

Design and validation of a five degrees of freedom robotic arm prototype for car industry test application

Ricardo Miguel Carvalho Santos

Dissertação de Mestrado

Orientador na FEUP: Prof. João Manuel R.S. Tavares

Orientador na empresa: Sezgin Goekcen



Mestrado Integrado em Engenharia Mecânica

Junho 2018

À minha família.

Abstract

The goal of this project was the design and validation of a five degrees of freedom robotic arm prototype with the purpose to fulfil the Bosch Car Multimedia optomechanical laboratory needs for product tests and validation. To develop this project, a set of functionalities and parameters that had to be fulfilled were considered. Among the most important are the number of degrees of freedom that allow carrying out the desired measurements and the possibility of applying different measuring devices.

An investigation on robotics was made, more specifically, robotic arms and their structure, as well as the various areas of interest for their development. The production process inherent to the production of some components of the prototype was also addressed.

Besides the necessary knowledge regarding robotics, it was also important to use projects with an open source philosophy to obtain useful resources for the project development. The design of the prototype was carried out in a CAD software in, order to analyze the assembly in a virtual environment before proceeding to its manufacture. The design of the components and their selection was based on the execution of simplified calculations and analysis.

For the correct movement control of the prototype, an analysis and simulation of the kinematics was made using appropriate software, in this case MATLAB. The electrical and firmware / software components, responsible for the control inputs, were also analyzed

The components were manufactured using 3D printing processes. Pre-assembly electrical tests were performed on the various components. After the assembly of the prototype motion components, the necessary verifications and calibrations were made to certify the correct functioning, as well as a performance analysis regarding motion control.

Desenvolvimento e validação de um braço robótico com 5 graus de liberdade para validação e teste de produtos na indústria automóvel

Resumo

O objetivo deste trabalho foi o desenvolvimento e validação do protótipo de um braço robótico com cinco graus de liberdade, capaz de preencher os requisitos do laboratório opto mecânico da Bosch Car Multimedia Portugal S.A. para validação e teste de produtos. Para o desenvolvimento deste projeto, foi tido em consideração um conjunto de funcionalidades e parâmetros que tinham que ser cumpridos. Entre os mais importantes encontra-se o número de graus de liberdade que permitem realizar as medições pretendidas e a possibilidade de aplicar diversos aparelhos de medição.

Fez-se uma investigação sobre robótica, mais concretamente braços robóticos e a sua estrutura, assim como as diversas áreas de interesse para o seu desenvolvimento. O processo de produção inerente à produção de alguns componentes do protótipo também foi abordado.

Para além do conhecimento necessário em relação à robótica, foi, também, importante o uso de projetos com filosofia *open source*, para obter recursos uteis ao desenvolvimento do projeto. O desenho do protótipo foi realizado num software de CAD, de modo a analisar a montagem num ambiente virtual antes de passar ao seu fabrico. O dimensionamento dos componentes e a sua seleção apoiou-se na execução de cálculos e análises simplificadas.

Para o correto controlo de movimento do protótipo, fez-se uma análise e simulação da cinemática, recorrendo a software apropriado, neste caso MATLAB. Foi, também, analisada a componente elétrica e de *firmware/software*, responsável pelos inputs de controlo.

Os componentes foram fabricados, usando processos de impressão 3D. Foram feitos testes elétricos e pré-montagem, dos vários componentes. Após a montagem dos componentes elétricos do protótipo, foram efetuadas as verificações e calibrações necessárias, de modo a certificar o correto funcionamento, assim como análises de performance relativamente ao controlo do movimento.

Acknowledgements

I would like to thank everyone that helped me to accomplish my goals regarding this thesis:

To my advisor at FEUP, Prof. João Manuel R.S. Tavares, for the support and guidance throughout the project and all the other teachers who helped me during my academic life.

To my advisor and manager at Bosch Car Multimedia, Sezgin Goekcen, to my mentor, João Marques, to my team mates Bruno Vilaça, Filipe Valente, Vasco Pinto, Pedro Monteiro, Pedro Moreira, Marcelo Domingues, Hernani Abreu and Rui Barros for the support, integration and the great working environment. A special mention to Rui Barros for all the direct support to accomplish my goals regarding this project. In addition, to everyone at Bosch Car Multimedia that at a certain moment helped me.

