joints; with the motors off, a member can be moved about ± 5 degrees, which added together for hip and knee, lead to a non-negligible overall foot position error.

Even more dysfunctional is the gait profile implemented in the main control. In fact, this profile is not even approximately similar to an anatomical profile of a healthy subject and therefore, this aspect will need to be revised.

In the second instance, other aspects follow that are functional in ALICE but can be improved. For example, the decentralized structure of the control box, allows easy access to modifications or repairs of the electronics, while also decreasing the weight on board the exoskeleton; however, it results in the use of a wiring harness with a not negligible weight and cost, as well as presenting a not insignificant impediment in an experimental setup.

The potentiometers used are certainly inexpensive and easy to replace and use, but by their nature, they are prone to noisy signals and frequent failures, especially if axial or bending forces are applied to their wiper arm. In addition, there's no precaution implemented to handle the case where a potentiometer is disconnected or breaks; it might be worth improving this aspect by finding an improved solution.

To conclude, as seen, the control unit has a MCU that is limited in performance, and could be problematic in case we want to manage and integrate new devices on it, or implement more complex control algorithms.

As can be understood from the statements just made, an imposing concept design and component research effort is expected in order to meet the proposed improvements; this design will also have to maintain (if possible) simplicity of manufacture/assembly and, at the same time, keep costs down, all of which positively distinguish the currently available ALICE exoskeleton.

3 Concept design

> *Strive for perfection in everything you do. Take the best that exists and make it better. When it does not exist, design it.*
>
> — **Henry Royce**
> Engineer

AFTER having analyzed in detail the characteristics of the ALICE exoskeleton and outlined the improvements needed, we move on in this chapter to examine commercially available solutions using as discriminating criteria the technical, economic and logistical characteristics of each component schematized in the summary tables. At the end of this analysis, final design choices will then be made for the design of the new exoskeleton joint in question equipped with relevant sketches and possible ideas for future developments.

3.1 Ready to use solutions

The most immediate and least design-intensive solution might be the use of integrated solutions in order to rule out any compatibility problems between components. However, since the design and production of exoskeletons is still limited primarily to the scientific sphere, no commercial solutions are available, so it was necessary to evaluate solutions adopted in similar application fields. Therefore, the application areas referred to are those of automation and industrial robotics.

Although there are existing solutions that achieve the performance designated in table 2.1, the two main problems that arise, transporting such solutions into the field of WR are two: axial size and cost. We can realize this in the solution in figure 3.1a which could be a potential off-the-shelf solution, but which has an axial size of 170 mm and which therefore, is excessive, given the need to minimize the overall dimension and weight of the exoskeleton.

Keeping on talking about solutions with BLDC motors, we have in the last column of the table 3.1 MyActuator's solution (figure 3.1c) that develops 50 $\mathrm{N\,m}$ of nominal torque and has a dimension that may be reasonable. Note that the data given in table 3.1 and in those that follow are necessary but not sufficient conditions for correct sizing. In fact, there are other metrics that are difficult to quantify, such as documentation provided, facilities in terms of assembly, hardware and software modifications, commercial and logistical support, and so on. The additional conditions just mentioned led me to put the RDM-X10 S2 in second place in my final choice of actuator.

Table 3.1: Ready to use actuators.

Features	**Actuators**			
	Automationware AW-J17	Doga 259	AM equipment 240	MyActuator RMD-X10 S2
Motor type	BLDC	BDC	BDC	BLDC
Stall torque (Nm)	?	130	40	?
Peak torque (Nm)	70	?	?	?
Nominal torque (Nm)	51	20	15 (?)	50
Nominal speed (rpm)	?	22	21 (?)	65
Nominal voltage (V)	24 or 48	24	24 or 48	48
Nominal power (W)	170	144	?	350
Weight (kg)	1.70	5.90	?	1.70
Reduction ratio (-)	100	?	?	35
Price (€)	?	579	280	570
Internal driver and sensors	✓	×	×	✓
Shaft diameter (mm)	No shaft	14 keyed shaft	11 D shape	No shaft
Dimensions (mm) (L x W x H)	170 x 94 x 94	275 x 162 x 99	(?) 200 x 120 x 80	74 x 120 x 120

the ? indicates that the data were not available from the manufacturer's datasheet
the (?) indicates that the data was derived implicitly from the others, given in the datasheet

Another approach, initially pursued, is to search for an electromechanical actuator similar to the one mounted in ALICE (as mentioned, a BDC motor) but with higher torques, in order to replace the existing one with the new model. This analysis is then integrated by the two actuators in the second and third columns of table 3.1 where we can see that the motor in figure 3.1b might be sufficient to cover the required torques, but at a cost, weight, and dimensions that demonstrate the impossibility of pursuing the path taken in the ALICE design. In fact, if we compare the actuators with BDC and BLDC shown in the table 3.1, the difference in terms of weight-to-power or torque-to-power ratio is sharp and in first analysis, even the price does not justify the choice of actuators with BDC motors.

(a) AutomationWare AW-J

(b) Doga 259

(c) MyActuator RMD-X10 S2

Figure 3.1: Selection of ready-to-use actuators in which the gearbox and possibly, the driver and the angular position sensor are integrated.

3.2 Actuator design

Since I did not find an off-the-shelf solution in the previous paragraph, I then went to analyze the individual commercial products available, in order to compose an actuator that would be able to meet the design specifications outlined in 2.4.

3.2.1 Gearbox selection

Let us start by analyzing the commercially available gearboxes that are suitable for the design in question. As discussed in 1.2.2 the design technologies that allow high reduction ratios while maintaining high efficiency are those with internal gears (figure 1.11). Thus, in table 3.2 we have mainly harmonic solutions, and there is an absence of planetary gearboxes, mainly because of their poor performance in terms of torque output, in relation to their axial size. These gearboxes in their classical implementations, indeed suffer from higher friction in comparison with their harmonic counterparts, leading them to be inferior in terms of mechanical efficiency [34] although they are generally less expensive.

Table 3.2: Gearbox selection.

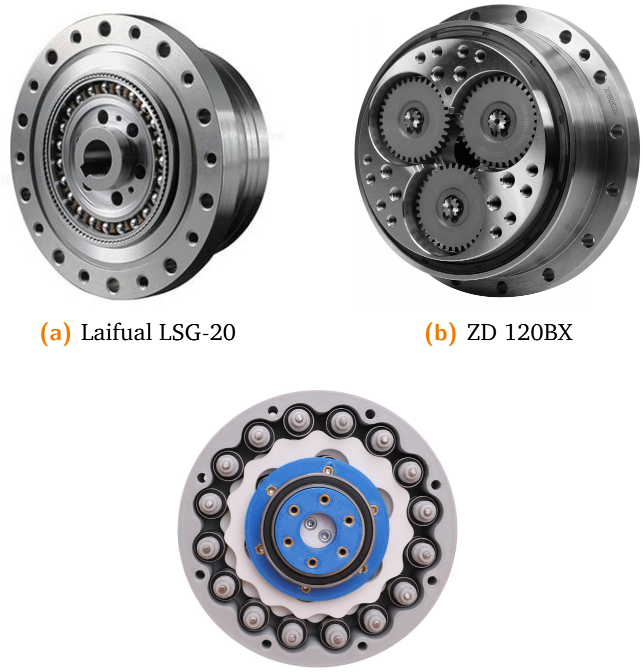
Features	Gearboxes			
	Laifual LSG-20	Hiwin WUT-S-20	HD CSG-2UH	Custom Cycloidal
Peak torque (Nm)	191	147	191	?
Peak torque (Nm) (at start/stop)	107	82	107	?
Nominal torque (Nm)	52	40	52	30
Nominal speed (rpm) (at input)	2000	2000	2000	?
Peak speed (rpm) (at input)	7000	6500	6500	?
Reduction ratio (-)	100	100	100	100
Weight (kg)	0.98	0.98	0.98	?
Price (€)	350	?	600	150
Hole diameter (mm) (at input)	8 modification on request	12 modification on request	12 modification on request	Custom
Dimensions (mm) (radial x axial)	93 x 41	93 x 45.5	93 x 45.5	?

the ? indicates that the data were not available from the manufacturer's datasheet

In table 3.2 it can be seen that in the first three columns there are very similar commercial solutions that are based on harmonic reduction. The selection therefore, is mainly based on price and the logistical and commercial availability that the manufacturer provides. Figure 3.2a shows the Laifual LSG-20 as an example.

Regarding cycloidal solutions, they are less commonly used than their harmonic counterpart because, the price is generally higher and they have greater mechanical backlash [37], as mentioned in 1.2.2. It is therefore difficult to find such solutions commercially and often, the price is on demand. An example of such gearboxes is shown in figure 3.2b, specifically such a gearbox is produced by ZD drive.

Remaining on cycloidal gearboxes, the last column of the table 3.2 shows the performance and price of a potential gearbox composed of 3D printed nylon parts fused deposition modeling (FDM) and recirculating ball bearings, which are in common commercial use. Such a hybrid solution, would make it possible to decrease actuator costs and maintain as much as possible the initial philosophy of ALICE and that is, to use 3D printed replicable parts or at least, parts that can be readily available and replaceable. 3D printed cycloidal gearboxes FDM are widely used in



(a) Laifual LSG-20

(b) ZD 120BX

(c) A 3D printed cycloidal gearbox

Figure 3.2: Gearbox models considered.

robots do it yourself (DIY) and an example of such an implementation is shown in figure 3.2c.

To conclude, I therefore chose the Laifual LSG-20 gearbox shown in the first column of the table 3.2 and shown in the figure 3.2a. The choice is motivated by the fact that most existing exoskeletons use solutions based on harmonic gearboxes [40, 41, 42, 43, 44, 45] from the availability of the Laifual LSG-20 gearbox and its low price. As a sustainable alternative, the design and production of a custom cycloidal gearbox, as discussed just above, is beyond the scope of this thesis and therefore I discarded that option for initial development of the new model of EXO.

3.2.2 Motor selection

In table 3.3 we have a small fraction of the commercial solutions evaluated at this stage (about thirty). The first candidate, the Maxon EC60 (figure 3.3a), is a motor widely used in exoskeleton design [42, 46] along with other models, also from Maxon [43, 47, 48, 39, 49, 40]. It can be assumed that the features for which such a manufacturer has been chosen by so many scientific projects might be for the build quality, reliability (combined with high performance), and rich documentation that the manufacturer provides. The most important technical data of such a motor are given in the first column of table 3.3.

Table 3.3: BLDC motors selection.

Features	Motors			
	Maxon EC60	Nanotec DFA90	T-motor R60	T-motor U8II
Peak torque (Nm)	4.3	1.5	2.3	?
Nominal torque (Nm)	0.536	0.450	0.75	?
Nominal speed (rpm)	3020	2720	4150	?
Nominal voltage (V)	24 or 48	24 or 48	48	24 to 48
Nominal power (W)	220	170	432	?
Weight (kg)	0.36	1.00	0.25	0.30
Price (€)	135	147	175	305
Internal hall sensors	✓	✓	✓	×
Shaft diameter (mm) and characteristics	8 machining on request	10 machining on request	No shaft	No shaft
Dimensions (mm) (radial x axial)	60 x 38	90 x 40	69 x 26	87 x 26
Features	**Motors**			
	Flipsky 6354	Flipsky H5045	Maytech 5055	
Peak torque (Nm)	7	0.83	?	
Nominal torque (Nm)	2	?	?	
Nominal speed (rpm)	3620	3360	4150	
Nominal voltage (V)	24 to 48	24 to 48	48	
Nominal power (W)	850	250	290	
Weight (kg)	0.56	0.36	0.33	
Price (€)	95	55	90	
Internal hall sensors	✓	✓	✓	
Shaft diameter (mm) and characteristics	10 or 8 keyed shaft with two lenght	8 D shape	6 no machining	
Dimensions (mm) (radial x axial)	63 x 54	50 x 45	50 x 55	

the ? indicates that the data were not available from the manufacturer's datasheet

Also in the table 3.3, in the second column, we find the Nanotec DFA90, a similar product to the Maxon EC60. However, in terms of technical characteristics, it is inferior to its direct competitor from Maxon, especially in terms of peak torque output (1.5 $\mathrm{N\,m}$ versus 4.3 $\mathrm{N\,m}$ of the Maxon EC60).

Leaving the industrial field, we find as a product that represents a strongly developing category, the T-motor U8II, a design choice that distinguishes [44] but with a similar pattern also in [50] in which precisely, they mount motors that are designed for high-performance unmanned aerial vehicles (UAV) applied for example in agriculture. Such motors generally do not have internal hall sensors, as rotor position estimation is done by measuring the back electro-motive force (BEMF) on the three phases of the motor.

This choice is driven by the fact that generally such motors are carried up to full speed (the range is from 4000 to 6000 RPM) without having variations in speed and load [51]; moreover, in the initial starting phase of the motor, the load is thrown off to the ground and motion can be easily engaged, then obtaining a BEMF of sufficient magnitude for such estimation [52, 53].

Estimation using BEMF is inefficient in the application on exoskeletons, where the load is constantly applied on the actuator and therefore obtaining a sustainable initial velocity, without knowing the actual rotor position, is difficult [53]. Although there are studies on sensorless solutions based on estimating the BEMF [54], it appears at the moment, still advantageous to add externally a position sensor in order to be able to correctly estimate the rotor position, and thus achieve adequate motor control.

Also, constructively they do not have a motor shaft, but have a flange present on the rotor endbell; this is because in the UAV it is easier to anchor the motor from the bottom and directly connect the impeller on the rotor. This solution is inconvenient, however, in cases where a gearbox is to be coupled to reduce the speed and increase the torque of the actuator. Thus, excluding T-Motor models, all motors in the table 3.3 have a flange on the stator and a shaft connected to the rotor.

Remaining on the same manufacturer, the third table column 3.3 shows the T-motor R60, developed to meet market demands from collaborative robotics and exoskeleton development (application areas that are explicitly stated in the product marketing). Such a motor is shown in the figure 3.3c, which compared to the Maxon EC60, promises higher nominal torque while lowering axial dimension and weight; however, the cost is reasonably higher.

Another economy of scale that is steadily growing and seeing massive use of BLDC motors is that of electric mopeds, scooters, and skateboards. The last three motors in the table 3.3 thus come from these types of applications (particularly the one for skateboards).

Unlike motors from UAV, these have three hall sensors dedicated to rotor position estimation, since, similar to what happens in an exoskeleton, the load is constantly

(a) Maxon EC60

(b) Flipsky 6354

(c) T-motor (Cubemars) R60

Figure 3.3: Selection of motors considered for a possible actuator to be designed.

applied in such applications. In addition, as mentioned earlier, they have a motor shaft, which allows easier coupling with the gearbox.

In this category stands out the Flipsky 6354, shown in the figure 3.3b and with characteristics given in the table 3.3, which has a largely oversized torque for the table requirements 2.1 (if we used the Laifual LSG-20 with reduction ratio 100, we would get 200 $\mathrm{N\,m}$ of nominal torque at the end shaft) and at the same time at a lower cost than its counterparts, but at the expense of increased weight and axial dimensions.

Comparing the technical specifications of the main candidates of figure 3.3, shown in the table 3.3 we have that the T-motor R60 has the right size-to-performance ratio for the application at issue. However, some CAD assembly tests with the selected gearbox revealed the difficulty of creating a coupling with a design that is easy to make and assemble.

Therefore, discarding such a motor, which as we can see from the table 3.3 has a power output of 432 W and thus intermediate when compared to the Maxon EC60 with 220 W and the Flipsky 6354 with 850 W, leaves the two motors just mentioned, which are at the two extremes in terms of power.

It should be noted that for these two motors, which have a similar form factor (see comparison between figure 3.3b and figure 3.3a) the power is directly proportional to the rated torque, making the data provided by Flipsky on request plausible (assuming the data taken from the Maxon EC60 documentation as truthful). The Flipsky 6354 has about four times the rated power and rated torque (see table 3.3) while maintaining less than twice the weight (200 gram difference) and an axial dimensions of 16 mm more than the Maxon EC60.

Finally, considering the motor shaft, we have that the Flipsky 6354 is sold in two sizes of different diameter and length and in addition, it already has a slot for a keyway, necessary for the coupling with the chosen gearbox. The Maxon EC60, on the other hand, has a smooth motor shaft and therefore, machining is done on demand or must be commissioned locally after purchase.

To conclude, keeping in mind all the considerations just made, I have chosen the Flipsky 6354 shown in figure 3.3b with the technical specifications given in the fifth column of table 3.3 as the motor for the actuator that I am attempting to design.

3.2.3 Driver selection

As an initial consideration, the design choices set out in this chapter have not been made in a linear fashion. As the motor, gearbox and driver are mutually dependent elements in the coupling design, several different requirements must be met. Although the choice of motor has already been exposed, in the table 3.4 all drivers evaluated at this stage are listed, which can be combined with the motors seen in the table 3.3 in the previous paragraph.

Looking at the table 3.4, we can see that there are some proprietary and some open source solutions. It therefore follows that the driver proposed by T-Motor can only be used in conjunction with an motor from that company. The same applies to the drivers proposed by Nanotec and Maxon, which, although they do not restrict use with other motors, provide ready-to-use parameterisations for their corresponding products.

It is a different matter, however, for drivers such as the INGENIA XCR-C or SOLO UNO V2, which, although proprietary solutions, have as their target universal compatibility with BLDC motors (and also others, in the case of the SOLO UNO V2). It is therefore worth noting the variability in price of the proposed solutions, which are linked certainly to the hardware used, but also to the proprietary firmware provided.

Table 3.4: Driver selection.

Features	Nanotec N5	Maxon EPOS4	T-motor ALPHA 60A	INGENIA XCR-C
Peak current (A)	40	30	?	20
Nominal current (A)	18	15	60	10
Range voltage (V)	12-48	10-50	24-48	8-60
Nominal power (W)	864	750	?	500
Price (€)	353	571	175	534
Open source	✗	✗	✗	✗
Programming options	1 (*)	1 (*)	1 (*)	1 (*)
Microcontroller	?	?	?	?
Gate driver	?	?	?	?
Dimensions (mm) (L x W x H)	149 x 74 x 44	59 x 65 x 35	75 x 32 x 16	42 x 29 x 19

Features	ST B-G431B	Flipsky FSESC6.7	Odrive V3.6	SOLO UNO V2
Peak current (A)	40	200	120	100
Nominal current (A)	18	70	40	32
Range voltage (V)	12-24	14-60	12-56	8-58
Nominal power (W)	430	3500	2000	1530
Price (€)	18	110	125	175
Open source	✓	✓	✓ | ✗	✗
Programming options	3 (*)	4 (*)	4 (*)	1 (*)
Microcontroller	STM32G431	STM32F407	STM32F405	TMS320F280
Gate driver	L6387E	DRV8301	DRV8301	?
Dimensions (mm) (L x W x H)	41 x 30 x 10	67 x 39 x 19	140 x 50 x 15	87 x 67 x 37

all drivers considered have a 1 $\mathrm{Mbit\,s^{-1}}$ *CAN controller on board*
the ? indicates that the data were not available from the manufacturer's datasheet
the $^{(*)}$ *indicates that there's at least one solution with configuration via GUI*

In this case, the SOLO UNO V2 (figure 3.4d) could be advantageous among the proprietary solutions, as it presents respectable performance and maintains an affordable price.

Moving on to the open source solutions, we note that the performance-to-price ratio is higher than its proprietary counterpart. For example, the ST solution with the B-G431B-ESC1 evaluation board (figure 3.4b) is incredibly affordable (only €18) and at the same time has documentation and various advanced programming environments. However, the driver from ST is designed for applications on small UAV, and has a maximum working voltage of around 24 V thus not allowing the use of batteries with higher voltages (which have the advantage, for the same power, of decreasing the current involved and thus, obtaining smaller cross-sectional power wires).

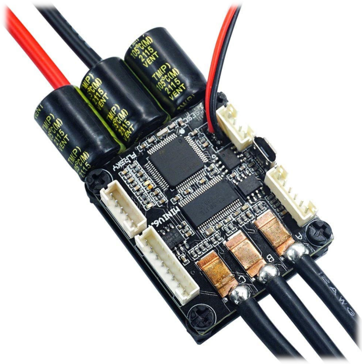
Developing 430 W continuous, it can still be sufficient to supply a motor that can deliver the required torque to the joint. The driver is also very compact, which on the one hand is an advantage, as it can be easily integrated into the design of the coupling; on the other hand, it is a disadvantage in terms of wiring, as it has no connectors and soldering the cables can be difficult. In the end, also driven by its low price, it is an excellent solution, either to start experimenting the control on BLDC motors or to use it in exoskeleton joints that require less torque.

Another open source solution successfully used in Project MARCH [22] is Odrive V3.6 (figure 3.4c). This driver was born with the aim of replacing stepper motors with BLDC in CNC machines and 3D printers; over the years, however, its field of application has also expanded to robotic and control projects in electric vehicles. Until recently, the company's drivers were entirely open source (and thus, in addition to the firmware, also the PCB schematic) however, in order to mitigate the counterfeit copies that had been created over the years, the company decided to have a closed source approach for the future [55].

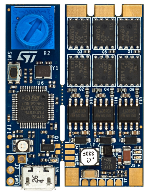
Similar discussions apply to the drivers produced by VESC (Benjamin Vedder), whose focus from the beginning was on electric vehicles (mopeds, scooters and skateboards) but which over the years has also expanded into other sectors, particularly in robotics and large-sized UAVs. As with Odrive, it has an open source firmware that is excellent in terms of performance and functionality; in addition, a graphic interface is provided that allows the firmware to be easily configured and updated, from which all the system's features can also be monitored.

VESC, unlike Odrive, still maintains an open source philosophy, although the problem of non-original copies is present. In the figure 3.4a there is in fact a cheaper copy of a VESC 6, the Flipsky FSESC6.7, with features listed in the sixth column of the table 3.4.

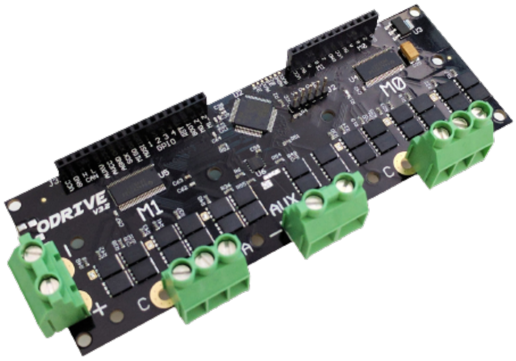
Comparing the Flipsky FSESC6.7 (figure 3.4a) with the Odrive V3.6 (figure 3.4c) using the table 3.4 we note how they mount a similar MCU and the same gate driver, however the Flipsky exhibits better performance, probably due to the MOSFETs chosen and the integrated heatsink. The form factor of the Flipsky FSESC6.7 is also better than the Odrive in that it uses connectors for the low-power signals (instead of a pin header, found on the odrive) while the high-power wires are directly soldered to the PCB (Odrive has screw terminal blocks). In addition, the Odrive v3.6 is designed to control two BLDC motors simultaneously, while the Flipsky FSESC6.7



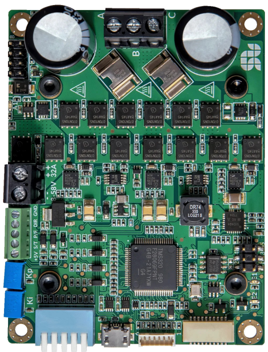
(a) Flipsky FSESC6.7



(b) STM B-G431B-ESC1



(c) Odrive V3.6



(d) Solo UNO V2

Figure 3.4: Final candidates for the driver selection.

only one, making the latter more compact and suitable for single-joint applications, such as the one at hand.

As can be seen from the specifications given in the table 3.4 the Flipsky FSESC6.7 turns out to be the absolute best driver in terms of performance. In addition, as seen from the previous comparison, it maintains a small footprint compared to competitors in its range (Odrive and SOLO UNO V2) and is slightly cheaper than them. To conclude, this driver has four different options for programming and developing the control of the BLDC motor, which will be mentioned in the following sub-section 3.3.3.

Finally, also considering the Flipsky 6354 in figure 3.3b as the motor of choice, I select the Flipsky FSESC6.7 as the driver on which to develop the single-joint control of the exoskeleton.

3.2.4 Position sensor

Considering the electromechanical actuator consists not only of the motor and gearbox, but also of the electronics that enable the actuation of these parts, we can also include in the design of the joint the position sensor placed after the reduction. Usually, in the scientific literature concerning lower limb exoskeletons, the hall sensors inside the motor are employed as incremental encoders, suitable for the proper operation of the innermost loops of the control. At the same time, another angular position sensor, in this case absolute, is employed to close the position loop, which is usually the outermost loop and placed in the high-level control [56, 57].

Theoretically, it would be possible to use a motor without hall sensors and place a single absolute angular position sensor on the motor shaft, from which one can control both the motor and calculate the post-reduction position; in reality, it is a good practice to maintain the rindundance exhibited before, thus increasing actuator reliability (for example) by implementing safety routines based on the difference in measurement of the two sensors. In addition, reading the post-reduction position in some applications allows more accurate control, which can thus take into account the elasticity of the gearbox and other non-linearities, introduced by it [58].

in view of this, let us go on to analyze the main solutions on the market, set out in table 3.5 in which two constructive technologies for position sensors are presented: hall-effect and resistive one.

The technological solution in which potentiometers are used (resistive) has been applied for decades in the automotive field to determine the angular position of component valves in a car engine, thus useful to the electronic controller unit (ECU) for optimal fuel consumption and performance control [59]. Thus, there are numerous products with such technology in that field, but they are hardly available to be purchased by an end-user. In addition, hall-effect solutions are supplanting resistive technology, since they are contact-less and thus have longer life cycles than a potentiometer, in which there are mechanical sliding parts [60].

Applications where it is easier to source various commercial products, as far as potentiometers are concerned, are those of joysticks, which are used in industry to control machines of a variety of types. One of these is the Murata SV01A103AEA0 with characteristics given in table 3.5 and figure 3.5e that can also be found commercially in breakout board form; it is an excellent alternative to the potentiometers used in ALICE, as it has a longer life cycle and linearity of $\pm 2\%$.

Table 3.5: Rotary position sensors.

Features	Piher PST-360	Piher MSC-360	Vishay 157	Vishay 351HE
Linearity (%)	± 1	± 1.5	± 2	± 0.5
Resolution (bit)	12 or 14 $^{(*)}$	12	-	?
Output format (-)	Ratiometric, PWM, SPI, CAN $^{(*)}$	Ratiometric, PWM $^{(*)}$	Ratiometric	Ratiometric, PWM $^{(*)}$
Angular range (°)	360 $^{(*)}$	360 $^{(*)}$	340	360 $^{(*)}$
Multiturn (-)	✓	✓	×	✓
Shaft diameter (mm) and characteristics	14 Through-Shaft	1.9 x 5.9 (rectangular)	3.17	6.35
Rotational life (cycle)	50M	7M	10M	10M
Supply voltage (V)	$5 \pm 10\%$ $^{(*)}$	$5 \pm 10\%$	-	$5 \pm 10\%$
Price (€) $^{(*)}$	62	57	47	41

Features	Murata SV01A103	AMS AS5600	AMS AS5047P
Linearity (%)	± 2	-	-
Resolution (bit)	-	12	12 or 14 $^{(!)}$
Output format (-)	Ratiometric	Ratiometric, PWM, I2C $^{(!)}$	Ratiometric, PWM, SPI $^{(!)}$
Angular range (°)	333.33	360 $^{(!)}$	360 $^{(!)}$
Multiturn (-)	×	✓	✓
Shaft diameter (mm) and characteristics	4 Through-Shaft	-	-
Rotational life (cycle)	2M	-	-
Supply voltage (V)	-	$5 \pm 10\%$ $^{(*)}$	$5 \pm 10\%$ $^{(*)}$
Price (€)	1.65	2.12	9.7

the $^{(*)}$ indicates that it's a hardware customization (by ordering or wiring)
the $^{(!)}$ indicates that it's a software customization

Another resistive solution is the Vishay 157, which unlike the Murata or common potentiometers, the material that makes up the resistor is conductive plastic (it is usually metal). This has a number of advantages over metal ones, namely, longer operating life, temperature-independent resistance, and in general, better linearity.

(a) Phirer PST-360 (b) Phirer MSC-360 (c) Vishay 157 (d) Vishay 351HE

(e) Murata SV01A103 (f) AMS AS5600 (g) AMS AS5047P

Figure 3.5: Final candidates for angular position sensor selection.

Turning instead to hall-effect solutions, there are the two models shown in the table 3.5 by Piher (an Amphenol company), which at a reasonable price, present absolute, factory-calibrated (and thus ready-to-use) angle sensors with analog or digital output (CAN, I2C, SPI, etc.) can be seen in figure 3.5a and 3.5b. Another product from Vishay, the 351HE (figure 3.5d), is also added, which has excellent features at a low price compared to Piher's products. However, being a panel-mount model, it may be difficult to integrate into an actuator design.

Finally, there are the solutions from AMS (an OSRAM company) in the last columns of table 3.5 which are mainly mounted on industrial robots or cobots. The AS5600 sensor in the figure 3.5f is very popular in the market (also in DIY field) and due to economy of scale it can be found at a low price, as can be seen from the table 3.5. Another integrated circuit (IC) from AMS, is the AS5047P which has a resolution of up to 14 bits when using the output serial pheriperal interface (SPI) and has advanced features such as dynamic angle error compensation (figure 3.5g). Both sensors have the ability to be programmed to choose the type of output to be used (PWM or Ratiometric) and at the same time there is always access to the communication interface (I2C for AS5600 and SPI for AS5047P) from which one can take angle readings and have diagnostic messages, for instance on the correct positioning of the magnet.

Summing up, we thus have the candidates in figure 3.5 all valuable products, very similar in terms of performance. We then divide that selection by price range in which we find in the high end the first four candidates (the two piher products and the two Vishay products) while in the low end, the last three (the Murata and the two AMS) with the same order as shown in the table 3.5.

With a view to integrating such sensors into the actuator at the design stage, the two products from Piher are very integrable into the joint design as they have available holes for anchoring, however having a protective housing, it could increase the axial size of the joint.

At the low end, on the other hand, the two AMS products require a more shrewd design in order to integrate them into the assembly, but they make it possible to achieve a more compact joint. In fact, it is necessary to design the housing for the sensor, the magnet, and the proper spacing between the two, as well as taking into account shielding (if there are ferromagnetic parts nearby) and sensor wiring. finally, it is necessary to calibrate the zero point, either by teaching or in the form of an offset at the software level.

the AMS AS5600 has a lower resolution than the AMS AS5047P, but more than sufficient for the application at hand (low demands in terms of accuracy and low post-reduction speed). And it is possible to calculate the resolution in terms of angular position as: $180°/2^{12} = 0.044°$ if one considers changing via inter integrated circuit (I2C) interface its range of operation; with such a sensor it is therefore possible to obtain a resolution of less than a tenth of a degree.

Considering the application field advised on the datasheet of the two AMS sensors and taking into account the performance-to-price ratio, I ended up choosing the AMS AS5600 as the designed actuator post-reduction angular position sensor. Since this sensor has an analog output, it can also be used in the ALICE exoskeleton.

An abstract of the datasheet of the AMS AS5600, shown in figure B.1, is shown in the appendix B, with its block diagram. Also, from the schematic of the chosen driver (figure B.2) it is possible to understand how to connect that sensor on the COMM connector shown on the schematic. Note that in that connector there is an analog input and pins that can be configured as needed, such as SPI or I2C interface. We can then choose whether to use the analog output of the AS5600 or its I2C interface (or possibly SPI if we want to switch to the AMS AS5047P in the future).

3.2.5 Preliminary joint design

In parallel with the selection of the actuator components discussed in the previous sub-sections, I produced several design sketches for each potential combination of motor, gearbox, driver and position sensor. This process helped to understand the advantages and disadvantages of each solution, with the goal of finding the best possible configuration, based on the project requirements. The final design sketches, which include the chosen components, for the construction of the rotary actuator of the newly developed exoskeleton joints are then shown below.

The first draft of the figure 3.6 shows the sectional view of the actuator, where through the labeling given in the related caption, the main components of that assembly can be outlined. As mentioned, the Laifual LSG-20 gearbox was chosen, and therefore, I made use of the excerpts from the manufacturer's manual, given in the appendix A, for the proper mounting with the motor, through a coupling flange.

The design of this flange, complies with what is reported by the manufacturer, shown in figure A.1 in appendix A, which, among other solutions, proposes to use a spacer ring between the shaft shoulder and the face of the wave generator, in conjunction with a washer and a screw placed on the head of the motor shaft; to axially constrain it to the wave generator.

This solution, which involves the use of a spacer ring, is used in cases where there is no possibility of having a custom-made motor shaft, as in this case. It is at this stage, based on the dimensions given in the gearbox manual, that I chose to use the Flipsky 6354 motor with a short shaft (24 mm) and diameter 8 mm instead of the long shaft version (34 mm) or the one with diameter 10 mm.

The result of this design is therefore remarkably simple since only a spacer ring and a coupling flange need to be produced. However, additional threading machining is required on the motor shaft, which already has a pre-drilled hole, probably necessary for the machining of the shaft by the producer.

Going into the details of the motor-gearbox coupling flange, in this first design it has been referred to the guidelines given in the LSG-20 manual, and therefore there is a seat for an o-ring for sealing the grease present inside the gearbox. That flange, has seats on the inside for hex head cap screws for coupling to the motor and as many screws on the outside, for coupling with the flange of the LSG-20. Although the strict geometrical tolerances reported by the manufacturer in figure A.2 in appendix A, the possibility of producing such a coupling flange by means of FDM technology is not excluded, given the low stresses involved.

To conclude the overview, an initial concept design that includes the remaining elements discussed and chosen in this chapter is shown in figure 3.7. This design is not intended to be functional for the production of it, but to assess whether the effective assembly of all chosen components is possible. The axial size obtained is around 120 mm, and the radial size, around 100 mm in diameter.

In the figure detail 3.7a we can see the sub-assembly shown earlier in the figure 3.6 with the addition of the AMS AS5600 position sensor and its mounting bracket, the aluminum profile anchor pieces, the Flipsky FSESC6.7 driver, and the joint protection cover.

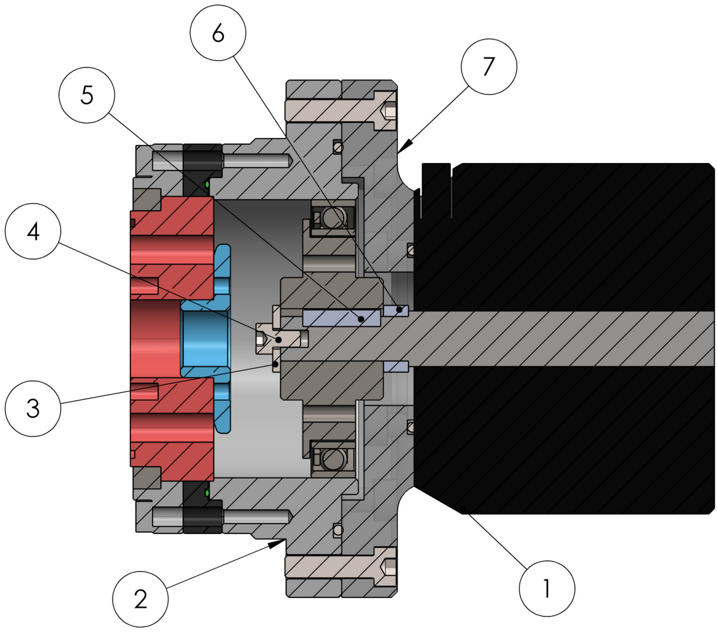
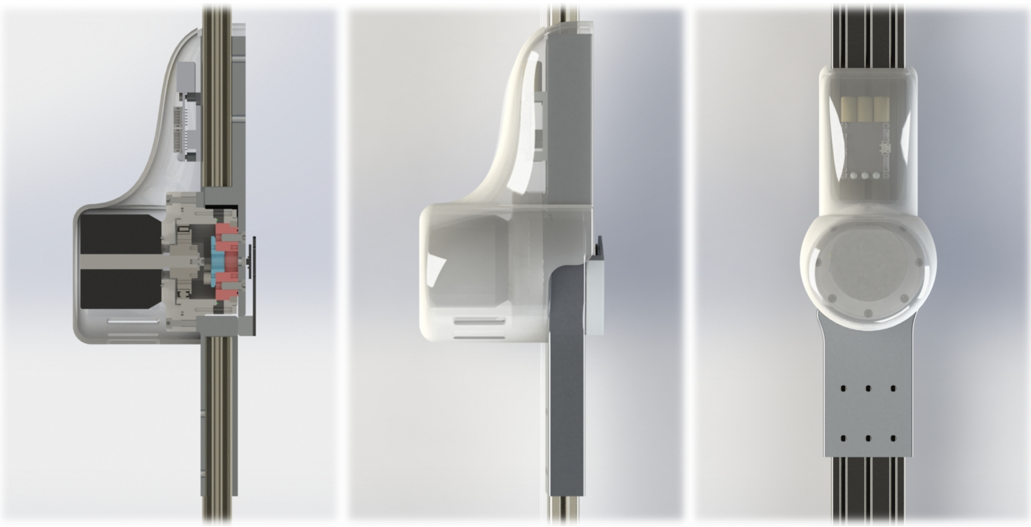


Figure 3.6: Sectional view of the assembly including motor, gearbox and coupling flange with related accessories: (1) Motor (2) Harmonic drive (3) Washer (4) Holding screw (5) Key (5) Spacer ring (5) Coupling flange



(a) Side sectional view (b) Side view (c) Frontal view

Figure 3.7: Concept design in which the developed actuator is applied to a joint (knee) of the new exoskeleton.

Of particular note is the choice to use aluminum profiles of section 60 x 20 mm, which will have to be subject to a FEM analysis, along with the bracket pieces with the gearbox. A possible solution, could be to use profile inserts (which are placed in the slots of them) for fastening and at the same time for adjusting the length of the mechanical links between the exoskeleton joints.

As far as wiring is concerned, it could be considered to place one two-pin connector for power supply and another, also two-pin, connector for CAN bus communication on the wall of the plastic cover, shown in the figure 3.7.

3.3 Other choices

As can be seen from the comments made in the paragraph 2.4, there are still many points to be resolved. In order to achieve our goals, we have maintained a top-down approach, thus going to analyze the entire system as a whole and then going into the detail of the individual, or a small group, of elements.

Before we descend further into the design of the joint, and thus its low-level control, which will be the ultimate goal of this thesis work, let us give space for further design choices that will allow us to close this high-level analysis.

The design choices will therefore be only superficial, in order to leave those who come after, useful insights to work from and further degrees of freedom for design choices.

3.3.1 Power distribution system

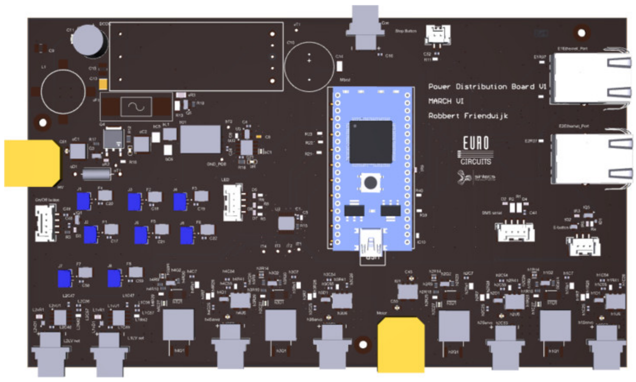
The choice of a more powerful actuator, directly impacts the power management and thus, the power distribution system. Regarding the working voltage, based on the design choices previously made (motor and driver) we can decide to use either a 24 or a 48 V system.

By switching from 12 V (working voltage of ALICE) to 48 V we would have the advantage of remaining contained in the sizing of the cross section of the power cables. After a simple dimensional calculation, it turns out that this jump in voltage (four times as much) allows us to leave the cross section of these cables unchanged and thus, use the same type used in ALICE also for the new version of this exoskeleton.

Regarding the battery pack, in order to sustain such a power increase, it is necessary to use lithium-ion batteries, which among other reasons, thanks to the economies of scale of electric scooters and mopeds, are easily available on the market with voltages of 48 V and up. An example product is shown in the figure 3.8b.

(a) Showcase of the main products of Amass company

(b) 48 V lithium-ion battery



(c) Control board for the power distribution system in one of the iterations of the exoskeleton designed by the Project MARCH team [22]

Figure 3.8: Overview of possible solutions that can be used for the power management of the new version of exoskeleton.

It might also be useful to replace the Aderson PowerPole connectors with some more common and more readily available Amass connectors. Shown in figure 3.8a are some families of connectors that this manufacturer produces. Note that such connectors are also used in the battery of figure 3.8b and in the board of figure 3.8c.

Finally, given the not negligible power involved, it would be necessary to design a board to control and distribute the power to the individual joints, as done by the Project MARCH team [22] where we can distinguish in figure 3.8c the macro functionalities of it:

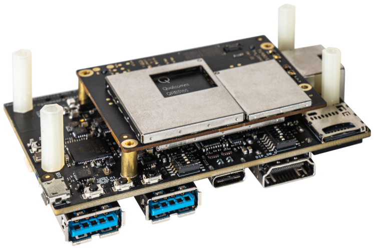
- on the left is the connector to which the battery is connected; just beyond that we have a stage for filtering and converting the voltage into smaller ratings, probably for managing secondary loads;
- at the bottom we have the distribution, through power MOSFETs, to the various loads and thus, to the various joints of the exoskeleton;
- in the center stands the MCU, which is intended to control such MOSFETs and simultaneously, monitor the input power, battery health and actual output consumption of all actuators. Such MCU is of crucial importance to take action and report in a timely manner, shutting down the loads in case of anomalies;
- finally, on the right we can see connectors for communication with other system components, probably with the central control unit.

3.3.2 High level controller (embedded PC)

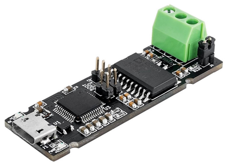
Having chosen the Flipsky FSESC6.7 as a low-level driver, we need to choose an embedded PC that handles high-level computations. The first distinction we can make in this category is by type of architecture: ARM or X86. In the case of ARM we have solutions in which consumption is between 2 and 7 W of power, they are generally more compact solutions, but at the software level there is not always full compatibility with all the applications one would like to use. On the other hand, x86 systems have a significantly higher average consumption, generally ranging from 40 W and up, but they have more computing power, in the same footprint, and more solid support for the software you are intended to use.

For an WR, it is important to consider power consumption, as they are independent, thanks to the battery on board them. In the case of the exoskeleton in question, being developed for a research context, battery life is not a primary aspect that is taken into consideration when making design choices.

(a) UDOO BOLT v8



(b) Qualcomm robotics RB5



(c) MKS CANable PRO

Figure 3.9: Possible hardware components for the realization of high-level control of the exoskeleton.

As an example product for ARM architecture we have in the figure 3.9b the Robotics RB5 development kit from Qualcomm, which is designed, precisely, for robotic applications. Supporting both Android and Linux and compatible with robot operating system (ROS) (version 2), it mounts a system on chip (SoC) with an ARM v8 (Kyro 585) octa-core central processing unit (CPU) running at 2.84 GHz. The SoC has several dedicated elements for specific tasks, supporting machine learning (ML) and computer vision (it can handle up to seven cameras simultaneously). Finally, the development board is compatible with a series of expansion boards, allowing it to cover every need in terms of input/output (I/O).

On the other hand, instead, the figure 3.9a shows an embedded PC with x86 architecture, the UDOO BOLT v8, which features an SoC AMD RYZEN Embedded V1650B, with a quad-core CPU up to 3.6 GHz on turbo boost and AMD Radeon VEGA 8 Graphics graphic processing unit (GPU). the UDOO BOLT V8 also has an 8-bit on-board MCU, an ATmega32U4 (Arduino Leonardo compatible) with a corresponding 40-pin connector that allows it to interface with the embedded PC from the low-level. Such a microcontroller could be useful for implementing a power distribution board as seen in the figure 3.8c by centralizing auxiliary control functions in the high-level control, thus saving interfacing boards. In addition, it is possible to use the general purpose I/O (GPIO) available to the SoC via another 40-pin connector.

CAN bus interface

In order to control the low-level controls of the individual joints, it is necessary to interface with the controller area network (CAN) bus, which is the chosen medium of communication; therefore, we have to find a solution to connect the driver with the high-level control.

There are various options on the market, and the choice depends on the hardware available. In the case where the SoC has an CAN bus controller inside it (like the Robotics RB5), it is simply necessary to add a transreceiver (a kind of logic converter), in order to connect to the bus safely and comply with the logic levels of the standard. A second solution, in case there is no built-in controller, is to communicate via SPI with a board that includes a controller and a transreceiver CAN bus (boards that mount the MCP2515 as a controller are very common in the Arduino ecosystem). Finally, the last alternative, is to use a USB to CAN adapter, in which a MCU having a CAN bus controller is used and also with a USB interface involved, to communicate with a PC.

The latter proposed solution, in the case of embedded PC's, seems to be the best in terms of flexibility. First of all, it is not dependent on the hardware chosen, since there is always a USB port and no hardware changes have to be made, unlike the first two proposed solutions. Second, in order to use the CAN bus controller internal to an SoC or the SPI interface, it is necessary to define it in the devicetree of the operating system kernel, or at any rate, through a low-level software configuration.

To conclude, since the operating system used is generally not real-time, depending on the tasks that the high-level control has to perform, one may run into latency problems, which are critical in the management of the CAN bus [61, 62, 63]. There are latency problems with USB to CAN adapters as well, but in this case, the computational and logistical work for the CAN message packets is in charge of the MCU of such adapter, in which a more deterministic operating system is generally employed; subsequently, the USB communication is handled more efficiently in a modern PC operating system.

With this solution, one is thus less constrained by the performance of the embedded PC, but conversely, one is dependent on the adapter firmware and drivers involved at the embedded PC side. As shown in [62], performance in terms of latency seems to be unfortunately very much related to the chosen hardware and software combination.

In conclusion, figure 3.9c shows the chosen USB to CAN adapter, the MKS CANable PRO, which has an STM32F072C8T6 MCU and an ADM3053BRWZ transreceiver (opto-isolated). The firmware, candlelight, is open source and it is compatible with SocketCAN or python-can.

3.3.3 Software development environment

As a final design choice, possible software development enviroments for the chosen hardware are exposed. Regarding the Flipsky FSESC6.7 driver, we have a very common MCU and that is an STM32F407 which, as mentioned, has several solutions for low-level motion control development:

- the first is to use the Motor Control Blockset on Simulink. This solution allows you to take advantage of the graphical programming that characterizes Simulink and allows you to create advanced simulations with or without the target hardware [64];
- the second is composed of the ST-MC toolchain developed precisely by the manufacturer of the MCU. This solution has an initial configurator that allows to autogenerate a first part of code and set, for example, the parameters of the motor, sensors, and so on, and then leave implementation flexibility in a classic embedded development environment (STM32CubeIDE) in low-level C++ [65];
- the third, is to use the VESC firmware already present on the driver from the factory. Since it is open-source, it is possible to download the sources and make the necessary changes to its own application. In addition, the firmware has a corresponding graphical interface to easily change the parameters of it, also enabling additional features, such as user-customized scripts [66];
- the fourth and final solution consists of development on the Arduino framework, by means of the STM32duino core, using the SimpleFOC library. The latter solution has the advantage of taking advantage of the many libraries available in the Arduino ecosystem and maintaining object-oriented and high-level programming, thanks to the core and the hardware abstraction layer (HAL) used [67].