To my integration team, Bruna Cunha, Patrícia Gonçalves, André Pires, Albino Alves and Carlos Pires for all the moments of fun and companionship.

To my family that were always there to support me, specially my parents and sister Ana, a friend for life.

To all friends who, since 1995, were always there for me, specially André Martingo, Gonçalo Sá and Pedro Capa; to all the others who crossed my way and helped me to achieve my personal and academic goals, specially Claudia Moreira, Mariana Winck and André Pires.

Contents

1	Introduction.....	1
1.1	Project Framework and Motivation.....	1
1.2	Bosch Group.....	2
1.3	Bosch Car Multimedia Portugal, S.A.	3
1.4	Project Goals.....	4
1.5	Methodology.....	4
1.6	Structure.....	5
2	Literature Review	7
2.1	Robotics	7
2.1.1	Industrial Robotics	9
2.1.2	Advanced Robotics	10
2.2	Mechanical Structure	11
2.2.1	Robot Manipulators.....	11
2.2.2	Mobile Robots.....	14
2.3	Robots Modelling, Planning and Control	14
2.3.1	Modelling	14
2.3.2	Planning.....	15
2.3.3	Motion Control	15
2.4	3D Printing	17
2.5	Summary.....	18
3	Development	19
3.1	Assumptions.....	19
3.2	Open Source Projects	19
3.2.1	Open-Source	19
3.2.2	BCN3D Moveo.....	19
3.2.3	Thor	20
3.2.4	AR2.....	21
3.2.5	6DOF Robotic Arm	21
3.2.6	Comparison	21
3.3	Robot Design	22
3.3.1	Module Waist.....	23
3.3.2	Module Shoulder.....	24
3.3.3	Module Elbow	26
3.3.4	Module Wrist.....	27
3.3.5	End Effector.....	28
3.4	Calculations.....	29
3.4.1	Stepper Motors	29
3.4.2	Power Supply.....	35
3.5	Summary.....	36
4	Motion Control.....	37
4.1	Kinematics.....	37
4.1.1	Direct Kinematics.....	37
4.1.2	Inverse Kinematics	40
4.1.3	Differential kinematics.....	41
4.2	Dynamic model	41
4.3	MATLAB Simulation	41
4.3.1	Direct Kinematics.....	42
4.3.2	Inverse Kinematics	44
4.4	Control.....	45
4.4.1	Electronics	45

4.4.2	Firmware.....	47
4.5	Summary.....	48
5	Implementation.....	49
5.1	3D Printing.....	49
5.1.1	PolyJet.....	49
5.1.2	FDM.....	50
5.1.3	Printed parts.....	51
5.2	Assembly and testing.....	53
5.3	Performance.....	53
5.4	Summary.....	55
6	Conclusions and Future Work.....	57
6.1	Final Conclusions.....	57
6.2	Future working perspectives.....	58
	References.....	61
Appendix A:	Other activities.....	64
Appendix B:	Data Sheets.....	66
Appendix C:	Mass properties.....	69
Appendix D:	Components datasheet.....	72
Appendix E:	Bill of Materials.....	75