In the concept design phase, I tested all the solutions just described. Following these preliminary tests, I then chose to use the SimpleFOC library on Arduino framework. This choice was mainly based on development time and personal expertise in embedded programming.

Beyond these aspects, I took into account design needs, such as having to establish CAN bus communication, to integrate an angular position sensor into the control, to develop safety routines, and so on.

In any case, the choice to use an open-source driver with an MCU that would provide multiple implementational ways is a desired plus in order to give more design freedom in any future developments.

Regarding high-level control, an initial evaluation phase showed that the use of ROS Toolbox on Matlab/Simulink could allow easy development and simulation of high-level control for the exoskeleton, decoupling from the chosen hardware. Indeed, the Middleware ROS would allow interfacing with all nodes of the system in a flexible manner, abstracting from the specific hardware and software libraries chosen (e.g., CAN bus communication).

That said, in the next chapter I will go on to expose the development of the low-level control, leaving the high-level control, along with the other design choices set out in this section, for future development of the project.

4 Control development

> *If everything seems under control, you're just not going fast enough.*
>
> — **Mario Andretti**
> Racing driver

HAVING thus concluded the analysis and design choice part of this thesis work, we now go on to address an initial exploratory implementation of the low-level control of the chosen BLDC motor, using the Flipsky FESC6.7 as the driver and the Flipsky 6354 as the motor.

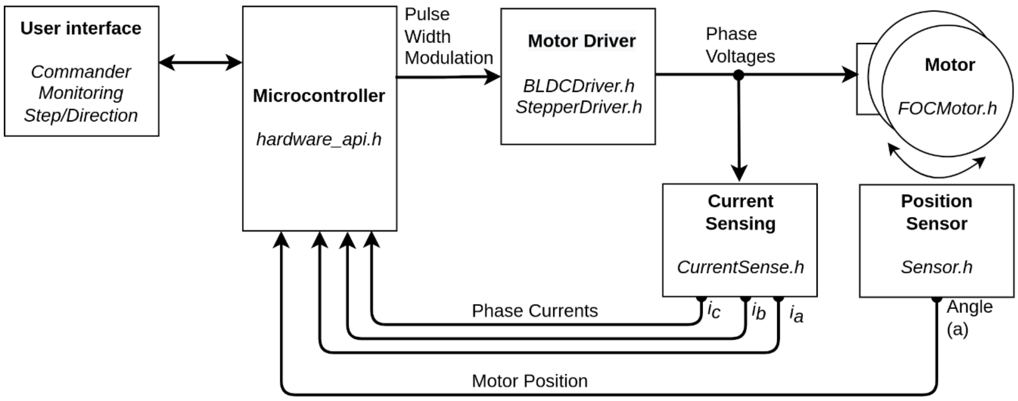
4.1 SimpleFOC library

The SimpleFOC library, thanks to the high-level C++ object-oriented abstraction and an ontological distinction of the code of most of the library's features, allows the user to adapt with few modifications, the provided code, according to his hardware needs. Indeed, we can see the modular structure of the library, shown in figure 4.1a where for each block, the main classes that determine its operation are given.

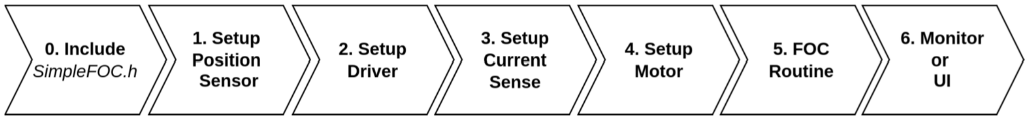
SimpleFOC supports various types of microcontrollers and platforms, such as Atmega328 or 2560, STM32 family, SAMD family, and ESP32, just to name a few. Equally varied is the choice of sensors, which inherit the `sensor` class and are specified in `HallSensor`, `MagneticSensor`, `Encoder` and `GenericSensor`.

The `CurrentSense` class, on the other hand, is used in case it is chosen to implement a current control, rather than a voltage one. With the current control, we can take full advantage of the FOC algorithm, this class is supported by other secondaries, depending on the type of current sensing used (Low-side, High-side or In-line).

On the other hand, regarding the motor and driver, there are two macro distinctions regarding the type of motor used (currently the library supports two types of motors) namely BLDC motors and stepper motors. In any case, both types use and inherit the same class `FOCMotor`.



(a) SimpleFOC modular architecture overview



(b) Steps to follow to setup the SimpleFOC library

Figure 4.1: Block diagrams describing the structure and setup procedure of the SimpleFOC library [68].

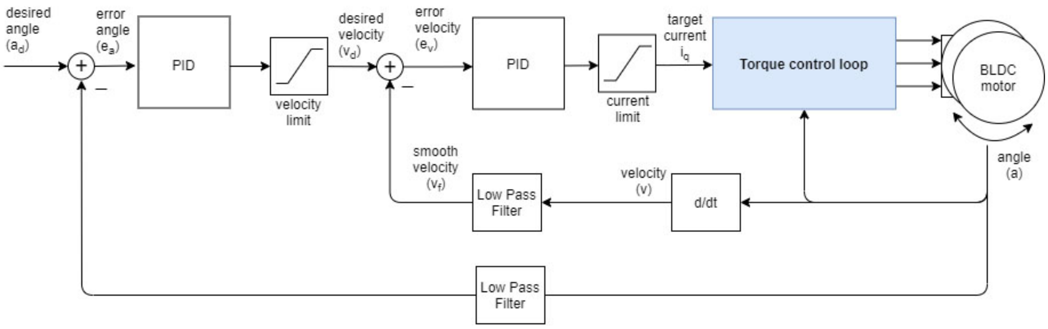
In addition to the already mentioned main classes, there are other secondary classes, which are used within the main classes. One worth mentioning, is the `PIDController` class, which is in fact, a classic implementation of an PID controller and which we will briefly go over below:

$$u(t) = Pe(t) + I\int_0^t e(\tau)\,d\tau + D\frac{d}{dt}e(t) \tag{4.1}$$

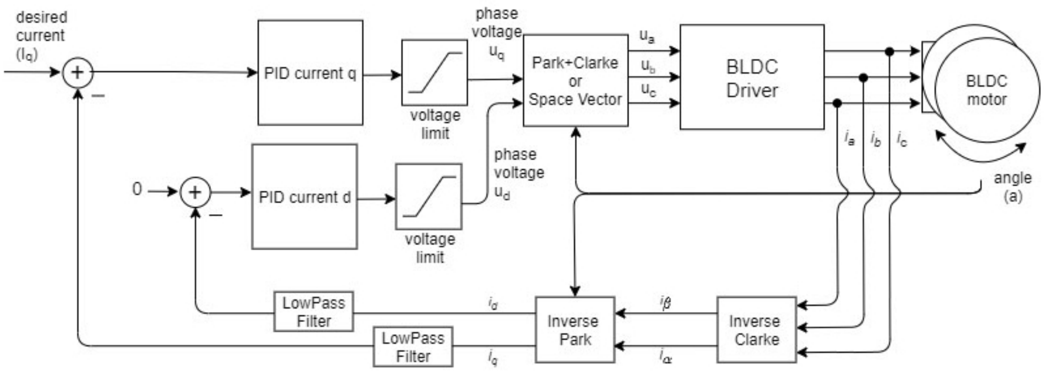
Where $e(t)$ is the error function between the set point and the current value of the measured quantity and $u(t)$ is the output reference. Also appearing in the equation are the constant terms P called the proportional term, I the integrative term, and finally D the derivative term; these terms are the parameters on which to act to change the performance of the control.

Moving to the discrete domain, we can represent the output $u(k)$ as a linear composition of three components:

$$u(k) = u_P(k) + u_I(k) + u_D(k) \tag{4.2}$$



(a) Typical motion control architecture, also implemented on SimpleFOC library



(b) Detail view of the torque control loop for the field oriented control method used in the SimpleFOC library

Figure 4.2: Entire control structure implemented in the SimpleFOC library [68].

Which were transformed from the continuous to the discrete domain in the following way:

$$Pe(t) = Pe(k) = u_P(k) \tag{4.3}$$

$$I\int_0^t e(\tau)\,d\tau = u_I(k-1) + I\frac{e(k)+e(k-1)}{2}T_s = u_I(k) \tag{4.4}$$

$$D\frac{d}{dt}e(t) = D\frac{e(k)-e(k-1)}{2}T_s = u_D(k) \tag{4.5}$$

Where T_s is the sample time of the discrete PID implemented in the specific MCU and the error $e(k)$ represents the difference in discrete time between the setpoint and the actual measured quantity, which will change depending on the PID in question (current, speed, and angular position). Figure 4.2 shows where these PIDs are used in the global control that the SimpleFOC library implements.

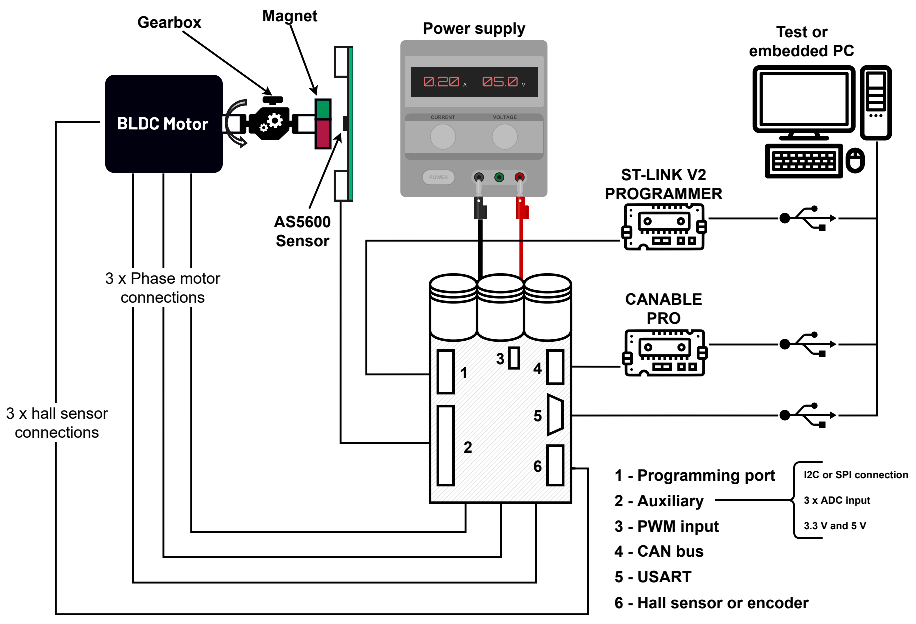


Figure 4.3: Block diagram of the electronic setup of the mechatronic joint under test.

To note how the term $u_I(k)$ and $u_D(k)$ (which are at current instant k) depend on the error function of the previous instant (namely $e(k-1)$). As an example, we give below, the error signal for the speed control PID:

$$e(k) = v_d(k) - v_f(k) \tag{4.6}$$

Where $v_d(k)$ is the desired speed and $v_f(k)$ is the actual speed.

4.2 Initial setup

4.2.1 Hardware setup

Figure 4.3 shows the block diagram of the experimental setup of the components selected for testing and control implementation. Moreover, the instrumentation in figure 4.4 was used, specifically in figure 4.4a we have the laboratory power supply used, a Hanmatek HM305P, which has sufficient features for use in these tests. It can in fact deliver a maximum power of 150 W with a voltage range of [5,30] V and a current range of [0,5] A. It has over voltage and over current protection, and the output is automatically disconnected in the case of short circuits. To make maximum

(a) Hanmatek HM305P

(b) PicoScope 2204A

(c) Gearbox assembly designed for testing

Figure 4.4: Additional material used for the realization of the experimental setup.

use of the power that can be delivered by the power supply, but at the same time, leave room for possible overvoltages, I set the working voltage for all experiments to 28 V.

In the figure 4.4b we have instead a portable oscilloscope, the PicoScope 2204A, with 10 MHz bandwidth and 8 KS memory sufficient and useful for small measurements, such as testing the commutations of the hall sensors of the BLDC motor.

Finally, shown in figure 4.4c is the gearbox designed and built for position control testing and load testing. I adapted the reducer in question from an existing design [69]; in fact, I added the motor coupling flange, the bracket for anchoring to the test bench and the end effector, with the housing for the position sensor, the mechanical limit switches with associated endstop switches for the emergency circuit and the attachment for the use of an aluminum profile from ALICE.

Such a reducer, is planetary type and has a stackable design, each stage has a reduction ratio of four and thus, by stacking three stages, a total reduction ratio of sixty-four was obtained; it has a production cost that is around €30 (bearings, screws and plastic material) and was made of PLA by FDM 3D printing.

The decision to use a prototype gearbox in the control tests, different from the one chosen in the design phase, has three main reasons:

- the first was to be able to test the motor and driver in conditions similar to the real scenario, without having the final gearbox available; in fact, the lead time for purchasing and logistics of such a gearbox would have been longer than the time for this thesis;
- the second is that the use of plastic components, allows for greater safety of the test bench; by doing so, in an emergency condition where the end effector reaches its mechanical limit, it is very likely to fail the weakest part of the system and thus, the gears of the last stage of the planetary gearbox, therefore disconnecting the motor from the end effector;

- the third, is that in the limiting case described earlier, that is, gearbox failure, with a modular structure there is more easily to repair it and at the same time with reduced cost.

Secondly, test bench safety is consolidated with the use of the laboratory power supply, in which a current (and consequently, power) limit is set and by software-level safety routines.

Finally, the experimental setup has in series with the aforementioned endstop switches, an emergency pushbutton, which can be pressed in an emergency situation to disconnect the positive power rail coming from the power supply.

Hall sensors low-pass filter

In the early stages of testing, I observed some instabilities in the speed control and therefore, possible causes of the problem were analyzed. In order to rule out any kind of measurement error of the hall sensors at the hardware side, I wanted to measure such commutations, through the use of the oscilloscope described and shown in figure 4.4b. From this evaluation, it was found that indeed the signal coming from the hall sensors has strong entities of noise, as shown in figure 4.5a, where some spikes can trigger the threshold value of the interrupt trigger, placed on the driver input, thus causing a false transition and thus, an incorrect motor speed measurement.

Such disturbances, are mainly caused by the movement of the rotor, in which the permanent magnets are placed. These changes in electromagnetic flux, are a source of disturbances not only for the hall sensors placed inside the motor, but can also be for devices close to it.

Analyzing the schematic of the driver in appendix B, we can see that in the input interface for the hall sensors, there is no low-pass filter to attenuate any noise, unlike other drivers seen in 3.2.3. This design choice was carried forward subsequent to VESC version 4.12 to overcome problems of incompatibility of such inputs with a filter, in case an encoder is used instead of hall sensors inside the motor. The lack of such a hardware filter, is compensated for in the VESC firmware with a software filter.

In Figure 4.5b is then shown the effect of such a low-pass filter, which was implemented on a breadboard to evaluate the actual benefit of it. As we can see, on the high logic value, there is a clear decrease in noise, and therefore it could actually help to avoid errors in motor speed measurements. It is also interesting to note that on the other hand, on the low logic value, there is no difference. To explain this phenomenon, it would be useful to derive a schematic or model, knowing the hall sensors used inside the motor. However, from the tests performed, such noise never exceeds the threshold value and therefore, I did not continue further.

Finally, for the purpose of fully analyzing the feedback from these sensors, we can see in figure 4.5c the jitter of a single hall sensor signal, measured by placing the control at constant speed. This jitter may be caused by an error in the relative positioning of the hall sensors within the motor [70, 71] or, to incorrect hardware or software filtering and subsequent calculation of rotor speed, thus sending incorrect references to the PID and resulting in current (torque) and speed ripples.

4.2.2 Software setup

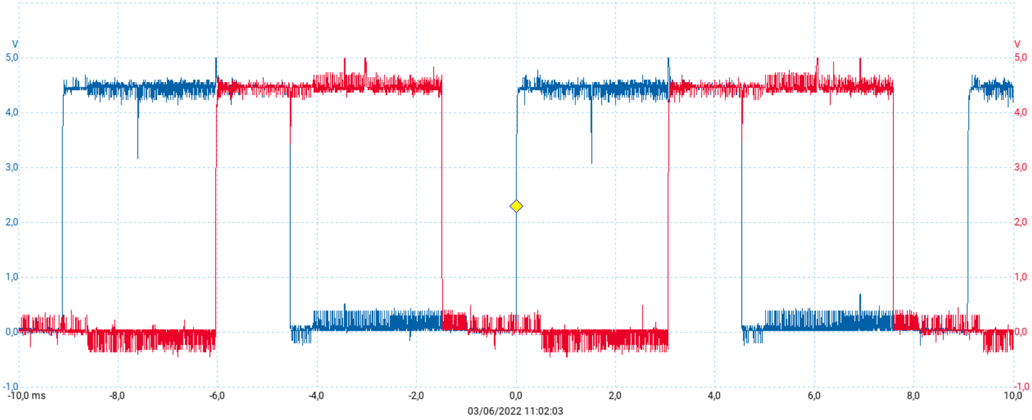
Switching instead to the software side, let us see the steps performed to set up the use of the SimpleFOC library, shown schematically in figure 4.1b. As a first step, following the driver circuit diagram shown in figure B.2 and B.3, shown in appendix B, I mapped the various GPIO of the MCU with the names of variables later used in the main of the project, and other low-level settings, such as clock configuration and the assignment of the various timers to the pulse width modulation (PWM) outputs. This map is given in the file `board_vesc_6.h` and then included in `main.h`.

With that done, I moved on to the definition of the project's main objects and variables. In the listing C.1 of appendix C, we find the definition of the `motor` object, initialized with the number of magnetic poles of the chosen motor; of the driver, which has six PWM inputs, mapped to the MCU; and finally, the sensors, the current sense, the motor hall sensors, and the position sensor placed after the reduction.

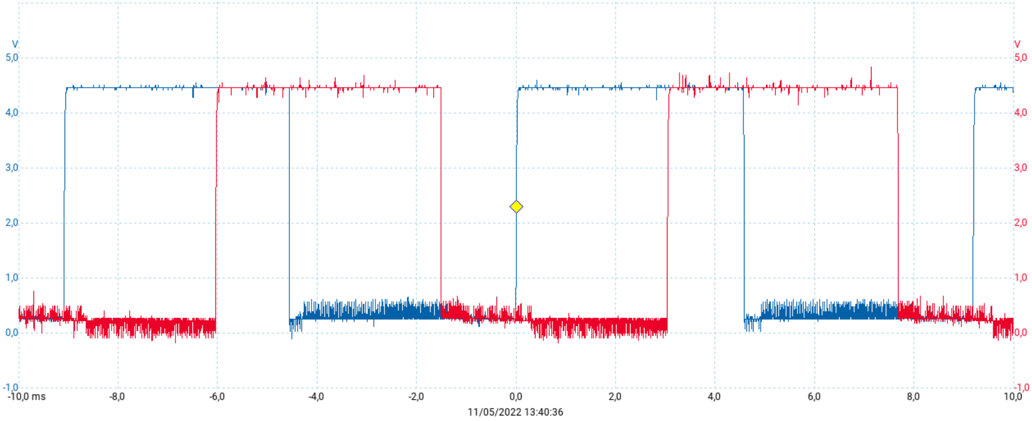
In order to use the post-reduction position sensor, for position control, and the motor hall sensors for the innermost loops of the control, I made a modification to the library using the classes and methods already made available by it. In fact, thanks to the `GenericSensor` class, it was possible to easily add the potentiometer reading (or in the final configuration, the AS5600) by simply customizing `readMySensorCallback` and `initMySensorCallback`. The implementation of these two callback functions are given in the listing C.4 of appendix C.

Finally, I initialized the objects related to serial communication and CAN bus communication. Note how both of them have the same structure and therefore, you can conveniently use the same command list for both serial and CAN bus, since the class `CANCommander` is derived from the class `Commander`.

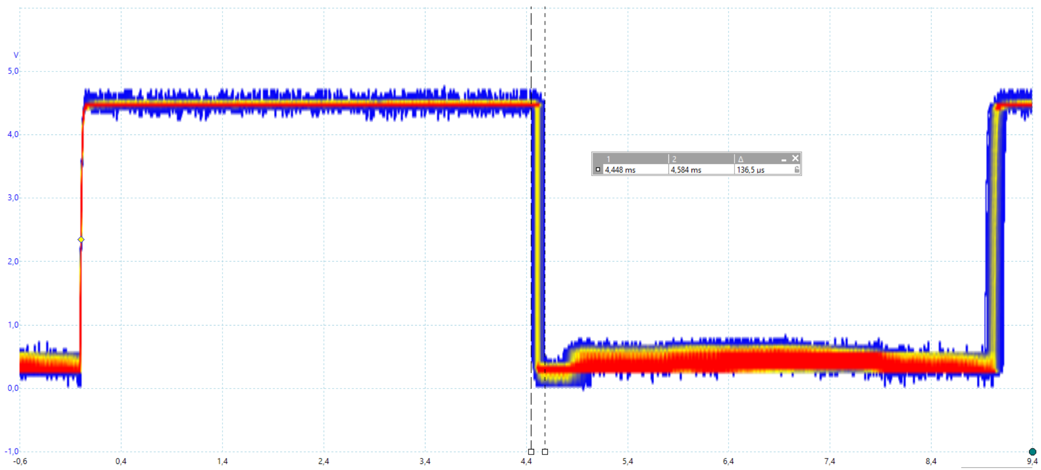
It is important to note that communication via CAN bus is not currently supported natively by the library, and therefore, I made changes to the design and SimpleFOC library in order to include the SimpleFOC CAN library for STM32, developed by the community [72]. Some communication tests were performed via CAN bus with the chosen USB dongle, the CANable PRO, in which it was possible to send speed



(a) Switching signals of two hall sensors of the motor, without the low-pass filter



(b) Switching signals of two hall sensors of the motor, with the low-pass filter



(c) Jitter of the signal of a hall sensor at stationary speed

Figure 4.5: Analysis of the switching signals of the hall sensors located inside the motor, referred to a speed of 50 $\mathrm{rad\,s^{-1}}$.

references to the driver via terminal on linux. Having ensured that the CAN bus communication worked properly, I then chose to perform all tests and tuning of the control via the serial interface, as a matter of convenience.

Continuing with the code settings, in the list C.2 we have an abstract of the `setup()` of the `main.cpp`, where I go to initialize the objects declared above (sensors, drivers, motor, and communications) and assign values to their key variables. For the motor, I then assign the sensors and driver to the associated object, via the `"link..."` methods. In the case of the driver, the library allows us to set thePWM frequency of the FOC, dead zone, DC bus voltage, and so on.

For the motor, on the other hand, I indicate the phase resistance and current limit (or alternatively the voltage limit) and in addition, in the latest versions of the library, also the KV rating, which is useful to have a correct estimate of the BEMF.

The most important part is related to the variables `motor.controller`, `motor.torque_controller` and `motor.foc_modulation`. The latter two, remain unchanged throughout the tests that will follow in this chapter, while as for `motor.controller`, there are more types, and the ones used are `torque`, `velocity` and `angle`.

In the following discussion, I will also go on to report the parameters of the aforementioned torque, speed and position controls, which were derived by manual tuning and thus, without any kind of theoretical or empirical technique (e.g., ziegler & nichols).

Finally, the serial is initialized, and the ability to monitor and plot certain variables for debugging purposes is enabled.

Modifications to the HallSensor class

Although a hardware low-pass filter was applied to the motor hall sensor inputs, as I continued with preliminary tests, I found that the speed control implemented with SimpleFOC continued to have instabilities. Using the VESC firmware as a comparison, I was able to outline the source of these instabilities and that is, the implementation of the `HallSensor` class.

Testing the function that calculates the motor speed based on hall sensors, I found that the implementation is not optimal and the function is not robust to disturbances.

In addition, because of the way the implementation is designed, I have found that the variable linked to the speed calculation sometimes undergoes an erroneous change of sign, and in addition, in the formula that returns that variable, a division by zero may occur. Such incorrect estimates, therefore bring false feedback to the speed-related PID, causing a sudden change in speed.

Once the system issues were identified, improvement changes were then made, which affect the `HallSensor` class. In the listing C.5 in appendix C I have given an excerpt of the main changes made, which impact the `getVelocity()` function.

Briefly describing these changes, I introduced the possibility of discarding for a certain number of times, values that are inconsistent (i.e., that deviate greatly from previous values) by reducing the division by zero and the change of sign as much as possible.

4.3 Tuning and test of the torque control

This section exposes the step response of the torque control, which, being the innermost one, plays a crucial role in the performance of the whole system. In figure 4.2b its implementation can be observed and in figure 4.2a instead, its use with cascaded speed and position controls. In the tuning phase, which as mentioned above is manual, I have preferred to relax performance requirements to avoid instability in subsequent control loops.

In the listing 4.1 the PID parameters for quadrature and direct current (being a vector control) are given. Note how the derivative term is not used since, through trial and error, I have seen that even with a small value, the control tends to become unstable. In addition, I have determined that as the derivative term increases, there is an increase in acoustic noise from the motor, at zero reference.

```
// Q axis
motor.PID_current_q.P = 0.1f;
motor.PID_current_q.I = 1;
motor.PID_current_q.D = 0;
motor.PID_current_q.limit = 0.1f;
motor.PID_current_q.output_ramp = 1e3;
motor.LPF_current_q.Tf = 0.1f;

// D axis
motor.PID_current_d.P = 1;
motor.PID_current_d.I = 10;
motor.PID_current_d.D = 0;
motor.PID_current_d.limit = 0.1f;
motor.PID_current_d.output_ramp = 1e3;
motor.LPF_current_d.Tf = 0.1f;
```

Listing 4.1: Tuning parameters for the PIDs of the I_q and I_d currents control loop.

Thus, we have in figure 4.6 the above step response of the torque control, performed by manually locking the rotor of the motor. In the upper part of the graph,

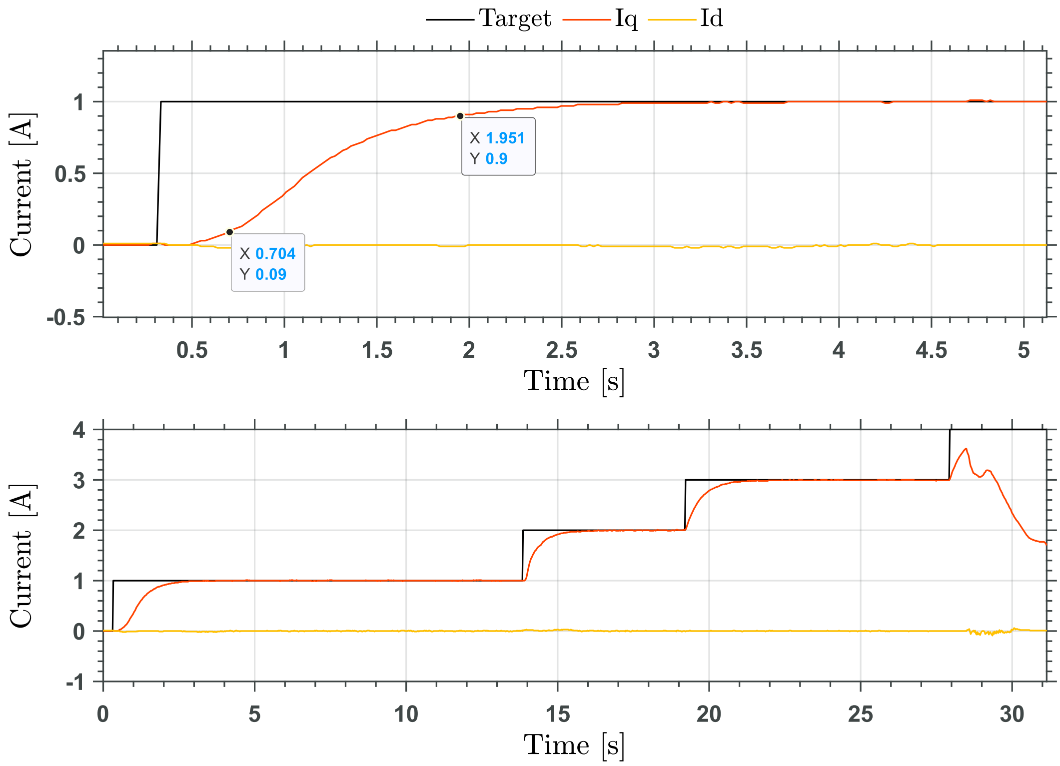


Figure 4.6: Step response of torque control loop.

we have the detail of the single step, and in the lower part, the succession of step references sent to the control.

As a positive note, we can see that the control has a first-order behavior and thus does not exhibit overshooting and oscillations worth mentioning. On the other hand, however, we suffer in terms of rise time, a useful parameter to quantify the responsiveness of the control. From the labels placed on the upper figure of the graph 4.6 we can calculate a $t_r = 1.25\,\mathrm{s}$. Note how there is an unusual initial delay from the time of command sent to the effect on the current I_q and hence on the torque provided.

This effect tends to decrease, as shown in the graph below, as the required current reference increases. As a final note, we can see how in the last reference sent to the control, the current exceeded the current limit that was set on the bench power supply, thus interrupting the test before the torque reference was reached.

4.4 Tuning and test of the velocity control

In this test, I set the variable `motor.controller` to `velocity` and again via serial, the step speed references are sent. Listing 4.2 shows the calibrated values for the

PID in question. The test was run with no load, with the motor anchored on the test bench without gearbox, so in the graph in figure 4.7 we have in the top, the behavior of the single step, where we can appreciate the performance of the control. In this case, we have a behavior that can be approximated to a second-order one (there is a slight overshoot but no oscillations) and as a metric we use the settling time at 2% and thus get $t_{s,\alpha\%} = 430\,\mathrm{ms}$.

```
motor.PID_velocity.P = 0.8f;
motor.PID_velocity.I = 0.2f;
motor.PID_velocity.D = 0.1f;
motor.PID_velocity.output_ramp = 1e3;
motor.LPF_velocity.Tf = 0.2f;
```

Listing 4.2: Tuning parameters for the PID of the velocity control loop.

Turning instead to some load tests at constant speed, we notice critical issues in the control system. These tests were performed before the improvement changes briefly described in the section 4.2.2.

In figure 4.8 I set the control in order to have 50 $\mathrm{rad\,s^{-1}}$ (i.e., about 477 rpm). Next, the application of a load by manual rotor friction is simulated, using a common work glove. After an initial settling of the current I_q to a steady-state value, the load is applied progressively, in the range of $3.5\,\mathrm{s}$ and we can see how an initial abnormal spike occurs on the motor speed graph.

Subsequently, near the gray dashed line intersecting the graphs at about $4.4\,\mathrm{s}$, we can see how there is a miscalculation of the instantaneous velocity (green curve) where a value of the opposite sign is computed ($-50\,\mathrm{rad\,s^{-1}}$) that greatly increases the value of the difference between it and the target reference ($50\,\mathrm{rad\,s^{-1}}$). This error in the input of the PID, results in a jump of opposite sign on the current I_q, as shown in the graph.

Such behavior, recurs several times in this test, with the worst effect toward the end of the graph, where we can see that around $11.3\,\mathrm{s}$ there was an almost constant load applied ($I_q \approx 6$) and suddenly, we record a flip of the current and speed measured.

After the modification of the `HallSensor` class, I tested the actual improvement in performance obtained. In the graph in figure 4.9 and in particular the second graph from the top (representing I_d and I_q and thus the load) we have three different load simulations, performed as previously mentioned, in a manual way.

In the first, we can classify them as load variations with an intermediate frequency; in the second, variations with a high frequency; and in the third, an attempt was made to simulate a constant load. It was tried, as much as possible, to maintain the same intensity (of the torque) and frequency, for each of the above phases.

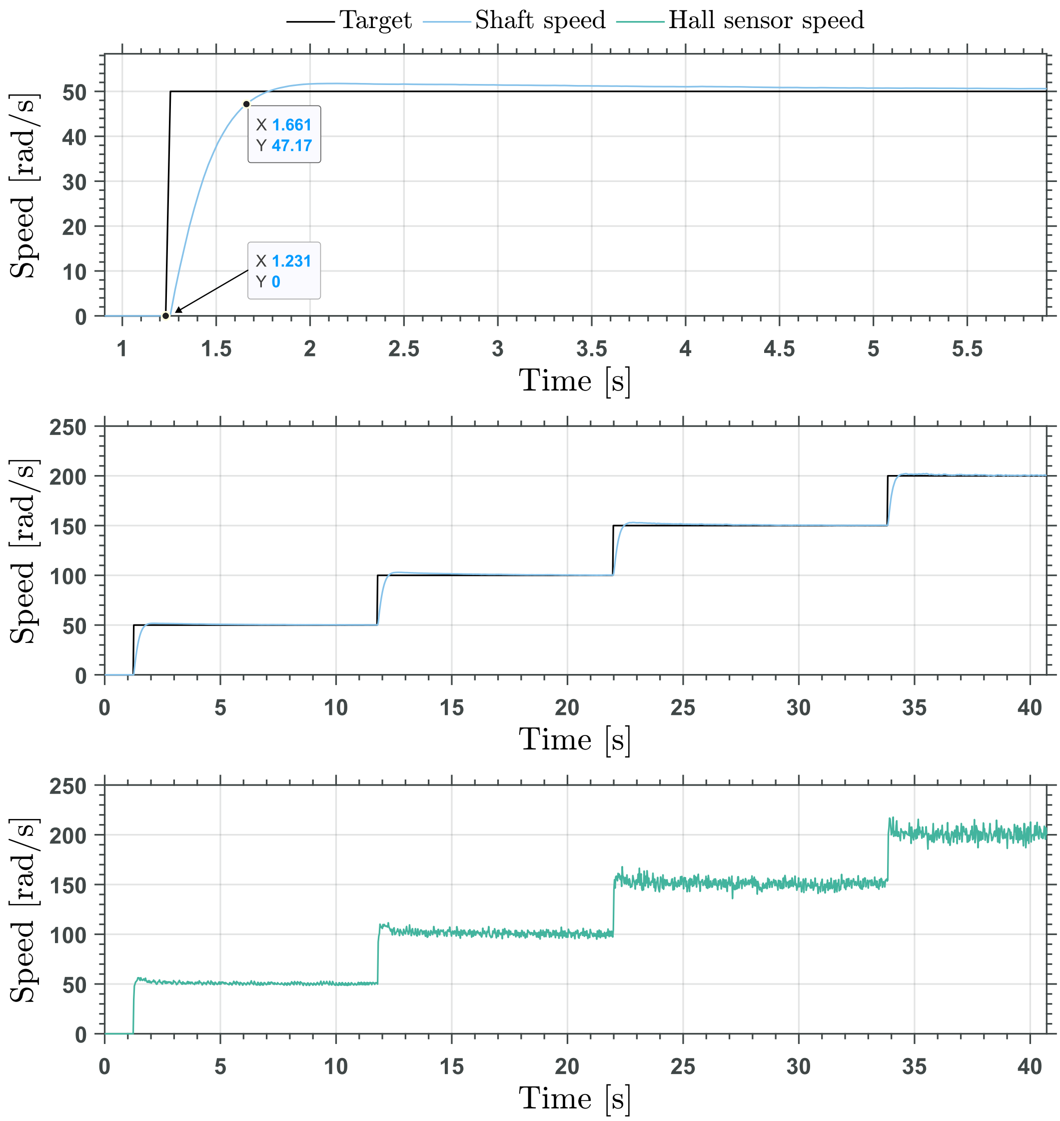


Figure 4.7: Step response of velocity control loop.

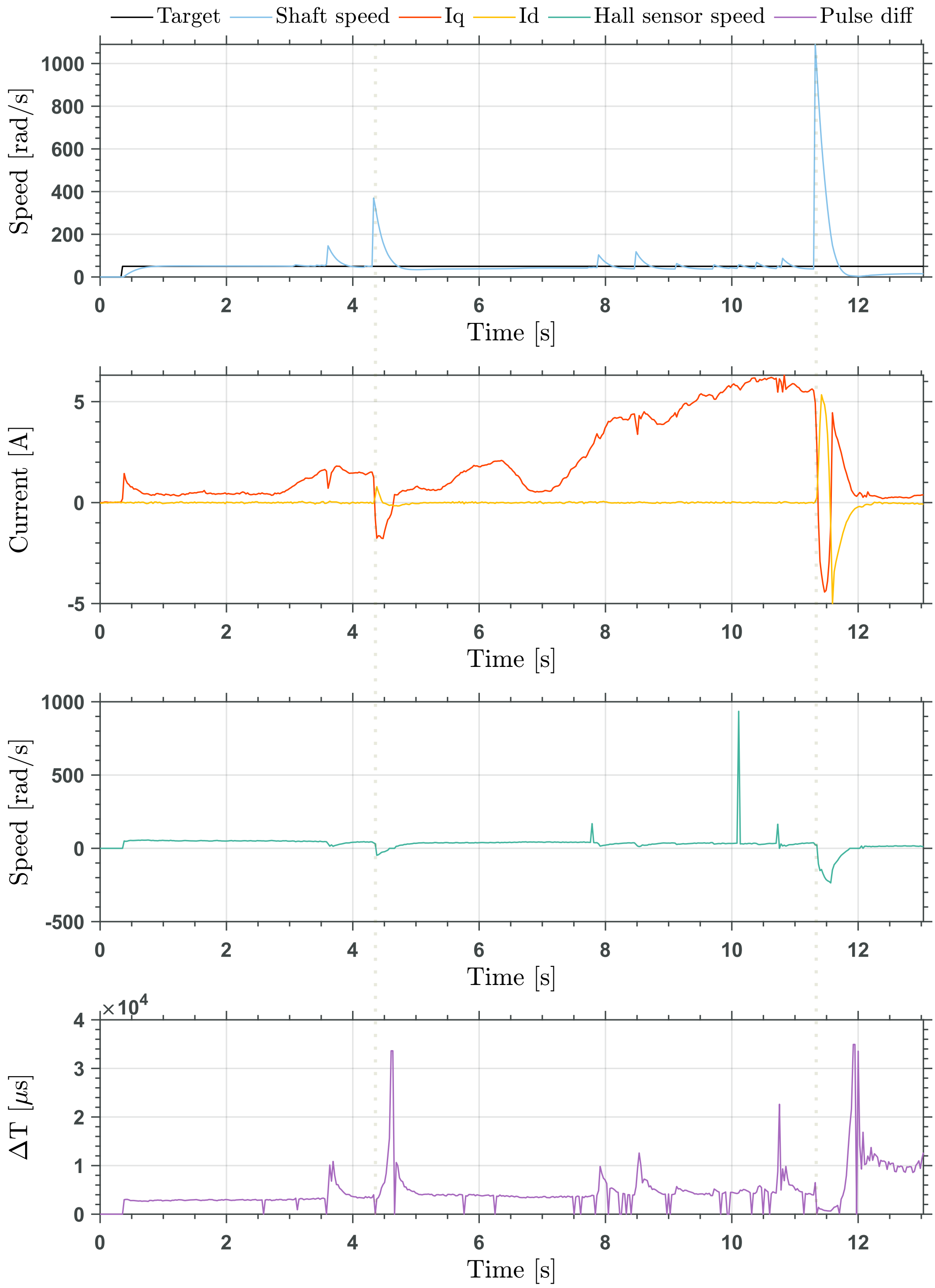


Figure 4.8: Stress test at constant speed (50 $\mathrm{rad\,s^{-1}}$) in which a disturbance, through a load, is applied to the motor.

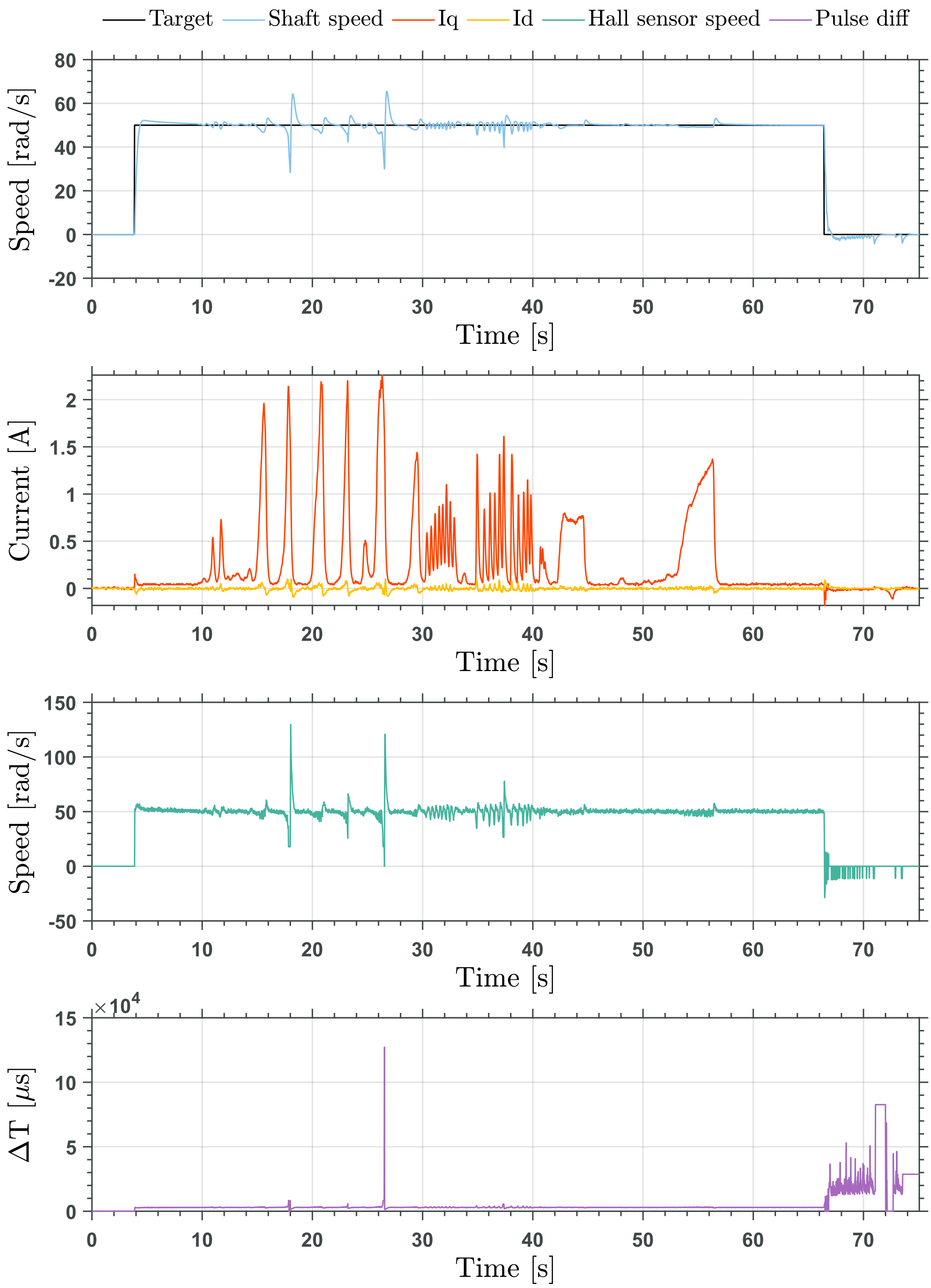


Figure 4.9: Stress test at constant speed (50 $\mathrm{rad\,s^{-1}}$) after an improvement of the library, in which a disturbance, through a load, is applied to the motor.

As we can see in figure 4.9 the behavior of the control is remarkably improved, thus suggesting the validity of the assumptions made (green curve). First of all, from the latter graph, we can see that there are only two instances in which the variable related to ΔT goes to zero, whereas in the graph in figure 4.8 (purple curve) there were about twenty occurrences (moreover, in a narrower time window). However, as can be seen, the problem still appears to be present, in a mitigated form.

In the first train of load pulses, two of them present a sudden change in the motor speed, and in fact, in the graph of ΔT we have in the first event a null value, and in the second, a very high value (about 13×10^4). In this graph, the behavior of the velocity calculation near the null reference (final part of the graph) is also shown, and as we can see, the calculation is not accurate and there is some period of instability in the motor stop phase. Although it was not present in the figure 4.8, this behavior was present even before the library changes and is related to the intrinsic implementation of the `HallSensor` class.

Steady state error and current analysis

To conclude, I wanted to analyze the difference under near steady-state conditions, between the speed reference and the actual motor speed. The graph in Figure 4.10 thus reports in the first graph, a difference of about $1 \mathrm{rad\,s^{-1}}$ and thus, an error of 2% in reference to a non-zero load producing an average current I_q of about $1\ \mathrm{A}$.

For expository purposes, in the graph below, the trend of the three phases of the motor was checked, where a sinusoidal trend can be vaguely recognized (as we expected from the type of control) due more than anything else, to the low sampling of these signals. In fact, it is not possible to increase the sampling and sending frequency, via the serial interface of these currents, as it would cause a slowdown in the performance of the control.

4.5 Tuning and test of the position control

In this last section, I performed the calibration of the final control loop, seen in figure 4.2a and that is the angular position loop. For this test, I first used the hall sensors inside the motor (figure 4.11) and then, the angular position sensor (potentiometer) placed after the gearbox (figure 4.12). This is made possible by the library modification performed, described briefly in 4.2.2.

In the listing 4.3 I report as done previously, the PID values, obtained by means of experimental tests. In the code listing, the parameterization of the sensor that is not used is commented.

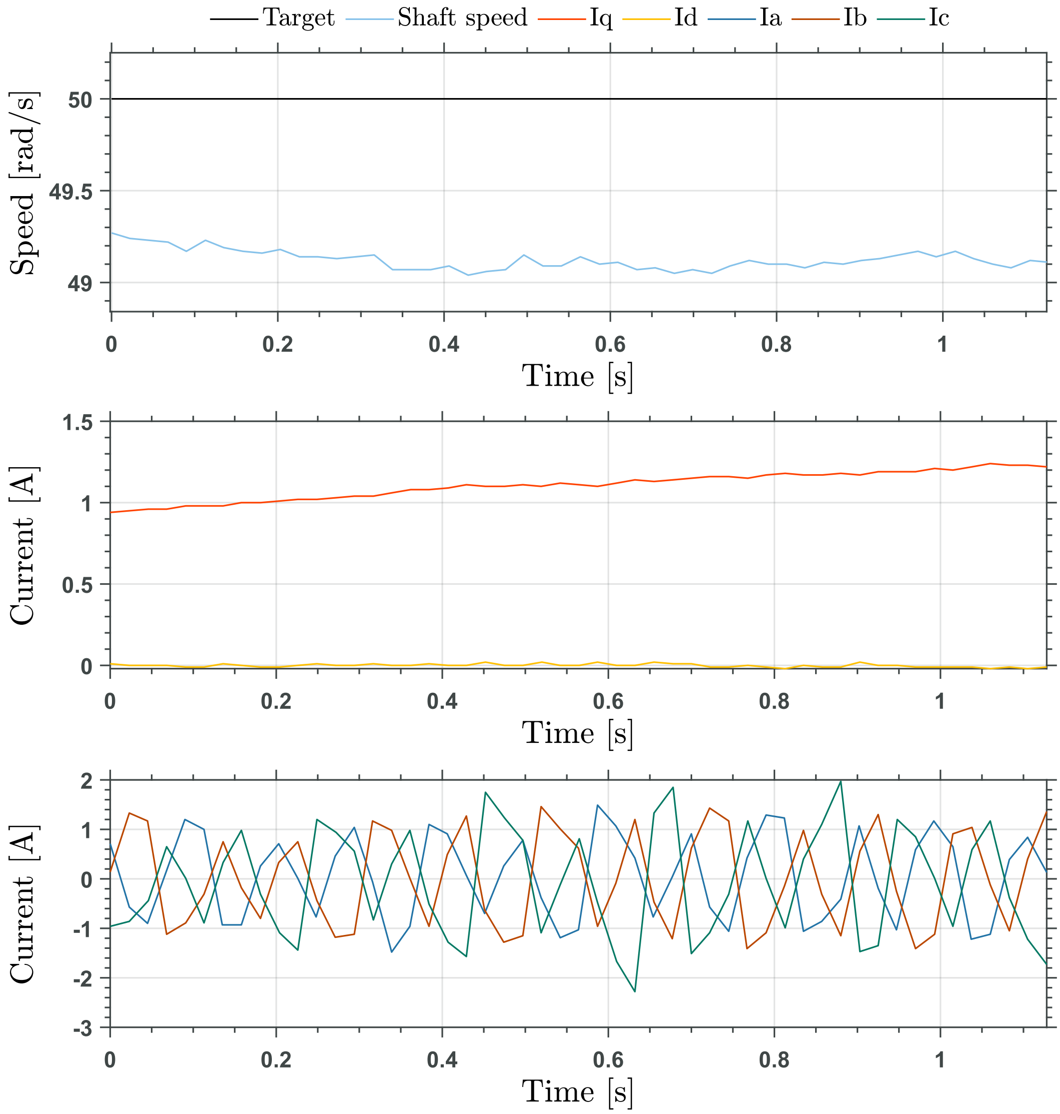


Figure 4.10: Analysis of the motor currents at steady state and under load conditions.