List of figures

Figure 1 - BMW CHUD.	1
Figure 2 - Bosch logo.	2
Figure 3 - Bosch business developments worldwide in 2017 (URL2).....	2
Figure 4 - Bosch Car Multimedia, S.A. development center in Braga.....	3
Figure 5 - 2017 Bosch data in Portugal (URL4).	3
Figure 6 - Flowchart of the developed project.	4
Figure 7 - Estimated annual supply of industrial robots at year-end by industries worldwide (Kutzbach 2017a).	8
Figure 8 - Projections for 2017 and 2018-2020 service robots (Kutzbach 2017b).	9
Figure 9 - Industrial robots on automotive industry (URL5).	10
Figure 10 - Mars Science Laboratory (URL6).	10
Figure 11 - Different joints for manipulators (adapted from (Santos 2004)).	11
Figure 12 - Cartesian manipulator work space (Siciliano et al. 2009).	12
Figure 13 - Cylindrical manipulator work space (Siciliano et al. 2009).	12
Figure 14 - Spherical manipulator work space (Siciliano et al. 2009).	13
Figure 15 - SCARA manipulator work space (Siciliano et al. 2009).	13
Figure 16 - Articulated or anthropomorphic workspace (Siciliano et al. 2009).....	14
Figure 17 - Autonomous mobile robot (URL7).....	14
Figure 18 - Robot manipulator trajectory planning (URL8).	15
Figure 19 - Arduino MEGA 2560 board, used to control the robot.	16
Figure 20 - RAMPS 1.4 shield, used to simply the interface between hardware.	16
Figure 21 - TB6560, used to drive a stepper motor.....	17
Figure 22 - 3D printing flow chart (Campbell et al. 2011).....	17
Figure 23 - BCN3D Moveo project (URL11).	20
Figure 24 - Thor robot project (URL12).	20
Figure 25 - AR2 robot project (URL13).....	21
Figure 26 - 6DOF robotic arm project (URL14).	21
Figure 27 - Evaluation methodology followed to choose the base project.	22
Figure 28 - Prototype drawing with modules identification.....	23
Figure 29 - Module base with printed parts identified.	23
Figure 30 - Base (10-1M2A) assembled on ThorLabs Solid Aluminum Optical Breadboards.	24
Figure 31 - Robot waist with description of used systems.....	24
Figure 32 - Module shoulder drawing.	24
Figure 33 - Module shoulder with the two parts divided and respective connections.	25
Figure 34 - Transmission system drawing, with the parts identified.....	25

Figure 35 - Module elbow drawing.	26
Figure 36 - Module elbow connections.	26
Figure 37 - 5:1 gearbox parts to 3D print.	27
Figure 38 - Module wrist drawing.	27
Figure 39 - Module wrist transmission systems.	28
Figure 40 - End effector base drawing.	28
Figure 41 - End effector base connections.	28
Figure 42 - Robotic arm completely stretched out. A_i -represents the actuators; W_i -represents the weight of each link; l_i -distance between joints; D -represents the weight of a device.	30
Figure 43 - Perspective drawing showing the z axis (the axis of rotation) and the parallel axis through the mass center (Serway and Jewett 2018).	31
Figure 44 - Scheme of flat belt transmission (Della Pietra and Timpone 2013).	33
Figure 45 - Power supply used in this project.	36
Figure 46 - Standard Denavit-Hartenberg link parameters (Corke 2017).	38
Figure 47 - Frame assignment for the prototype.	39
Figure 48 - Denavit-Hartenberg parameters matrix command window.	42
Figure 49 - $r = SerialLink(dh)$ command window.	43
Figure 50 - Zero joint angles matrix.	43
Figure 51 - Prototype plot for the zero joint angles.	43
Figure 52 - Different positions simulation.	44
Figure 53 - End effector position matrix.	44
Figure 54 - $ikine$ command.	44
Figure 55 - Inverse kinematics check.	45
Figure 56 - Electronic scheme.	45
Figure 57 - TB6560 settings.	46
Figure 58 - Marlin_BCN3D_Moveo firmware opened in IDE platform.	47
Figure 59 - Pronterface, the software used to control the robot stepper motors.	48
Figure 60 - Objet30 Prime.	49
Figure 61 - Polyjet process (Singh 2011).	50
Figure 62 - FDM process (URL17).	50
Figure 63 - Prototype drawing with the location of 3D printed parts.	51
Figure 64 - Precision and repeatability performance feature (adapted from (Santos 2004)). ...	54
Figure 65 - Prototype reachable distance.	54
Figure 66 - End effector motion.	58
Figure 67 - JIG to measure light guide efficiency.	64
Figure 68 - First version of the prototype, produced using Polyjet technology.	65
Figure 69 - Thorlabs aluminum breadboard 300 x 1200 x 12.777 (URL22).	66

Figure 70 - Objet30 Prime specifications (URL23).	67
Figure 71 - VEROCLEAR RGD810 proprieties (URL24).	67
Figure 72 - FLIR Flea 3 datasheet (URL19).	68
Figure 73 - Mass properties for joint 1.	69
Figure 74 - Mass properties for joint 2.	69
Figure 75 - Mass properties for joint 3.	70
Figure 76 - Mass properties for joint 4.	70
Figure 77 - Mass properties for joint 5.	71
Figure 78 - LIN ENG 4118M-01-RO datasheet (URL25).	72
Figure 79 - ST 5709L1108-B datasheet (URL26).....	73
Figure 80 - 42BYGHW811 datasheet (URL27).....	73
Figure 81 - Arduino Mega datasheet (URL28).	74