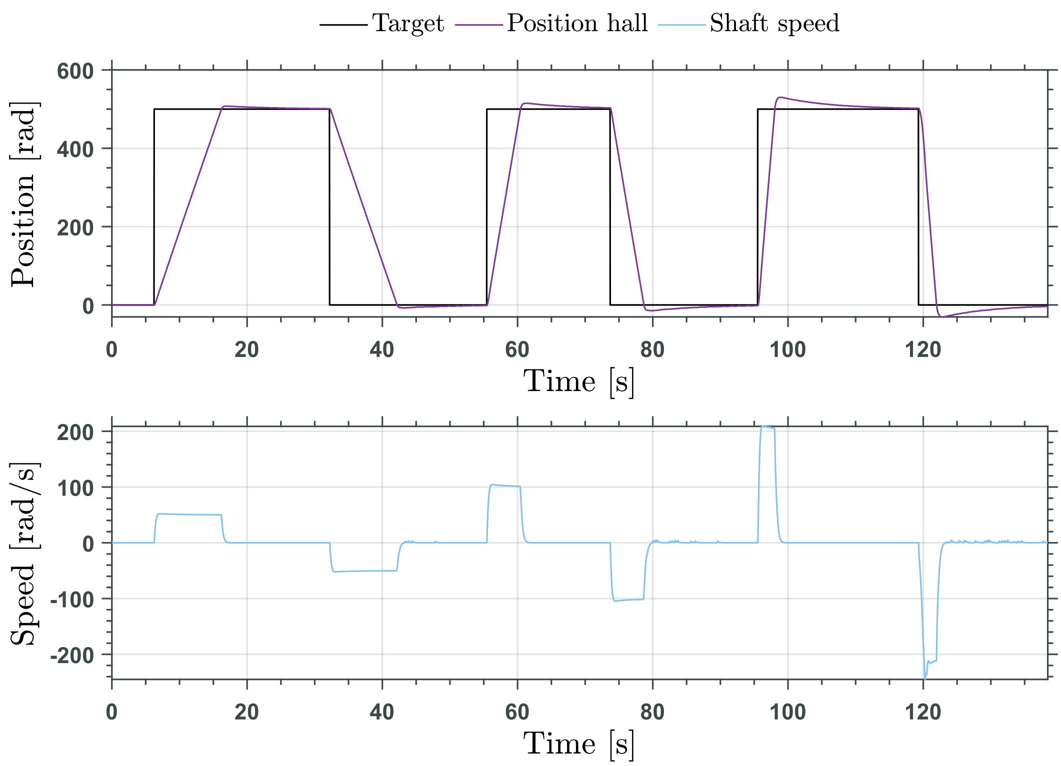


Figure 4.11: Position control behavior without gearbox using the hall sensors, at constant reference, changing the speed limit value.

We can see that in the case of the angular sensor placed after the reduction, we have a much larger proportional value P (but also I) compared to the parameterization for internal hall sensors. This is due to the fact that the angular position is reduced, in the case of the gearbox used, by a sixty-four ratio.

Moving then to the analysis of the graphs just mentioned, we can see the performance of the control based on hall sensors in figure 4.11. Three angular position references are presented, which in this case are very large (the motor has to make about 160 revolutions) at which a different speed limit is set; in sequence I set a speed limit of 50, then 100 and finally 200 $\mathrm{rad\,s^{-1}}$.

In the first graph from the top, we can see the effect of these speed limits, namely, the slope of the initial and final ramp to reach the required target. Note how in the first reference there are no noteworthy overshoots, while from the second and lastly in the third, a noticeable overshoot is visible.

To be considered that given the chosen configuration (motor and gearbox) depending on the torque to be obtained in the final application, it determines a lower maximum speed rather than the one used in the present test (200 $\mathrm{rad\,s^{-1}}$ and thus about 1910 rpm) chosen to test the limits of the control.

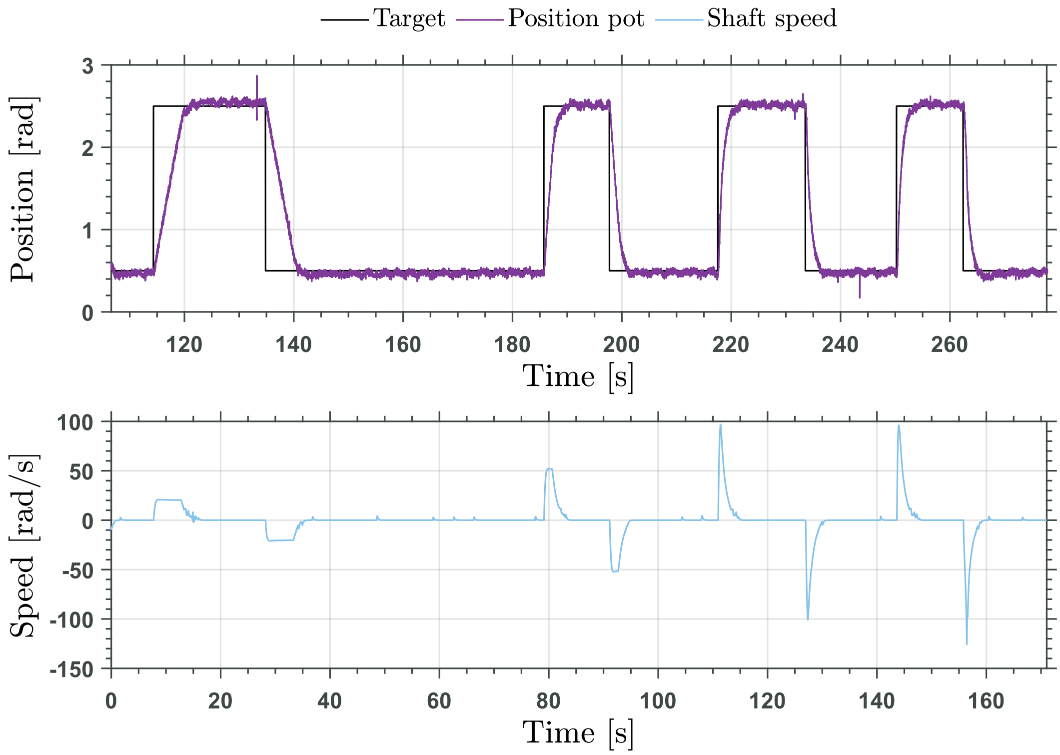


Figure 4.12: Position control behavior with gearbox using the potentiometer, at constant reference, changing the speed limit value.

In the graph in figure 4.12, the plastic gearbox in figure 4.4c and a potentiometer as the angular position sensor are used. In this case, the position target is lower (41 motor revolutions, to achieve about 2 rad) and we use the same speed limits as described above. We note how the control reaches the required target even before reaching the speed limit, since the angular position reference is lower than in the previous case.

We see that in the second graph in figure 4.12, in the last iteration and more specifically, in the zero reference (about 0.5 rad to avoid touching the endstop) we have a slightly higher velocity limit than its positive mirror. One explanation could be the weight of the rod, which in one direction is antigravity, while in the other case, in favor of gravity.

```
/*
  // Position parameter (hall sensors)
  motor.P_angle.P = 6;
  motor.P_angle.I = 0.8;
  motor.P_angle.D = 0.5;
  motor.P_angle.output_ramp = 1e4; // default 1e6 rad/s^2
  motor.LPF_angle.Tf = 0.1f;
*/

  // Position parameter (Gearbox sensor - 64 ratio )
  motor.P_angle.P = 150;
  motor.P_angle.I = 10;
  motor.P_angle.D = 0;
  motor.P_angle.output_ramp = 1e4; // default 1e6 rad/s^2
  motor.LPF_angle.Tf = 0.1f;
```

Listing 4.3: Tuning parameters for the PID of the velocity control loop

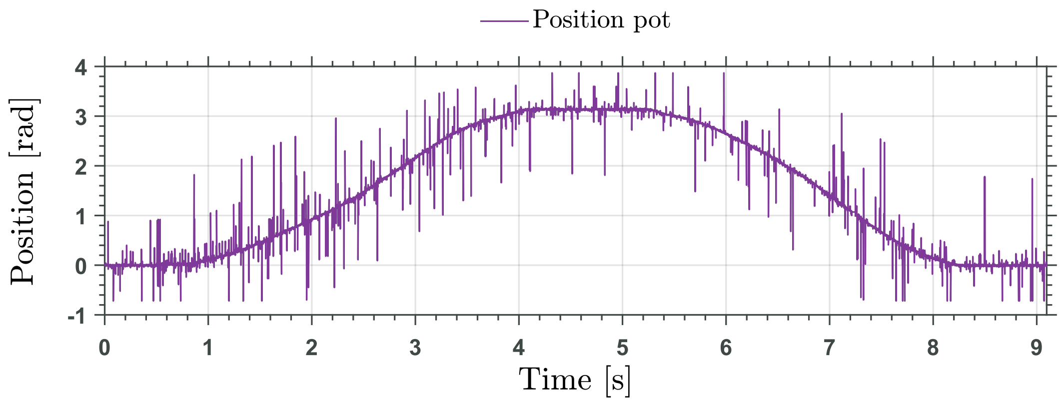
Potentiometer and AS5600 angular position sensor

To conclude, we observe that the potentiometer signal in figure 4.12 was very disturbed, and that this fact compromised the performance of the control.

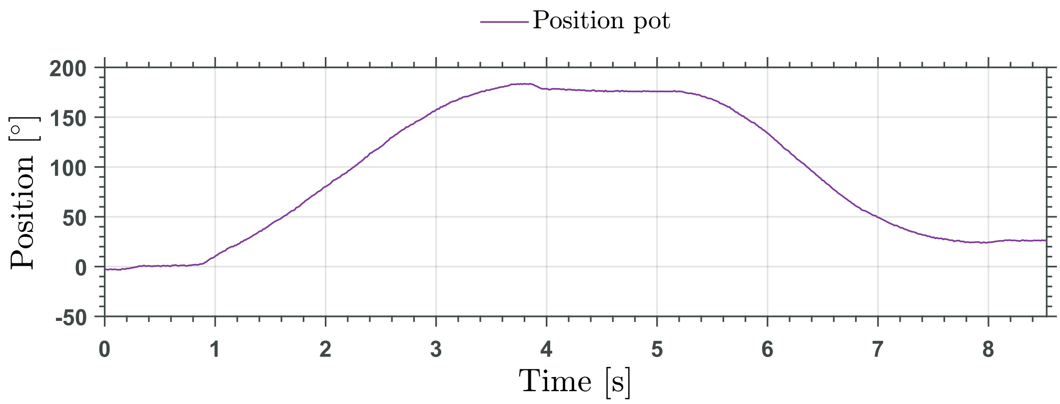
To improve the quality of the potentiometer signal, figure 4.13 shows the difference between the unfiltered signal, with the 10-bit resolution of the ADC (default configuration in the SimpleFOC library) in figure 4.13a, with the same potentiometer, but setting a resolution of 12-bit and applying a filter visible in the listing C.4 in appendix C. The result of such filtering is shown in figure 4.13b.

In the last graph (figure 4.13c), the operation of the magnetic angle sensor seen in figure 3.5f was tested. The digital signal from it was read via its I2C interface (with connection shown in figure 4.3) and has the advantage of not needing any filtering; comparing it with the potentiometer signal, the latter is better, in addition to the safety benefits that serial communication benefits (we can handle sensor breakage or magnet misplacement).

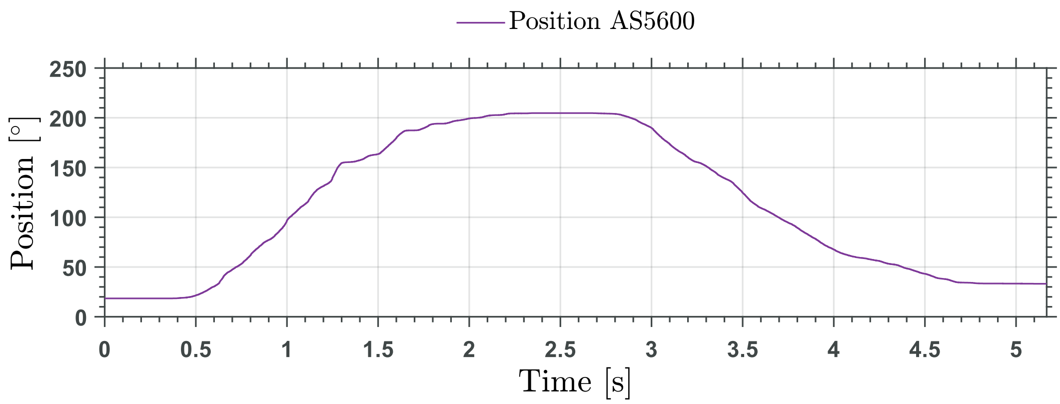
The signal of the AS5600 has fluctuations that are simply due to the fact that it was not tested on the actual gearbox, but by means of a knob (which had some friction on the centering flange) and a manual movement. If we compare the two graphs in the stationary sections alone, we can appreciate the above peculiarities.



(a) Unfiltered signal from potentiometer (10-bit resolution)



(b) filtered signal from potentiometer (12-bit resolution)



(c) Signal from the AMS AS5600

Figure 4.13: First comparison of pre-filtered and post-filtered signal of potentiometer; second comparison with filtered signal of potentiometer with that of AMS AS5600 angular position sensor.

5 Validation tests

> *Science progresses by trial and error, and when it is forbidden to admit error there can be no progress.*
>
> — **Joan Robinson**
> Economist

IN this chapter, we will conclude the experimental part carried out so far in this thesis work. The aim is to validate the performance individually obtained for each control loop, globally and under real and dynamic conditions, with or without load. The goal at this stage, therefore, is not to implement an efficient and definitive high-level control, but rather a test that can confirm the functionality of the low-level control and subsequently, to investigate and experiment on the best high-level control strategy to be implemented in the final version of the exoskeleton.

5.1 Predefined gait pattern

In order to best simulate a real condition, I wanted to take as a reference for the control system used so far, the knee gait profile on healthy users [73, 74] and with regard to weight, I referred to the anatomical tables given in [75] in which it is shown that in the case described above, namely a person 180 cm and weighing 80 kg (same anthropometric values as in the table 2.1) we have a leg weight of 3.6 kg and a limb length of 0.48 cm.

Therefore, for practical purposes, I found it reasonable to use a weight of 5 kg and an aluminum profile of length 0.38 cm for the present tests, obtaining under worst-case conditions, torque values similar to those that would be obtained with the theoretical values obtained from the article [75]. In figure 5.1 it is reported the actual experimental setup, seen before in a schematic way in figure 4.3, where we can find all the elements mentioned before.

5.2 Control state machine

To simplify the discussion, in figure 5.2 the block diagram of the script carried out on Matlab for low-level control validation is exposed. As can be seen, the script is

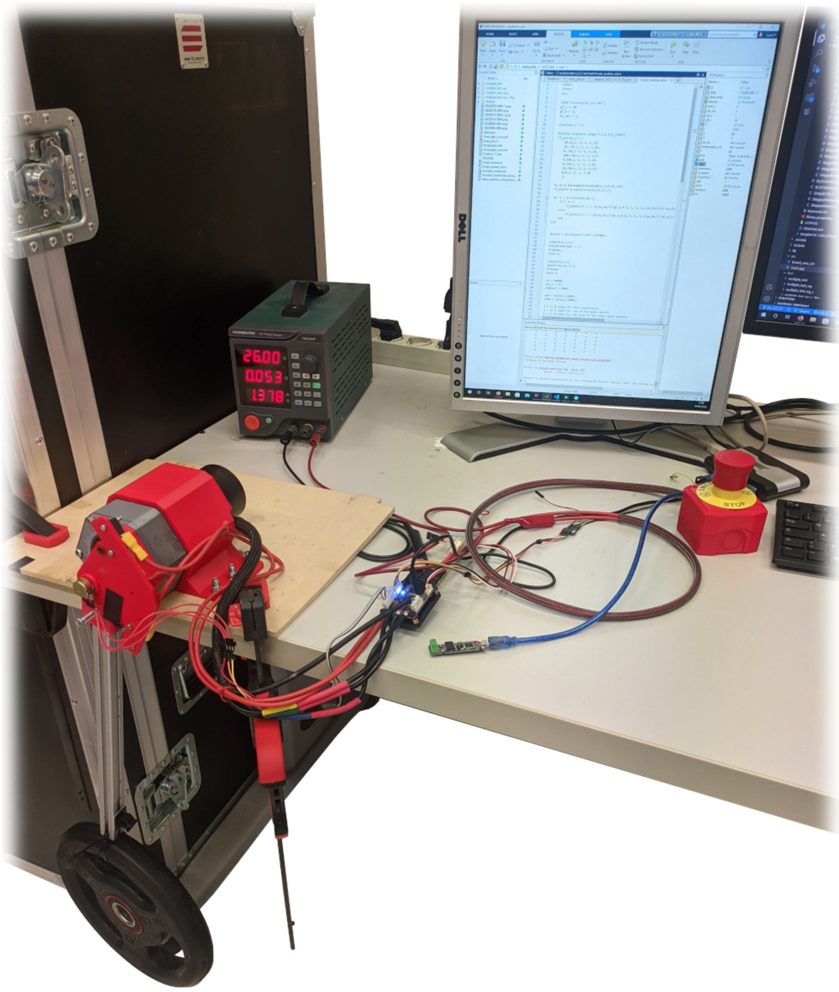


Figure 5.1: Implementation of the experimental setup to perform tests on low-level joint control at the IAS-Lab, University of Padua.

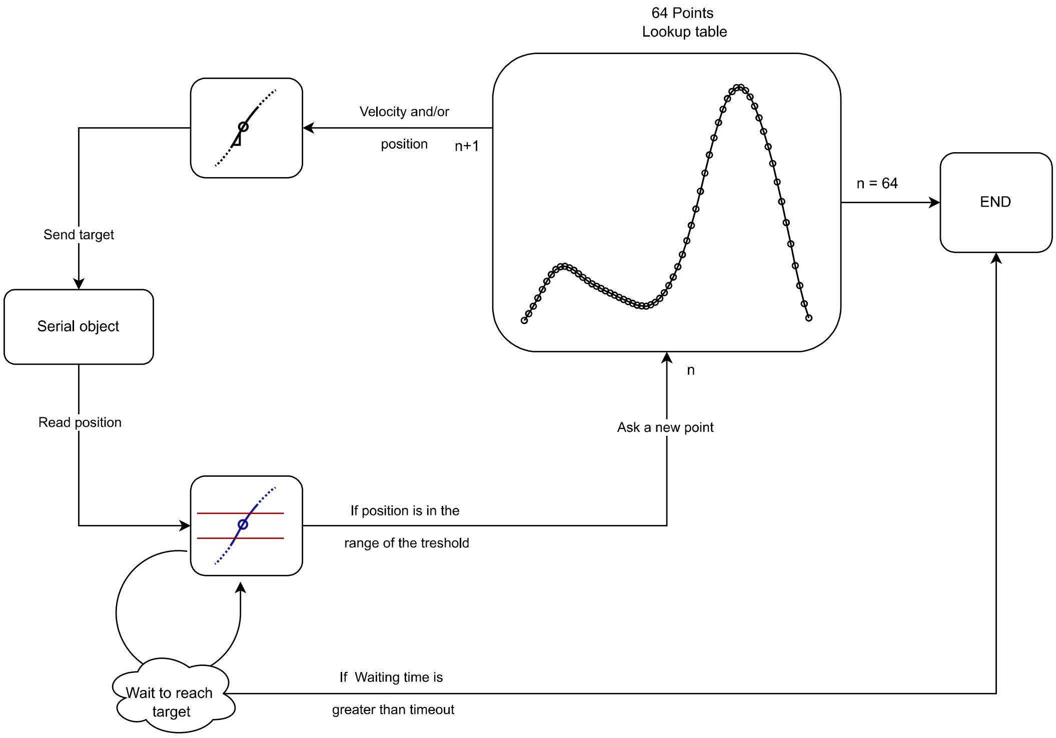


Figure 5.2: Block diagram describing the operation of the MATLAB script for low-level control validation.

a simple state machine, taking values from a lookup table, which is the result of a downsampling of the step profile reported as mentioned in [73, 74].

Starting from the first point in the lookup table, we begin to send each point in sequence to the driver via the serial interface. For each one, we wait until the set target is reached, and then we read the post-reduction angular position value, again via serial, that the driver communicates. At this point, there is a simple wait, and the output condition is relaxed to a range of values that is determined by $v(n) \pm th$, where n is the index associated with the setpoint that we are sending, while th is an integer threshold value that for simplicity has been chosen for trial and error, equal for each n.

To conclude, the exit condition occurs on two occasions: when the wait for the setpoint to be reached is greater than a certain timeout value, or when the end of the lookup table has been reached.

As a final note, the finite-state machine can send either a position reference or a velocity reference, the latter calculated by doing the discrete derivative between point n and point $n + 1$.

5.3 Trials performed

For each trial that will be shown in the next paragraphs, I performed $N = 10$ consecutive trials, without discarding any trials in between. For clarity of exposition, in the graphs that follow, I am going to report the following curves:

Target trajectory this curve is the envelope of points coming from the lookup table and constituting the gait profile we want to obtain.

Mean is the average curve, defined as the sum of the values obtained over all trials, at the n-th instant, divided by the N trials.

Median is the curve of median values, obtained by rearranging the values taken by the trials and choosing exactly the central value, for each n-th instant.

MAD the Mean Absolute Deviation is a measure of the dispersion of a given data set; the formula used is the following:

$$\frac{1}{n}\sum_{i=1}^{n}|x_i - m(X)|. \tag{5.1}$$

where $m(X)$ is the average value of the data set, n the number of the data values and x_i the data values in the data set [76].

The MAD is then applied to the median curve in which the mean deviation is minimized. The MAD median offers a direct measure of the scale of a random variable around its median.

5.3.1 Position reference test - no load

In this test, shown in figure 5.3 we have a first analysis of the behavior of the control system under dynamic conditions. I have chosen to use a position reference, and as we can see from the graph, we have poor performances. Evaluating the individual trials, it was noticed that the behavior of the position loop PID (see figure 4.12) upon approaching the reference given to it, is a first-order one (due to the calibration done); therefore, since there is no overshoot, we have a slowdown in terms of speed and acceleration, resulting in irregularities that occur near the next reference to be given to the control system and thus, appreciable as well in the average curve shown in figure 5.3.

From this test, I could see that using a position control is not an effective way to be able to replicate the gait profile intended. By repeating a few tests with the chosen load, this behavior became so pronounced that I could no longer get out of

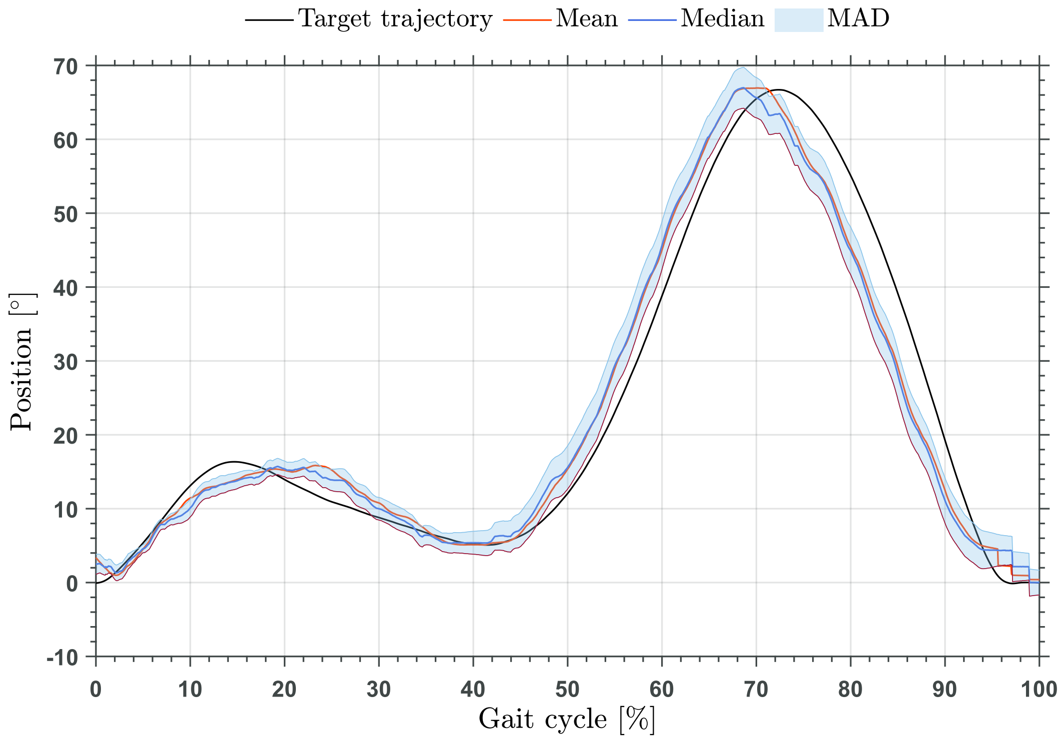


Figure 5.3: Low-level control response to positional references, without applied load, in order to replicate a gait (knee) profile.

the minimum threshold required to give a new position reference, locking the rod of the joint into the last position reference sent.

This behavior is due to the fact that as the position reference is approached, the position error given by the difference between the reference and the feedback is very small, and due to the chosen parameterization of the PID (and by varying the load) the timing required to follow the gait profile is not met.

5.3.2 Velocity reference test - no load

From the experimental observations seen in the previous paragraph, I wanted to change my approach, thus excluding position control and then moving to the next loop, namely velocity control. In this case I obtained acceptable results, shown in figure 5.4 in which we can clearly appreciate the smoothness of the curve, compared to the previous figure 5.3. Note how in the first part, the control follows the reference curve perfectly, while in the second part of the graph there is a lag, while maintaining the same shape (note that the constant velocity sections are actually parallel to the counterpart of the gait profile to be followed).

This delay is caused by the implementation of the finite-state machine, more specifically by the threshold for sending the next reference, which if it has too large

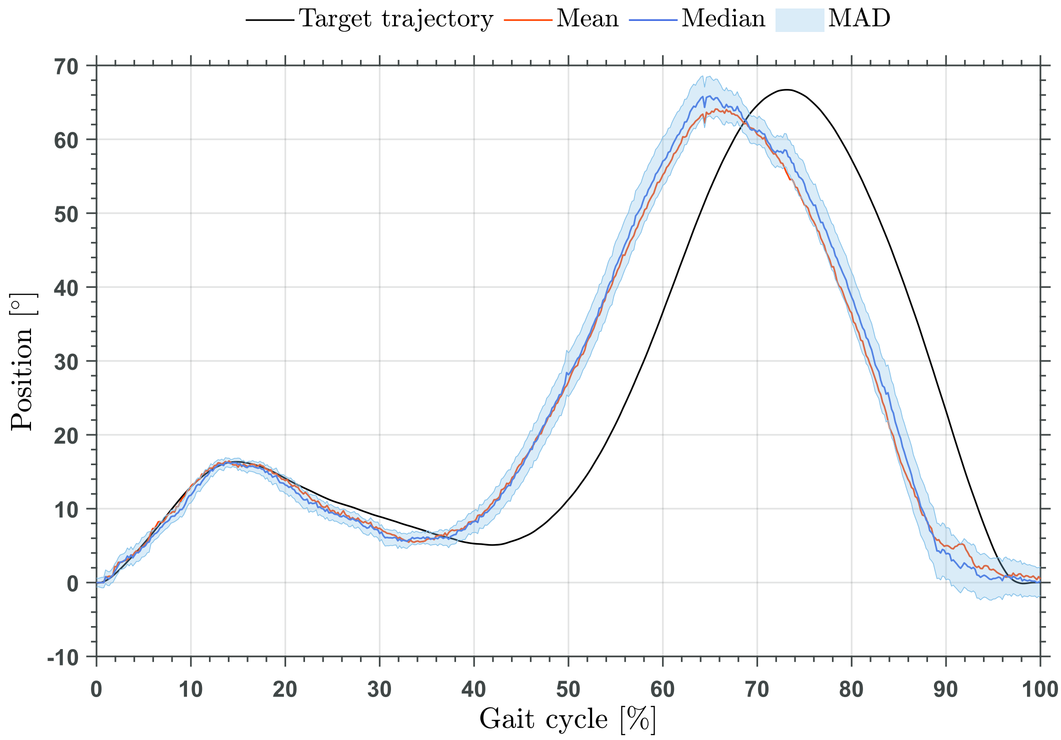


Figure 5.4: Low-level control response to speed references, without applied load, in order to replicate a gait (knee) profile.

value, it causes an anticipation of the control curve, relative to the reference curve. This behavior is particularly pronounced in sections where the derivative of the joint position is zero, namely, at zero velocity.

We also note how the second section at constant speed (starting from the left, the change from around 17° to 5°) fails to follow the designated gait profile correctly, and then settles down almost parallel. We find similar behavior in the change of direction of around 65°, which again, settles down in a short time.

Finally, we observe that the tracts of greatest variability that were recorded in the trials were in the peak of 65° and in the final slowdown of 0°.

5.3.3 Velocity reference test - with load

Following the positive results of the speed control test shown in the previous paragraph, I then applied the load to the experimental setup I made and observed the effects on the control (figure 5.5). First, we can say that all trials ended successfully, being able to move the weight with ease, without the state machine getting stuck. Secondly, we note the sharp worsening of the control behavior, particularly near the critical issues just described.

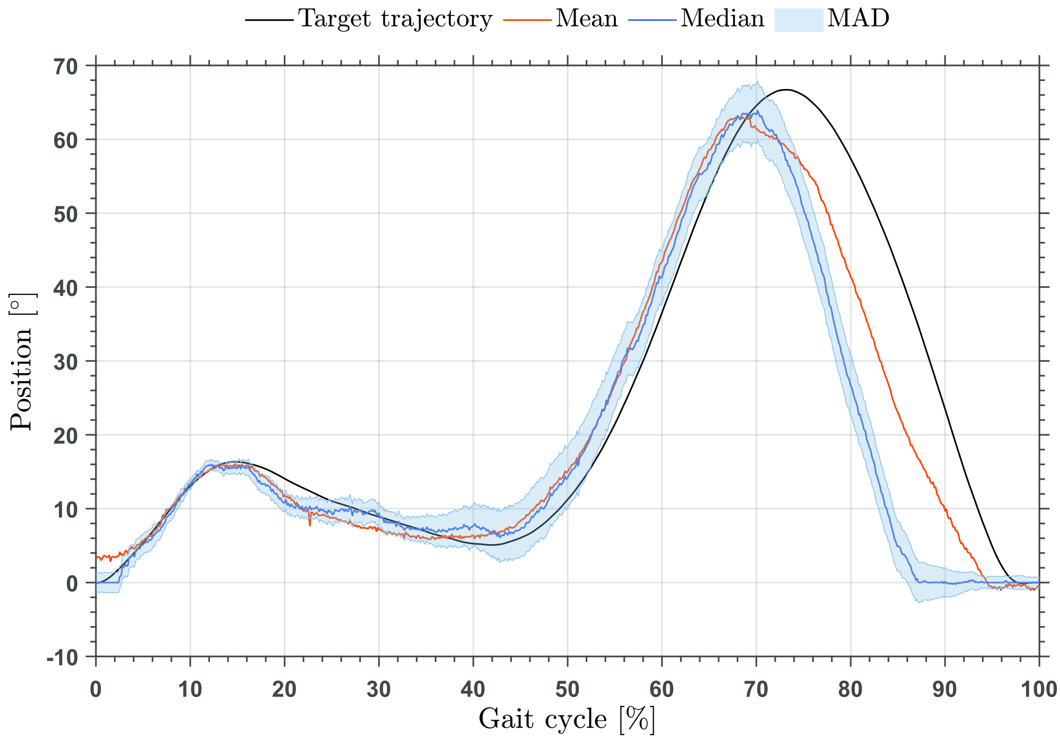


Figure 5.5: Low-level control response to speed references, with load applied, in order to replicate a gait (knee) profile.

It should be specified that in the described sections, the control is dealing with the load shifting from against gravity, to in favor of gravity, having to counterbalance this sudden change quickly (observe the image of the experimental setup, from figure 5.1). Therefore, the present test does not accurately simulate knee motion, in a gait profile performed in plane by a healthy subject, but presents a worse condition than true use of the joint for the exoskeleton.

In this test, I was also able to verify the power absorbed by the mechatronic system, which settles around 5 W in the 90° rod position with the aforementioned 5 kg.

6 Conclusions and future developments

> *The search for knowledge is a long and difficult task.*
>
> — **Fabiola Gianotti**
> Physicist

THE main objective of the present work was to improve a lower limb exoskeleton present at the Intelligent Autonomous System Laboratory (IAS-Lab) of the University of Padua. Along this path, I have faced various problems, which have led to review and study of new solutions. This final chapter will therefore comment on the steps taken during this experience. Finally, some possible future developments of the project will be outlined.

6.1 Discussion

In this thesis, in the first chapter, the concepts of fundamental importance in order to better know the reasons and choices behind the design of lower limb exoskeletons were exposed. Then in the second chapter, it was possible to leverage these concepts, in order to analyze and understand, what have been the design decisions of an existing exoskeleton, namely ALICE, available in the IAS-Lab. From this analysis, I was able to outline goals for improvement, based on needs supported by the scientific literature, which led to a path of product research and exoskeleton redesign.

Therefore, in the third chapter, I adopted concurrent engineering techniques from a mechatronic perspective that led as final output to conceptual sketches and the definition of the entire structure of the possible new exoskeleton or an improved version of the existing one. The commercial products found, were analyzed using a methodical approach and the final choice was guided according to the project specifications. Most of the products chosen are low-cost and open-source, but at the same time they offer market-leading performance, as demonstrated by the comparison tables.

At this stage, the use of an harmonic drive was chosen, supported by the articles mentioned in 3.2.1 in the context of lower limbs exoskeletons. The chapter 1.2.2

had set out the main differences between harmonic and cycloidal drives. An analysis of the scientific literature showed that the cycloidal type is commonly used in the field of industrial robotics, unlike applications on exoskeletons, which at present, have few noteworthy implementations with such a design choice. One of these is the exoskeleton presented by Angel Robotics, the WalkON Suit [77], which ranked first in the related competition to the Cybathlon 2020 (international contest in which physically disabled individuals compete in performing daily life tasks through the application of cutting-edge technologies) [78], demonstrating the successful design of such an exoskeleton.

As discussed in the conclusions in [37], cycloidal drives should be considered for applications in WR, especially those in which size, inertia, and efficiency take precedence over backlash and torque ripple. As seen in chapter three, the axial size is a crucial design parameter for the joint of an exoskeleton, as much as inertia, as seen in [40] for the proper implementation of the transparency of the system, from the perspective of the user of the WR.

Moreover, the choice of using a motor with far higher torque than the Maxon EC60 (which, as mentioned, is a classic choice in the scientific literature) is preferred in view of obtaining a transparent control, as described in [40]. At the same time, it makes it possible to reduce the gear ratio of the gearbox, in line with the choices made for the cycloidal gearbox in [77] which has a reduction ratio of 31 and a motor with a nominal torque of 3.7 $\mathrm{N\,m}$ (even better performance than the one chosen in this thesis, with 2 $\mathrm{N\,m}$ of nominal torque).

As mentioned, a hybrid solution shown in figure 3.2c could be more flexible for an exoskeleton for research purposes. Designing a joint that includes a cycloidal gearbox produced by FDM techniques and commonly used mechanical parts could bring the advantage of reducing costs, as reported in [79], decreasing the lead time for purchasing the parts needed for the project and increasing the repairability of the robot.

Regarding the torques produced by a possible gearbox of this type, there are promising findings coming from DIY robot projects [80, 81] and thus confirm the advantage of designing such a gearbox. This choice is also motivated by the actual operating time of the joints in a research exoskeleton, which is certainly low in magnitude and impulsive in type, as opposed to joints in industrial robots, which must provide much longer life cycles, with movements involving the use of the joints continuously and with the requirement to have low backlash and low torque ripples.

The design solutions carried out in this thesis work, are therefore in line with recent technological developments on lower limb exoskeleton, as seen in the scientific article on the design of WalkON Suit by Angel Robotics [77] and supported by the previously cited articles [37, 79, 40].

In the fourth chapter of this thesis, I then went into detail about the software implementation of the low-level control of the joint, having chosen the SimpleFOC library on Arduino framework for the development of the motion; each control loop in that library was tuned using a classical approach. In order to develop the control, I prepared an experimental setup, in which I used a 3D-printed gearbox to simulate real conditions and increase the safety of the test bench, as well as speed up the development and final design of the joint.

Limitations of this library emerged from this development, and these were investigated in depth; improved solutions were then proposed that resulted in performance benefits. These limitations, seen particularly in the graphs in Figure 4.8 have recently been solved by the SimpleFOC community [82]. The problem was indeed related to the function `getVelocity()`, as correctly found in chapter four. The division by zero was caused by the variable used in calculating the speed of that function, which was being overridden by a callback function, generating the problem exposed in chapter three. In the listing C.5 it is possible to compare these improvements with those developed in the aforementioned chapter.

It turns out to be important, therefore, to combine the latest library developments with those pursued in this thesis work. In addition, it might be useful to repeat some measurements and automatize experiments to evaluate the performance of the various control loops. In the case of the load simulation described in 4.4 and seen in figure 4.9, being a manual procedure, it is difficult to repeat and with empirical disturbance parameters. The introduction of a second motor (coupled with the Flipsky 6354), suitable for load simulation, would allow a more effective analysis of system performance.

The measurement of phase currents, which was made by the voltage values read at the across the shunt resistors placed on the three phases of the motor in the driver, and sent via serial, is also of low effectiveness. The addition of an oscilloscope with four channels and differential probes for direct measurement of currents to such a test bench would allow for a more thorough evaluation of the motor and driver.

In the fifth chapter, the control and hardware used were validated by implementing a routine that simulated high-level control and thus replicated the angular profile of the knee, in the gait, on healthy subjects (from data available in the scientific literature). In this way, I was able to test the low-level control under dynamic conditions, in which the overall performance of the control developed on SimpleFOC library and the motor and driver combination was proven.

From the analysis done in this chapter, it emerges the clear improvement compared to the knee profile replicated by ALICE (figure 2.7) and the performance of the new designed actuator (figure 5.5 and 5.4), which are promising. Part of the goals set in the improvement analysis in the paragraph 2.4 have therefore been met.

In conclusion, this thesis work gave the research group the opportunity to deepen technical knowledge related to mechatronic design, and more specifically, on lower limbs exoskeletons. From the project, collaborations between different research laboratories, both within and outside the university, have emerged, which could enable them to address complex and long-term research projects.

Finally, as a personal note, this thesis work has allowed me to increase and concretize the engineering knowledge that the academic path gave me. I was also able to work in a research team and acquire a scientific approach, while maintaining an industrial approach, which I had gained in previous experiences.

6.2 Future developments

The following section will outline key areas for future development from this thesis work, presenting opportunities for enhancing control strategies, optimizing mechanical design, and exploring novel actuation systems.

Test the joint with harmonic drive and AS5600 sensor due to the lead time associated with the design and supply of the harmonic gearbox, the performance of the control was evaluated using a plastic planetary gearbox and a potentiometer as the position sensor. it is important then, to test the coupling in its final version with the harmonic gearbox and the AMS AS5600 as the post-reduction angular position sensor.

Design of a cycloidal drive in parallel with joint testing, it may be worthwhile, for the reasons described in the previous section, to proceed with the design of a cycloidal gearbox.

Continue the development of low-level control control performance obtained from the SimpleFOC library provides a good basis for continuing development of the entire exoskeleton. However, the possibility of improving this control, or moving to an implementation on VESC firmware, should not be excluded.

Begin the development of high-level control from the tests carried out in chapter five to validate the low-level control, it was found that it is necessary to design a high-level control that sends speed or probably better, torque references, as in many existing exoskeletons (for example, in [44, 83]). In addition, it may be necessary to integrate a trajectory planning [84, 44].

Functional design of the joint assembly a first draft of the joint assembly was shown in the third chapter. It turns out to be necessary to detail the component parts with a view to final production.

Design of the complete exoskeleton assembly the thesis work has focused on the single joint, which will then need to be replicated and specified for the joints it will support (hip, knee, or ankle). It will then be necessary to compose the complete exoskeleton assembly, which also includes the auxiliary components seen in 3.3 and the mechanical structure discussed in chapter 1 and 2.

Test high-level control with multiple joints connected in the final stage of exoskeleton development, it will be important to test the high-level control as a whole, namely, with all the joints that make up the exoskeleton and that will communicate with the high-level control via CAN bus communication.

A Mechanical drawings and datasheets

THIS section provides drawings and datasheets of the mechanical components used in the final joint design.

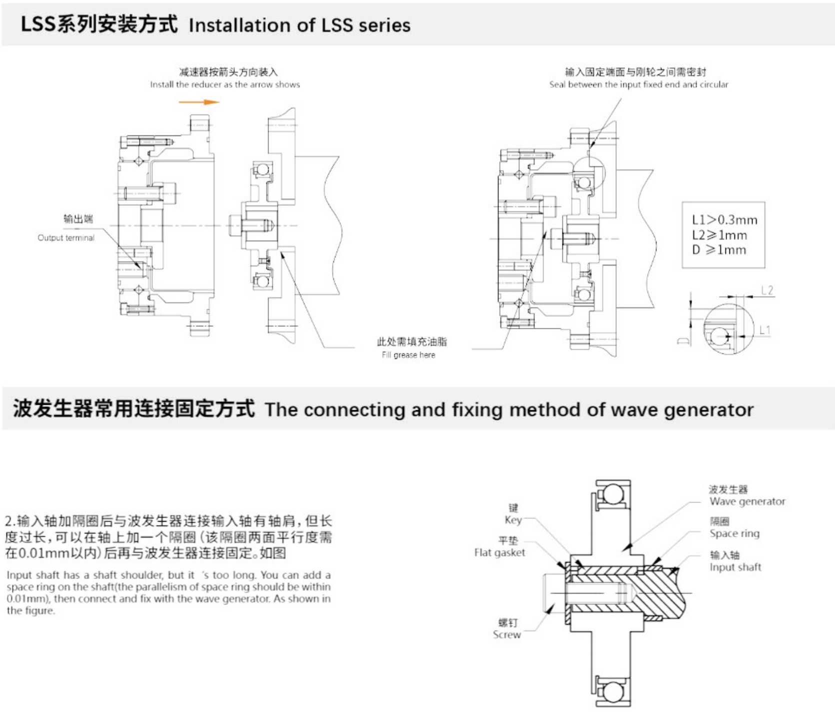
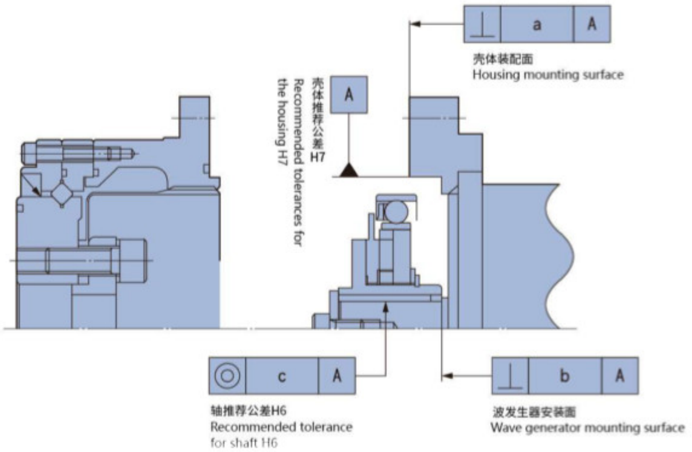


Figure A.1: Gearbox installation and instructions for correct coupling of the wave generator to the motor shaft [85].

LSS系列组装精度 Assembly accuracy of LSS series

在安装设计时，为充分发挥组合型所具备的优良性能，请确保使用如下图表精度。

To make sure LSS series play its excellent performance when assemble, please make sure to use the following accuracy.



组装壳体的推荐精度 Recommended accuracy of the assembled housing

单位unit：mm

型号 Model / 符号 Symbol	14	17	20	25	32
a	0.011	0.015	0.017	0.024	0.026
b	0.017	0.020	0.020	0.024	0.024
	(0.008)	(0.010)	(0.010)	(0.012)	(0.012)
c	0.030	0.034	0.044	0.047	0.050
	(0.016)	(0.018)	(0.019)	(0.022)	(0.022)

Figure A.2: Recommended geometric and positional tolerances for proper gearbox assembly and operation [85].

B

Electronic drawings and datasheets

THIS section provides drawings and datasheets of the electronic components used in the final joint design.

AS5600 – General Description

Benefits	Features
• Low-power consumption	• Automatic entry into low-power mode
• Easy setup	• Automatic magnet detection
• Small form factor	• SOIC-8 package
• Robust environmental tolerance	• Wide temperature range: -40°C to 125°C

Applications

The AS5600 is ideally suited for contactless potentiometers, contactless knobs, pedals, RC servos and other angular position measurement solutions.

Block Diagram

The functional blocks of this device are shown below:

Figure 2:
Functional Blocks of AS5600

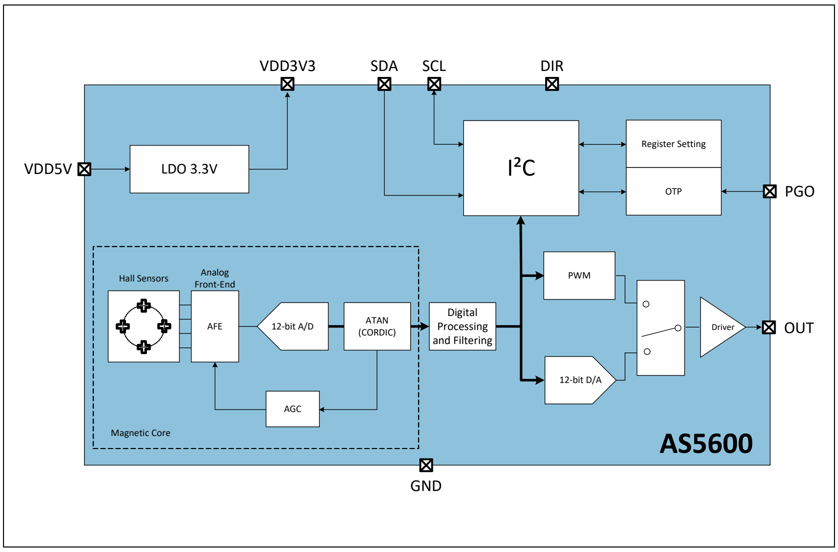


Figure B.1: Functional block of the AMS AS5600 architecture, with its main features, taken from the manufacturer's datasheet [86].

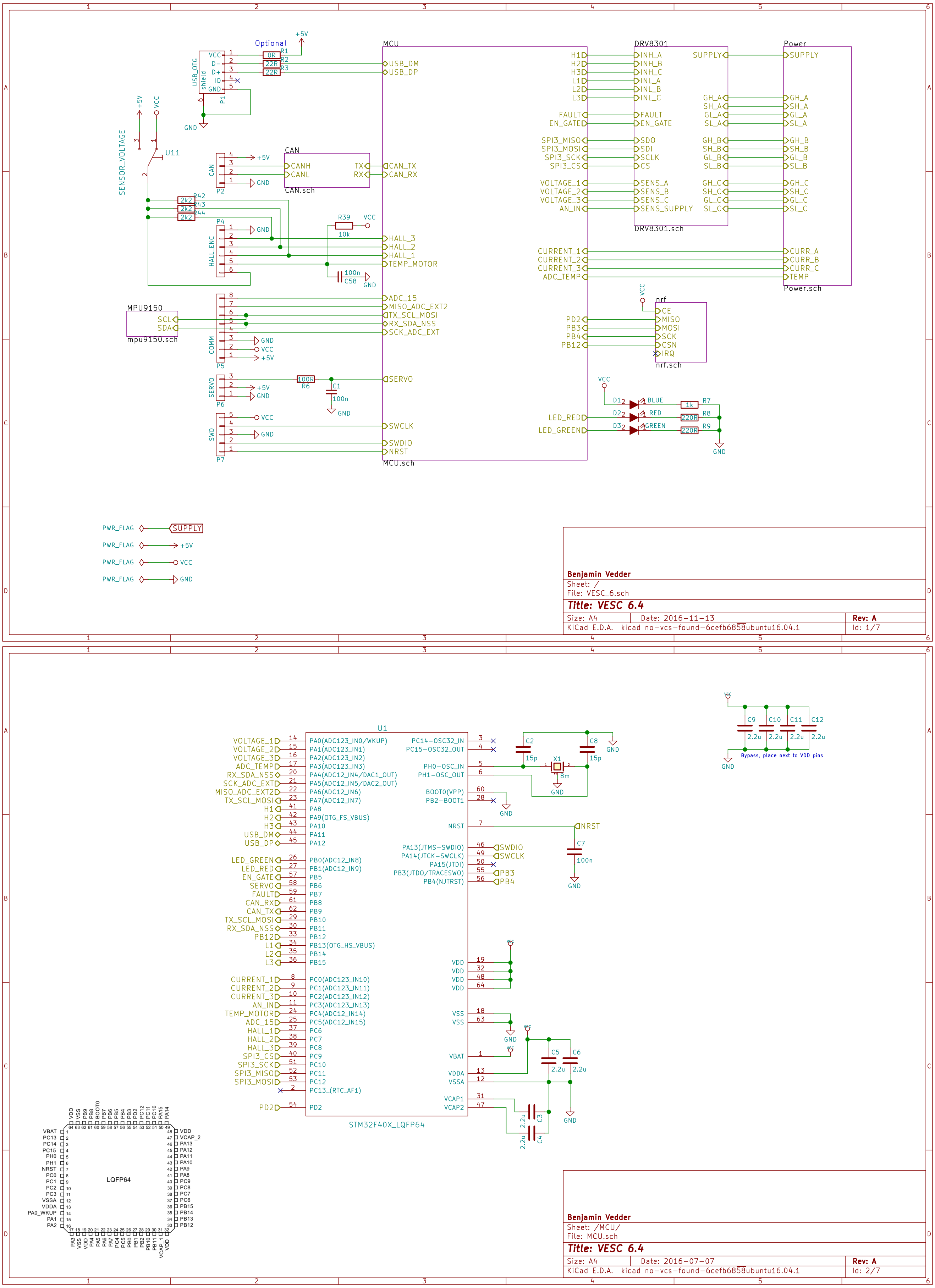
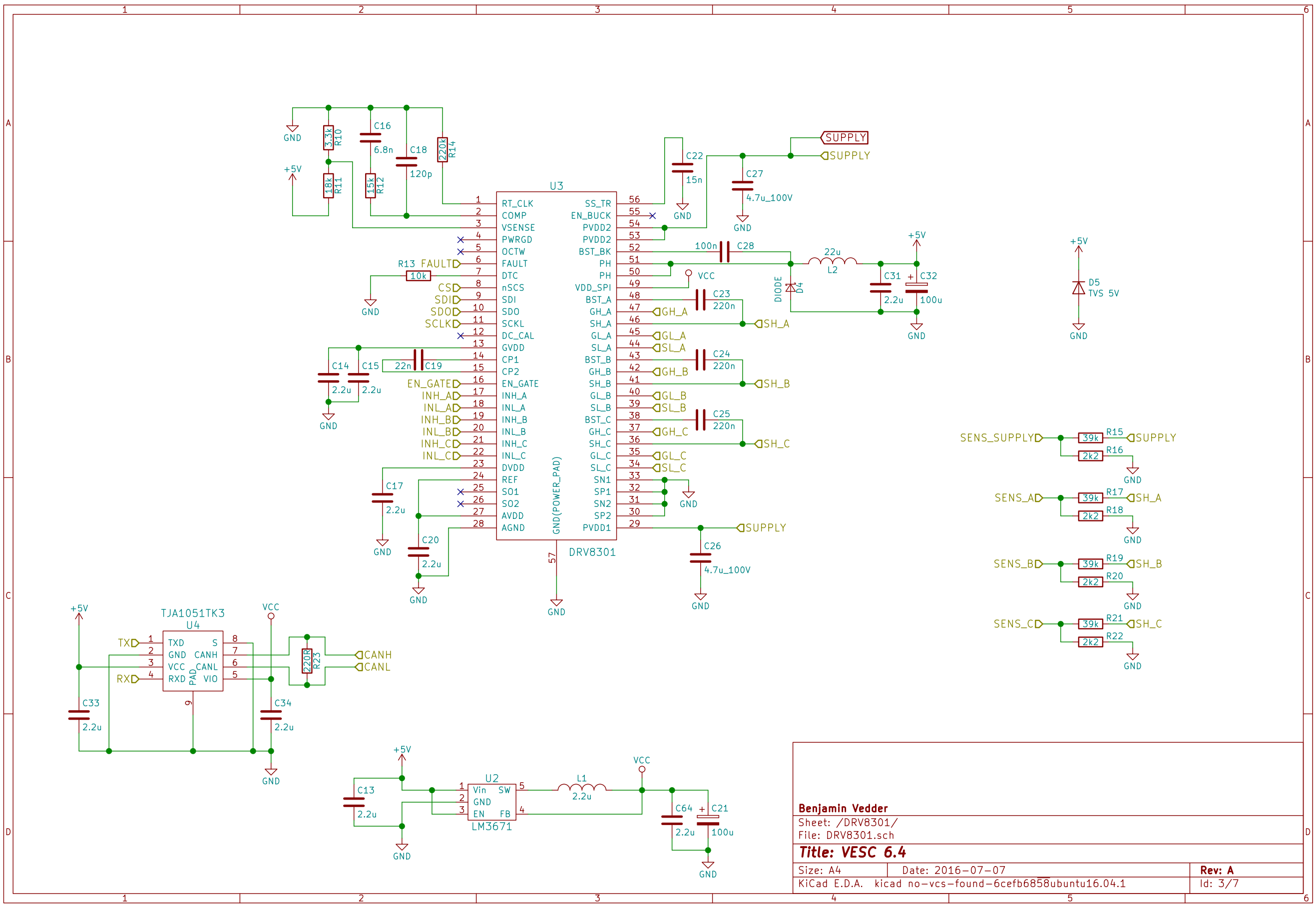


Figure B.2: Block diagram of the entire Flipsky FSESC6.7 driver and connections to the MCU [87].



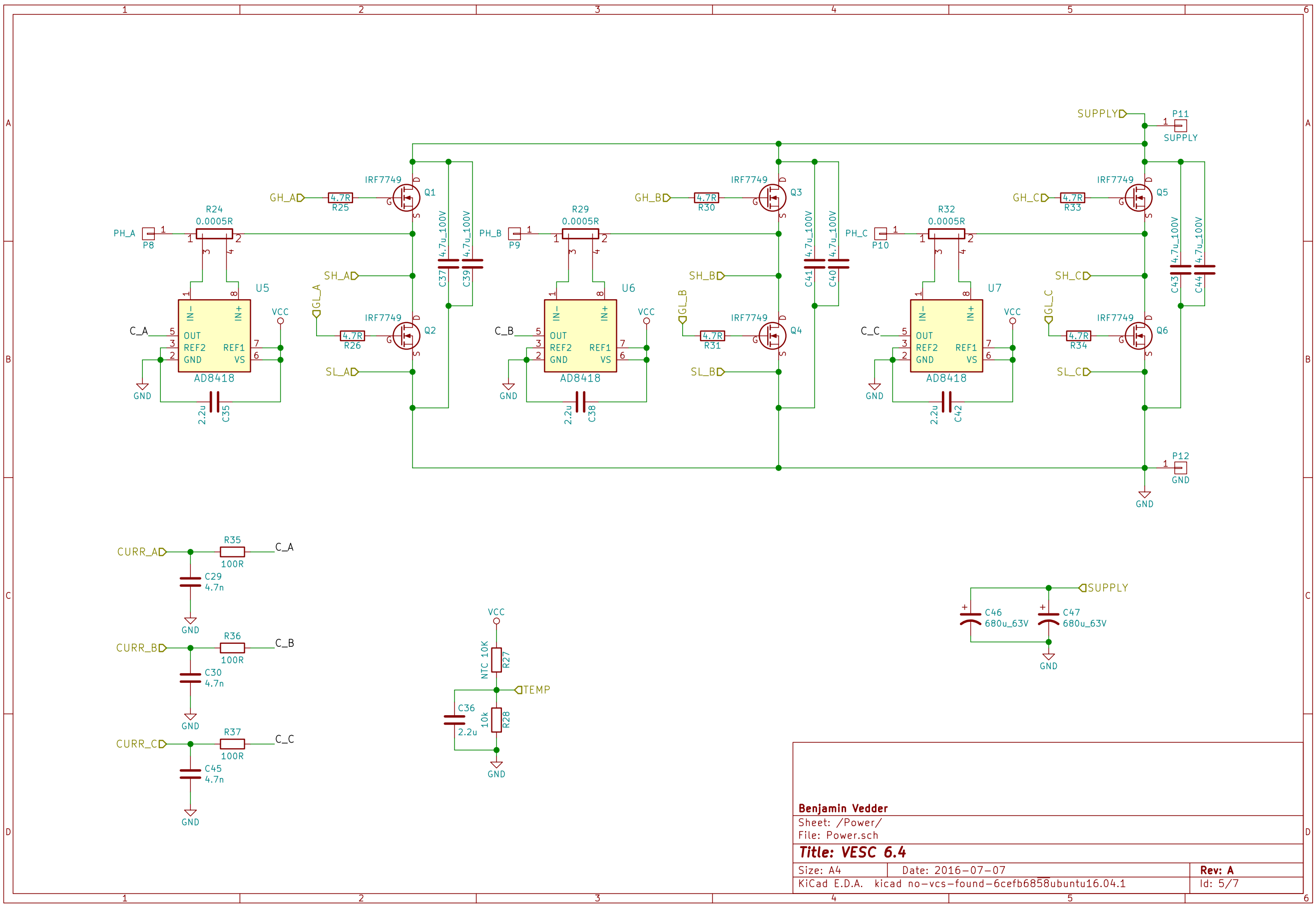


Figure B.3: Gate driver (DRV8301) and power stage with related MOSFETs (IRF7749) and operational amplifiers (AD8418) for in-line current sensing [87].

C Software description

THIS section reports some snippets of the firmware for low-level control of the realized robotic joint.

```
BLDCMotor motor = BLDCMotor(motorPolePairs);
BLDCDriver6PWM driver(H1, L1, H2, L2, H3, L3, EN_GATE);
InlineCurrentSense currentSense = InlineCurrentSense(0.0005, 200,
    CURRENT_1, CURRENT_2, CURRENT_3);

GenericSensor reduction_sensor =
    GenericSensor(readMySensorCallback, initMySensorCallback);

// encoder instance
HallSensor hall_sensor = HallSensor(HALL_1, HALL_2, HALL_3,
    motorPolePairs);

// Interrupt routine intialisation
// channel A and B callbacks
void doA() {hall_sensor.handleA();}
void doB() {hall_sensor.handleB();}
void doC() {hall_sensor.handleC();}

// instantiate the commander
Commander command = Commander(Serial);

//CAN Bus Communication Instance
CANDriver canD = CANDriver(CAN_RX, CAN_TX);
CANCommander canCommand = CANCommander(canD);

void doCommander(char* cmd) { command.motor(&motor, cmd); }
void doCommanderCAN(char* cmd) { canCommand.motor(&motor, cmd); }
```

Listing C.1: Initialization of variables for setting up the SimpleFOC library.

```
void setup()
{
  // initialize sensors hardware
  hall_sensor.init();
  hall_sensor.enableInterrupts(doA, doB, doC);
  reduction_sensor.init();

  driver.pwm_frequency = 25000;
  driver.voltage_power_supply = 26; // power supply voltage [V]
  driver.voltage_limit = 20; // Max DC voltage allowed
  driver.dead_zone = 0.03; // dead_zone [0,1] - default 0.02 - 2%
  driver.init();

  motor.phase_resistance = 0.053; // [Ohm]
  motor.KV_rating = 140;
  //motor.voltage_limit = 1.5; // [V]
  motor.current_limit = 200; // [Amps]
  motor.velocity_limit = 20; // [rad/s]

  // link the motor to the sensors and the driver to motor
  motor.linkVelSensor(&hall_sensor);
  motor.linkPosSensor(&reduction_sensor);
  motor.linkDriver(&driver);

  currentSense.init();      // current sensing init
  currentSense.skip_align = true;
  motor.linkCurrentSense(&currentSense);

  // set motion control loop to be used
  motor.controller = MotionControlType::velocity;
  motor.torque_controller = TorqueControlType::foc_current;
  motor.foc_modulation = FOCModulationType::SinePWM;
  ...
  // use monitoring with serial
  Serial.begin(115200);
  motor.useMonitoring(Serial);
  ...
  motor.init();
  motor.initFOC(3.14, Direction::CW);
}
```

Listing C.2: Excerpt of the void setup() for the low-level control firmware.

```
void loop()
{
  // FOC control algorithm
  motor.loopFOC();
  motor.move();
  ...
  // CAN Bus and serial Communication
  canCommand.runWithCAN();
  command.run();

  reduction_sensor.update();

  // simple and rudimental safety routine
  if(pot_value_deg > 180) motor.move(0);
  if(pot_value_deg < 0) motor.move(0);
 }
```

Listing C.3: Excerpt of the void loop() for the low-level control firmware.

```
float pot_value = 0;
float pot_value_deg = 0;
float pot_value_deg_prev = 0;
float pot_value_rad = 0;
float pot_value_rad_prev = 0;
float discard_tr = 0.5;
int discard_nu = 5;
int discard_av = discard_nu;

float fmap(float x, float in_min, float in_max, float out_min,
    float out_max)
{
  return (x - in_min) * (out_max - out_min) / (in_max - in_min) +
      out_min;
}

float readMySensorCallback(){
 // read my sensor
  analogReadResolution(12);
  pot_value = analogRead(ADC_15);
  pot_value_deg = fmap(pot_value,800.0,3265.0,-10.0,190);
  //pot_value_rad = fmap(pot_value,800.0,3265.0,-(_PI/10),(_PI -
      0.5));
  // return the angle value in radians in between 0 and 2PI

if ((abs(pot_value_deg - pot_value_deg_prev) > discard_tr) &&
    (discard_av > 0))
{
  pot_value_deg = pot_value_deg_prev;
  discard_av--;
  if (discard_av <= 0) discard_av = 0;
}
else
{
  pot_value_deg_prev = pot_value_deg;
  discard_av = discard_nu;
}

 return pot_value_deg;
}

void initMySensorCallback(){
  pinMode(ADC_15, INPUT);
}
```

Listing C.4: Implementation for the reading of the angular position sensor, used in the final position loop, by customizing the readMySensorCallback() and initMySensorCallback() methods.

```
float HallSensor::getVelocity(){

    Ts = pulse_diff * 1e-6;
    if (Ts != 0) vel_cal = direction * (_2PI / cpr) / Ts;

    if(pulse_diff > 14000)
    {
      vel_cal = 0;
    }

    if ((abs(vel_cal - vel_cal_old) > vel_cal_dis_tr) &&
        (vel_cal_dis_av > 0))
    {
      vel_cal = vel_cal_old;
      vel_cal_dis_av--;
      if (vel_cal_dis_av <= 0) vel_cal_dis_av = 0;
    }
    else
    {
      vel_cal_old = vel_cal;
      vel_cal_dis_av = vel_cal_dis_nu;
    }

    return vel_cal;


    // Original implementation of the library on the following
        commented lines
/*
  if (pulse_diff == 0 || ((long)(_micros() - pulse_timestamp) >
      pulse_diff) ) { // last velocity isn't accurate if too old
    return 0;
  } else {
    return direction * (_2PI / (float)cpr) / (pulse_diff /
        1000000.0f);
  }
*/
    // New implementation on the latest version of the library
        that fix the problem
/*
  long last_pulse_diff = pulse_diff;
  if (last_pulse_diff == 0 || ((long)(_micros() - pulse_timestamp)
      > last_pulse_diff) ) { // last velocity isn't accurate if too
      old
    return 0;
  } else {
```

```
    float vel = direction * (_2PI / (float)cpr) / (last_pulse_diff
        / 1000000.0f);
    // quick fix
        https://github.com/simplefoc/Arduino-FOC/issues/192
    if(vel < -velocity_max || vel > velocity_max)  vel = 0.0f;
        //if velocity is out of range then make it zero
    return vel;
  }
*/

}
```

Listing C.5: Comparison of different versions of the getVelocity() method in the case of using hall sensors. In succession, the original implementation (v2.2.2), the modification made in this thesis and finally, the implementation of the latest version of the library (v2.3.0) is given

Acronyms

DC direct current

In direct current (DC), the electric charge (current) only flows in one direction. Electric charge in alternating current (AC), on the other hand, changes direction periodically. The voltage in AC circuits also periodically reverses because the current changes direction.

BDC brushed DC motor

An electromechanical device converting electrical energy into mechanical motion. It consists of a rotor with windings and a commutator, and a stator with permanent magnets or electromagnets. The interaction between the rotor and stator's magnetic fields generates rotational motion. BDC motors are cost-effective and offer high starting torque, but require maintenance due to brushes and commutator.

BLDC brushless DC motor

Is an electric motor that converts electrical energy into mechanical motion, commutated electronically instead of by brushes like in conventional DC motors. BLDC motors are more popular than the conventional DC motors nowadays, but the development of these type of motors has only been possible since the 1960s when semiconductor electronics were developed. It consists of a rotor with permanent magnets and a stator with windings. BLDC motors offer advantages such as higher efficiency, lower maintenance, and improved reliability. They find applications in robotics, electric vehicles, drones, and industrial machinery.

SCI spinal cord Injury

Damage to any part of the spinal cord or nerves at the end of the spinal canal. Often causes permanent changes in strength, sensation and other body functions below the site of the injury.

SoC system on chip

an integrated circuit that integrates most or all components of a computer or other electronic system. These components almost always include on-chip CPU, memory interfaces, input/output devices, input/output interfaces, and secondary storage interfaces, often alongside other components such as radio modems and a GPU) all on a single substrate or microchip. SoCs may contain digital, and also analog, mixed-signal, and often radio frequency signal processing functions.

UAV unmanned aerial vehicles

Commonly known as a drone, is an aircraft without any human pilot, crew, or passengers on board.

BEMF back electro-motive force

Voltage that occurs in electric motors where there is relative motion between the armature and the magnetic field produced by the motor's field coils or permanent magnet field.

WR wearable robots

Wearable robots are advanced human symbiotic robotic systems characterized by suitable shape, kinematic, and weight factors to be worn on the human body with the function of either augmenting and assisting (exoskeletons) or restoring human limb function (prosthetic robots).

EXO exoskeleton

Robotic exoskeletons are mechanically made, taking a user's anatomy into consideration, to improve mobility and endurance. They involve the application of robotics and bio-mechatronics. It is designed to assist humans by enhancing, reinforcing, or restoring, depending on the circumstances, an individual's physical performance. Exoskeletons can also work to reduce the energy it takes to move joints, making repetitive tasks easier, and also work to improve human movement in cases of mobility loss.

HMI human machine interface

Is defined as a feature or component of a certain device or software application that enables humans to engage and interact with machines. Some examples of common Human Machine Interface devices that we encounter in our daily lives include touchscreens and keyboards.

IMU inertial measurement unit

It is a sensor device that precisely measures an object's acceleration and angular rate using accelerometers and gyroscopes. IMUs are widely utilized in robotics, navigation systems, and other applications to track and analyze motion, enabling accurate determination of position, orientation, and velocity.

ECU electronic controller unit

Is an embedded system in automotive electronics that controls one or more of the electrical systems or subsystems in a car or other motor vehicle.

SPI serial pheriperal interface

Serial communication interface used for data transmission between electronic devices. It involves a synchronous clock line and one or more data lines for bidirectional communication. SPI is often used to connect MCUs, sensors, displays, and other peripheral devices to main integrated circuits.

I2C inter integrated circuit

Serial communication protocol used for connecting multiple integrated circuits in a system. It enables communication between a master device (typically a MCU or a CPU) and multiple slave devices. I2C utilizes a two-wire interface consisting of a clock line (SCL) and a data line (SDA). It allows for bidirectional data transfer and supports multiple devices sharing the same bus. Each slave device on the I2C bus has a unique address, allowing the master to address and communicate with specific slaves

ROS robot operating system

Is an open-source framework for developing robot applications. It provides software libraries and tools that enable communication and coordination between different components of a robotic system. With a modular design and support for multiple programming languages, ROS simplifies development and promotes code reusability. It is widely used in academia and industry for building and integrating software components in robotics applications.

ML machine learning

Field that teaches computers to learn and make predictions or decisions from data, without being explicitly programmed. It involves creating algorithms and models that enable computers to recognize patterns and make informed predictions based on the information they have learned. ML is used in various applications like image recognition, language processing, and recommendation systems.

I/O input/output

It refers to the communication between a computer or a system and the outside world, including devices such as keyboards, mice, monitors, printers, disks, and network connections. Input refers to data or signals received by the computer or system from external sources, while output refers to data or signals sent by the computer or system to external devices. Input/Output operations are essential for data exchange, user interaction, and the functioning of computer systems.

GPIO general purpose I/O

It stands for General Purpose Input/Output and encompasses a set of pins or interfaces available on MCUs or single-board computers. These pins can be configured to function either as inputs or outputs, providing flexibility for connecting and controlling external devices. GPIO pins allow for the reception of digital signals from external sources (input) or the transmission of digital signals to control external components (output).

CPU central processing unit

It is the primary component of a computer responsible for executing instructions and performing calculations. Often referred to as the "brain" of the computer, the CPU interprets and carries out instructions, controls data flow, and manages overall system functioning.

GPU graphic processing unit

A specialized electronic circuit designed for high-speed manipulation and rendering of images, videos, and animations. GPUs excel in graphics-intensive applications and parallel processing tasks, offering efficient computation and accelerated visual computing.

PID proportional integrative derivative

Control algorithm used in engineering and automation systems to regulate and stabilize processes. The PID controller adjusts the system's output based on the error between the desired setpoint and the current value, using proportional, integral, and derivative components. It helps achieve stability and accuracy in controlling various applications such as temperature, robotics, and industrial automation.

CAN controller area network

Widely used communication protocol in automotive and industrial applications. It enables multiple devices to communicate with each other over a shared network, providing reliable and real-time data exchange. CAN bus is known for its robustness, fault tolerance, and scalability, making it suitable for applications such as in-vehicle networks and industrial automation.

FOC field oriented control

Control technique used in electric motor drives for precise and efficient control of motor speed and torque. By aligning the motor's magnetic field with a reference frame, FOC enables independent control of flux and current (torque) components, resulting in improved performance and energy efficiency. It is widely used in applications such as electric vehicles, industrial machinery, and robotics.

PWM pulse width modulation

Technique used to control the average power delivered to a load by adjusting the width of electrical pulses in a digital signal. PWM is widely used for applications like motor speed control, LED dimming, and power regulation, providing efficient and precise control over voltage or power output.

MCU microcontroller unit

Is a compact integrated circuit that combines a microprocessor core, memory, and peripherals on a single chip. MCUs are commonly used in embedded systems for controlling and monitoring functions in various applications, such as consumer electronics, industrial automation, and automotive systems. They are optimized for low power consumption, compact size, and real-time control.

IC integrated circuit

A tiny electronic device that integrates multiple components onto a single semiconductor material, enabling miniaturization and improved performance of electronic systems. ICs are the building blocks of modern electronics, used in various applications like computers, smartphones, and automotive systems. Their invention revolutionized the industry by enhancing reliability, affordability, and efficiency of electronic devices.

HAL hardware abstraction layer

Software interface that provides a standardized and simplified way for software components, particularly in embedded systems like MCUs, to interact with hardware devices. The HAL abstracts the hardware-specific details, enabling software to be written in a hardware-independent manner. It simplifies development, enhances portability, and promotes code reusability across different embedded systems. The HAL acts as an intermediary between the software and hardware, providing a unified and consistent interface for software components to access and control the underlying hardware.

FDM fused deposition modeling

Widely adopted 3D printing technology that utilizes a thermoplastic filament. The filament is melted and extruded through a nozzle, layer by layer, to create a three-dimensional object. FDM is known for its versatility, affordability, and ease of use, making it a popular choice in industries such as prototyping, product development, and small-scale manufacturing.

DIY do it yourself

Is a practise that involves creating, modifying, or repairing things independently without relying on professional assistance. It encompasses a wide range of activities that allow individuals to showcase their skills, personalize items, and experience a sense of achievement. DIY projects have gained popularity across various domains and are facilitated by online resources, tutorials, and accessible tools.

Bibliography

[1] Robert Bogue. “Exoskeletons–a review of industrial applications”. In: *Industrial Robot: An International Journal* (2018) (cit. on p. 1).

[2] Simone Cosimi. “Esselunga sperimenta un esoscheletro per i suoi magazzinieri”. In: *Wired Italia* (2022) (cit. on p. 1).

[3] Tingfang Yan, Marco Cempini, Calogero Maria Oddo, and Nicola Vitiello. “Review of assistive strategies in powered lower-limb orthoses and exoskeletons”. In: *Robotics and Autonomous Systems* 64 (2015), pp. 120–136 (cit. on p. 1).

[4] Di Shi, Wuxiang Zhang, Wei Zhang, and Xilun Ding. “A review on lower limb rehabilitation exoskeleton robots”. In: *Chinese Journal of Mechanical Engineering* 32.1 (2019), pp. 1–11 (cit. on p. 2).

[5] M Patrice Lindsay, Bo Norrving, Ralph L Sacco, et al. *World Stroke Organization (WSO): global stroke fact sheet 2019*. 2019 (cit. on p. 2).

[6] Slavka Viteckova, Patrik Kutilek, and Marcel Jirina. “Wearable lower limb robotics: A review”. In: *Biocybernetics and biomedical engineering* 33.2 (2013), pp. 96–105 (cit. on p. 2).

[7] World Health Organization: *Spinal cord injury*. [Online; accessed 27. Feb. 2022] (cit. on p. 2).

[8] Shiqian Wang, Letian Wang, Cory Meijneke, et al. “Design and control of the MINDWALKER exoskeleton”. In: *IEEE transactions on neural systems and rehabilitation engineering* 23.2 (2014), pp. 277–286 (cit. on p. 2).

[9] *APPLICATION CASES – Comau Mate*. [Online; accessed 26. Feb. 2022] (cit. on p. 3).

[10] *A History of Our Past Exoskeleton Products - Ekso Bionics*. [Online; accessed 26. Feb. 2022] (cit. on p. 3).

[11] Zhijun Li, Yuxia Yuan, Ling Luo, et al. “Hybrid brain/muscle signals powered wearable walking exoskeleton enhancing motor ability in climbing stairs activity”. In: *IEEE Transactions on Medical Robotics and Bionics* 1.4 (2019), pp. 218–227 (cit. on p. 4).

[12] Weiguang Huo, Samer Mohammed, Juan C Moreno, and Yacine Amirat. “Lower limb wearable robots for assistance and rehabilitation: A state of the art”. In: *IEEE systems Journal* 10.3 (2014), pp. 1068–1081 (cit. on p. 4).

[13] Brock Laschowski, William McNally, Alexander Wong, and John McPhee. “ExoNet database: Wearable camera images of human locomotion environments”. In: *Frontiers in Robotics and AI* 7 (2020), p. 562061 (cit. on p. 4).

[14] Shiyin Qiu, Wei Guo, Darwin Caldwell, and Fei Chen. "Exoskeleton online learning and estimation of human walking intention based on dynamical movement primitives". In: *IEEE Transactions on Cognitive and Developmental Systems* 13.1 (2020), pp. 67–79 (cit. on p. 4).

[15] J.G. Betts, P. Desaix, E.W. Johnson, et al. *Anatomy & Physiology*. Open Textbook Library. OpenStax College, Rice University, 2013 (cit. on pp. 4, 5).

[16] *Project MARCH: behind the technology of robotic exoskeletons*. [Online; accessed 26. Feb. 2022] (cit. on p. 5).

[17] Michael W Whittle. *Gait analysis: an introduction*. Butterworth-Heinemann, 2014 (cit. on p. 6).

[18] Henry G Chambers and David H Sutherland. "A practical guide to gait analysis". In: *JAAOS-Journal of the American Academy of Orthopaedic Surgeons* 10.3 (2002), pp. 222–231 (cit. on p. 6).

[19] Can Tunca, Nezihe Pehlivan, Nağme Ak, et al. "Inertial sensor-based robust gait analysis in non-hospital settings for neurological disorders". In: *Sensors* 17.4 (2017), p. 825 (cit. on p. 7).

[20] Sajid Iqbal, Xizhe Zang, Yanhe Zhu, and Z Jie. "Nonlinear time-series analysis of human gaits in aging and Parkinson's disease". In: *2015 international conference on mechanics and control engineering (MCE 2015)*. 2015 (cit. on p. 7).

[21] Javier A de la Tejera, Rogelio Bustamante-Bello, Ricardo A Ramirez-Mendoza, and Javier Izquierdo-Reyes. "Systematic review of exoskeletons towards a general categorization model proposal". In: *Applied Sciences* 11.1 (2020), p. 76 (cit. on pp. 8–10).

[22] *Project MARCH*. [Online; accessed 26. Feb. 2022] (cit. on pp. 11, 41, 50, 51).

[23] Max Felser. "The fieldbus standards: History and structures". In: *Technology Leadership Day* (2002) (cit. on p. 12).

[24] J-P Thomesse. "Fieldbus technology in industrial automation". In: *Proceedings of the IEEE* 93.6 (2005), pp. 1073–1101 (cit. on p. 12).

[25] Omid Dehzangi, Mojtaba Taherisadr, and Raghvendar ChangalVala. "IMU-based gait recognition using convolutional neural networks and multi-sensor fusion". In: *Sensors* 17.12 (2017), p. 2735 (cit. on p. 12).

[26] P Chinmilli, Sangram Redkar, Wenlong Zhang, and Tom Sugar. "A review on wearable inertial tracking based human gait analysis and control strategies of lower-limb exoskeletons". In: *Int. Robot. Autom. J* 3.7 (2017), p. 00080 (cit. on p. 12).

[27] Ana Cecilia Villa-Parra, Denis Delisle-Rodriguez, Jessica Souza Lima, Anselmo Frizera-Neto, and Teodiano Bastos. "Knee impedance modulation to control an active orthosis using insole sensors". In: *Sensors* 17.12 (2017), p. 2751 (cit. on p. 12).

[28] Sang-Hoon Kim. *Electric motor control: DC, AC, and BLDC motors*. Elsevier, 2017 (cit. on pp. 13–16).

[29] David G Dorrell, Min-Fu Hsieh, and Andrew M Knight. "Alternative rotor designs for high performance brushless permanent magnet machines for hybrid electric vehicles". In: *IEEE Transactions on Magnetics* 48.2 (2012), pp. 835–838 (cit. on p. 15).

[30] Guan Qiao, Geng Liu, Zhenghong Shi, et al. "A review of electromechanical actuators for More/All Electric aircraft systems". In: *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science* 232.22 (2018), pp. 4128–4151 (cit. on p. 16).

[31] Stefano Tornincasa Emilio Chirone. *Disegno tecnico industriale*. Il capitello, 2006 (cit. on p. 17).

[32] Yoshio TERAUCHI, Kazuteru NAGAMURA, and Kiyotaka IKEJO. "Study on friction loss of internal gear drives: Intluence of pinion surface finishing, gear speed and torque". In: *JSME international journal. Ser. 3, Vibration, control engineering, engineering for industry* 34.1 (1991), pp. 106–113 (cit. on p. 17).

[33] *Overview of gear types - tec-science*. [Online; accessed 16. Jun. 2022] (cit. on p. 17).

[34] Pablo López García, Stein Crispel, Elias Saerens, Tom Verstraten, and Dirk Lefeber. "Compact gearboxes for modern robotics: A review". In: *Frontiers in Robotics and AI* 7 (2020), p. 103 (cit. on pp. 17, 34).

[35] Elias Saerens, Stein Crispel, Pablo Lopez Garcia, et al. "Scaling laws for robotic transmissions". In: *Mechanism and Machine Theory* 140 (2019), pp. 601–621 (cit. on pp. 17, 18).

[36] JE Huber, NA Fleck, and MF Ashby. "The selection of mechanical actuators based on performance indices". In: *Proceedings of the Royal Society of London. Series A: Mathematical, physical and engineering sciences* 453.1965 (1997), pp. 2185–2205 (cit. on p. 17).

[37] Jonathon W Sensinger and James H Lipsey. "Cycloid vs. harmonic drives for use in high ratio, single stage robotic transmissions". In: *2012 IEEE International Conference on Robotics and Automation*. IEEE. 2012, pp. 4130–4135 (cit. on pp. 17, 34, 88, 89).

[38] *INDI by Jesús Tamez-Duque*. [Online; accessed 1. Jul. 2022] (cit. on p. 21).

[39] Volker Bartenbach, Marcel Gort, and Robert Riener. "Concept and design of a modular lower limb exoskeleton". In: *2016 6th IEEE International Conference on Biomedical Robotics and Biomechatronics (BioRob)*. IEEE. 2016, pp. 649–654 (cit. on pp. 28, 35).

[40] Rafhael M Andrade and Paolo Bonato. "The role played by mass, friction, and inertia on the driving torques of lower-limb gait training exoskeletons". In: *IEEE Transactions on Medical Robotics and Bionics* 3.1 (2021), pp. 125–136 (cit. on pp. 35, 88, 89).

[41] Manuel Cardona, Cecilia García Cena, Juan Andrés Martín, and Estrella Rausell. "Actuation System Selection of ALICE Exoskeleton Robot Based on Dynamic Simulation". In: *2020 IEEE ANDESCON*. IEEE. 2020, pp. 1–6 (cit. on p. 35).

[42] Magdo Bortole, Anusha Venkatakrishnan, Fangshi Zhu, et al. "The H2 robotic exoskeleton for gait rehabilitation after stroke: early findings from a clinical study". In: *Journal of neuroengineering and rehabilitation* 12.1 (2015), pp. 1–14 (cit. on p. 35).

[43] Matteo Laffranchi, Stefano D'Angella, Christian Vassallo, et al. "User-Centered Design and Development of the Modular TWIN Lower Limb Exoskeleton". In: *Frontiers in neurorobotics* 15 (2021) (cit. on p. 35).

[44] C Meijneke, G van Oort, V Sluiter, et al. "Symbitron Exoskeleton: Design, control, and evaluation of a modular exoskeleton for incomplete and complete spinal cord injured individuals". In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 29 (2021), pp. 330–339 (cit. on pp. 35, 37, 90).

[45] Peter D Neuhaus, Jerryll H Noorden, Travis J Craig, et al. "Design and evaluation of Mina: A robotic orthosis for paraplegics". In: *2011 IEEE international conference on rehabilitation robotics*. IEEE. 2011, pp. 1–8 (cit. on p. 35).

[46] Miguel Sanchez-Manchola, Daniel Gómez-Vargas, Diego Casas-Bocanegra, Marcela Munera, and Carlos A Cifuentes. "Development of a robotic lower-limb exoskeleton for gait rehabilitation: AGoRA exoskeleton". In: *2018 IEEE ANDESCON*. IEEE. 2018, pp. 1–6 (cit. on p. 35).

[47] Wei Yang, Can-jun Yang, and Qian-xiao Wei. "Design of an anthropomorphic lower extremity exoskeleton with compatible joints". In: *2014 IEEE International Conference on Robotics and Biomimetics (ROBIO 2014)*. IEEE. 2014, pp. 1374–1379 (cit. on p. 35).

[48] David Eguren, Manuel Cestari, Trieu Phat Luu, et al. "Design of a customizable, modular pediatric exoskeleton for rehabilitation and mobility". In: *2019 IEEE international conference on systems, man and cybernetics (SMC)*. IEEE. 2019, pp. 2411–2416 (cit. on p. 35).

[49] Hyunjin Choi, Byeonghun Na, Solrim Kim, et al. "Angel-suit: A modularized lower-limb wearable robot for assistance of people with partially impaired walking ability". In: *2019 Wearable Robotics Association Conference (WearRAcon)*. IEEE. 2019, pp. 51–56 (cit. on p. 35).

[50] *Project MARCH - Joint design*. [Online; accessed 21. Jul. 2022]. July 2022 (cit. on p. 37).

[51] Dwi Mutiara Harfina, Zaini Zaini, and Wisnu Joko Wulung. "Disinfectant spraying system with quadcopter type unmanned aerial vehicle (UAV) technology as an effort to break the chain of the COVID-19 virus". In: *Journal of Robotics and Control (JRC)* 2.6 (2021), pp. 502–507 (cit. on p. 37).

[52] Darren Lance Gabriel, Johan Meyer, and Francois Du Plessis. "Brushless DC motor characterisation and selection for a fixed wing UAV". In: *IEEE Africon'11*. IEEE. 2011, pp. 1–6 (cit. on p. 37).

[53] Kellen D Carey, Nathan Zimmerman, and Cristinel Ababei. "Hybrid field oriented and direct torque control for sensorless BLDC motors used in aerial drones". In: *IET Power Electronics* 12.3 (2019), pp. 438–449 (cit. on p. 37).

[54] Pei Pei, Zhongcai Pei, Zhengqiang Shi, Zhiyong Tang, Yang Li, et al. "Sensorless control for joint drive unit of lower extremity exoskeleton with cascade feedback observer". In: *Mathematical Problems in Engineering* 2018 (2018) (cit. on p. 37).

[55] *Customisation of New ODrive Generation*. [Online; accessed 30. Apr. 2023]. June 2022 (cit. on p. 41).

[56] Slávka Net'uková, Martin Bejtic, Christiane Malá, et al. “Lower Limb Exoskeleton Sensors: State-of-the-Art”. In: *Sensors* 22.23 (2022), p. 9091 (cit. on p. 43).

[57] Aaron J Young and Daniel P Ferris. “State of the art and future directions for lower limb robotic exoskeletons”. In: *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 25.2 (2016), pp. 171–182 (cit. on p. 43).

[58] Patrick Mesmer, Michael Neubauer, Armin Lechler, and Alexander Verl. “Challenges of linearization-based control of industrial robots with cycloidal drives”. In: *2021 IEEE International Conference on Mechatronics (ICM)*. IEEE. 2021, pp. 1–8 (cit. on p. 43).

[59] William J Fleming. “Overview of automotive sensors”. In: *IEEE sensors journal* 1.4 (2001), pp. 296–308 (cit. on p. 43).

[60] AS Anil Kumar, Boby George, and Subhas Chandra Mukhopadhyay. “Technologies and applications of angle sensors: A review”. In: *IEEE Sensors Journal* 21.6 (2020), pp. 7195–7206 (cit. on p. 43).

[61] Sakari Junnila, Risto Pajula, Mickey Shroff, et al. “Design of High-Performance CAN Driver Architecture for Embedded Linux”. In: *13th international CAN Conference Part*. Vol. 5, pp. 1–9 (cit. on p. 53).

[62] Enrico Pozzobon, Nils Weiss, Sebastian Renner, and Rudolf Hackenberg. “A survey on media access solutions for can penetration testing”. In: *ACM Computer Science in Cars Symposium, CSCS*. Vol. 18. 2018 (cit. on p. 53).

[63] Marco Matteo Bassa. “Development of the communication system for a lower limb human exoskeleton using the ros middleware”. MA thesis. Università degli studi di padova, 2015 (cit. on p. 53).

[64] *Motor Control Blockset*. [Online; accessed 8. Apr. 2023]. Apr. 2023 (cit. on p. 54).

[65] *ST-MC-SUITE - STMicroelectronics*. [Online; accessed 8. Apr. 2023]. Apr. 2023 (cit. on p. 54).

[66] *VESC Project*. [Online; accessed 8. Apr. 2023]. Apr. 2023 (cit. on p. 54).

[67] *Arduino Simple Field Oriented Control project*. [Online; accessed 8. Apr. 2023]. Apr. 2023 (cit. on p. 54).

[68] Antun Skuric, Hasan Sinan Bank, Richard Unger, Owen Williams, and David González-Reyes. “SimpleFOC: A Field Oriented Control (FOC) Library for Controlling Brushless Direct Current (BLDC) and Stepper Motors”. In: *Journal of Open Source Software* 7.74 (2022), p. 4232 (cit. on pp. 58, 59).

[69] *Stackable planetary gearbox*. [Online; accessed 26. Dec. 2022]. Dec. 2022 (cit. on p. 61).

[70] Pooya Alaeinovin and Juri Jatskevich. “Hall-sensor signals filtering for improved operation of brushless DC motors”. In: *2011 IEEE International Symposium on Industrial Electronics*. IEEE. 2011, pp. 613–618 (cit. on p. 63).

[71] Marko Nerat and Damir Vrančić. “A novel fast-filtering method for rotational speed of the BLDC motor drive applied to valve actuator”. In: *IEEE/ASME transactions on mechatronics* 21.3 (2015), pp. 1479–1486 (cit. on p. 63).

[72] *CAN BUS interface - SimpleFOC standard identifiers? - SimpleFOC Community*. [Online; accessed 26. Dec. 2022]. Feb. 2021 (cit. on p. 63).

[73] David A. Winter. “Appendix A: Kinematic, Kinetic, and Energy Data”. In: *Biomechanics and Motor Control of Human Movement*. Chichester, England, UK: John Wiley & Sons, Ltd, Sept. 2009, pp. 296–360 (cit. on pp. 79, 81).

[74] David A Winter. *Biomechanics and motor control of human movement*. John Wiley & Sons, 2009 (cit. on pp. 79, 81).

[75] Renato Contini. “Body segment parameters, Part II”. In: *Artificial limbs* 16.1 (1972), pp. 1–19 (cit. on p. 79).

[76] Contributors to Wikimedia projects. *Average absolute deviation - Wikipedia*. [Online; accessed 6. May 2023]. Apr. 2023 (cit. on p. 82).

[77] Jungsu Choi, Kyeong-Won Park, Jeongsu Park, et al. “The history and future of the walkON suit: A powered exoskeleton for people with disabilities”. In: *IEEE Industrial Electronics Magazine* 16.4 (2021), pp. 16–28 (cit. on pp. 88, 89).

[78] DDmon. *ANGEL ROBOTICS*. [Online; accessed 3. Jun. 2023]. June 2023 (cit. on p. 88).

[79] Luis I Minchala, Anthony J Velasco, Jonathan M Blandin, Fabian Astudillo-Salinas, and Andres Vazquez-Rodas. “Low cost lower limb exoskeleton for assisting gait rehabilitation: design and evaluation”. In: *Proceedings of the 2019 3rd International Conference on Automation, Control and Robots*. 2019, pp. 55–60 (cit. on pp. 88, 89).

[80] James Bruton. *3D Printed Cycloidal Drive V2 - Much Better!* [Online; accessed 26. Dec. 2022]. Mar. 2021 (cit. on p. 88).

[81] James Bruton. *Testing: Cycloidal vs Harmonic Drive 3D Printed Reducers*. [Online; accessed 14. Jul. 2022]. Apr. 2021 (cit. on p. 88).

[82] simplefoc. *HallSensor.cpp volatile access bug*. [Online; accessed 4. Jun. 2023]. June 2023 (cit. on p. 89).

[83] Tomislav Bacek, Marta Moltedo, Kevin Langlois, et al. “BioMot exoskeleton—Towards a smart wearable robot for symbiotic human-robot interaction”. In: *2017 International Conference on Rehabilitation Robotics (ICORR)*. IEEE. 2017, pp. 1666–1671 (cit. on p. 90).

[84] Mojtaba Sharifi, Javad K Mehr, Vivian K Mushahwar, and Mahdi Tavakoli. “Autonomous locomotion trajectory shaping and nonlinear control for lower limb exoskeletons”. In: *IEEE/ASME Transactions on Mechatronics* 27.2 (2022), pp. 645–655 (cit. on p. 90).

[85] *Laifual-Zhejiang Laifual Drive Co., Ltd.* [Online; accessed 11. Jun. 2023]. June 2023 (cit. on pp. 93, 94).

[86] *AS5600 - ams*. [Online; accessed 11. Jun. 2023]. June 2023 (cit. on p. 95).

[87] *Documentation | VESC Project*. [Online; accessed 11. Jun. 2023]. June 2023 (cit. on pp. 96, 97).

Colophon

This thesis was typeset with LaTeX. It uses the *Clean Thesis* style developed by Ricardo Langner. The design of the *Clean Thesis* style is inspired by user guide documents from Apple Inc.

`http://cleanthesis.der-ric.de/`.

Inoltre per chi è agli inizi e volesse cimentarsi nell'arte della scrittura con LaTeX si consiglia la guida di Lorenzo Pantieri. Scarica *L'arte di scrivere con LaTeX* al link:

`http://www.lorenzopantieri.net/LaTeX.html`.