List of tables

Table 1 - Open source project's evaluation.	22
Table 2 - Torque due to gravity.	31
Table 3 - Moment of inertia.	32
Table 4 - Torque due to inertia.	32
Table 5 - Total torque at each joint.	33
Table 6 - Data for transmission calculations.	34
Table 7 - Torque transmission.	34
Table 8 - Required and available torque comparison.	35
Table 9 - Components needs, regarding power supply.	36
Table 10 - Denavit-Hartenberg parameters.	38
Table 11 - Prototype D-H parameters.	40
Table 12 - Denavit-Hartenberg parameters for the MATLAB simulation.	42
Table 13 - TB6560 configuration.	46
Table 14 - Drawing and 3D printed parts.	51
Table 15 - Performance results.	55
Table 16 - Performance results.	55
Table 17 - Future improvements.	59
Table 18 - Project bill of materials.	75

1 Introduction

This chapter intends to present the framework and motivation for this project, as well as the Bosch group and Bosch Car Multimedia Portugal, S.A., where the internship was carried out.

The project goals, methodology and structure of this thesis are also described.

1.1 Project Framework and Motivation

The internship was performed at the CM/CI2-ECM8 team that integrates the Bosch Car Multimedia Portugal development department. The team works in the development and support to production of the new Ford and BMW combiner head up display (CHUD). Figure 1 shows the BMW CHUD, used in several BMW group cars; this system projects useful information for the driver without the need to look away from the road.

Due to the necessity of performing tests and validations that need precision and stability, the goal of the internship was the design and validation of a robotic arm prototype with five degrees of freedom, fabricated using rapid prototyping technologies. With the creation of a new laboratory for the development department, the robotic arm will allow not only the CM/CI2-ECM8 team but also all the department to test and validate faster and more efficiently.



Figure 1 - BMW CHUD.

In a personal level, the project was interesting and challenging, because it gave the possibility to work in mechanical, electrical and software topics. The possibility to design and build something that has an important role in industry nowadays, a robot, was something that motivated me since day one.

1.2 Bosch Group

The group Bosch (Figure 2) was founded in 1886 when Robert Bosch opened the “Workshop for Precision Mechanics and Electrical Engineering” that was located in Stuttgart, Germany (URL1).



Figure 2 - Bosch logo.

Nowadays, the administrative structure of Robert Bosch GmbH allows enterprising freedom to the Bosch Group, making possible to think in long term. Robert Bosch Stiftung GmbH, a charitable foundation, is the major shareholder with 92%; the Bosch family owns 7% and Robert Bosch GmbH the remaining 1%. Regarding the vote rights, Robert Bosch Industrietreuhand KG, an industrial branch, owns 93%, leaving the remaining 7% to the Bosch family (URL2).

Under the slogan "Invented for life" Bosch wants its products to spark enthusiasm, improve people's life quality and help to preserve natural resources. The group operations are divided into four business sectors:

- Solutions of Mobility (BBM);
- Industrial Technology;
- Energy and Building Technology;
- Goods of Consumption.

In Figure 3, some 2017 Bosch group data shows a sales amount of 78.1 billion euros, representing a growth of 6.8%, and record earnings of 5.3 billion euros. The number of global associates is roughly 402200, about 62500 of them working on research and development, distributed by 125 locations (URL2).



Figure 3 - Bosch business developments worldwide in 2017 (URL2).

1.3 Bosch Car Multimedia Portugal, S.A.

Bosch Car Multimedia history in Braga started in 1990; nowadays, Braga plant is the main manufacture of Car Multimedia division, due to the expertise acquired through time, being a reference in the market of the car electronics (URL3). Recently, investment in development has increased, with the creation of the development center (Figure 4).



Figure 4 - Bosch Car Multimedia, S.A. development center in Braga.

Braga plant's main business is the production and development of information, entertainment and instrumentation systems for the car industry (URL3). The two wheels market is getting a bigger share of the work developed in Braga Plant and it is seen as a bet for the future. As shown in Figure 5, Bosch is one of the largest industrial employers in Portugal, with more than 4800 collaborators and an amount of 1.5 thousand millions euros in sales, in 2017 (URL4).

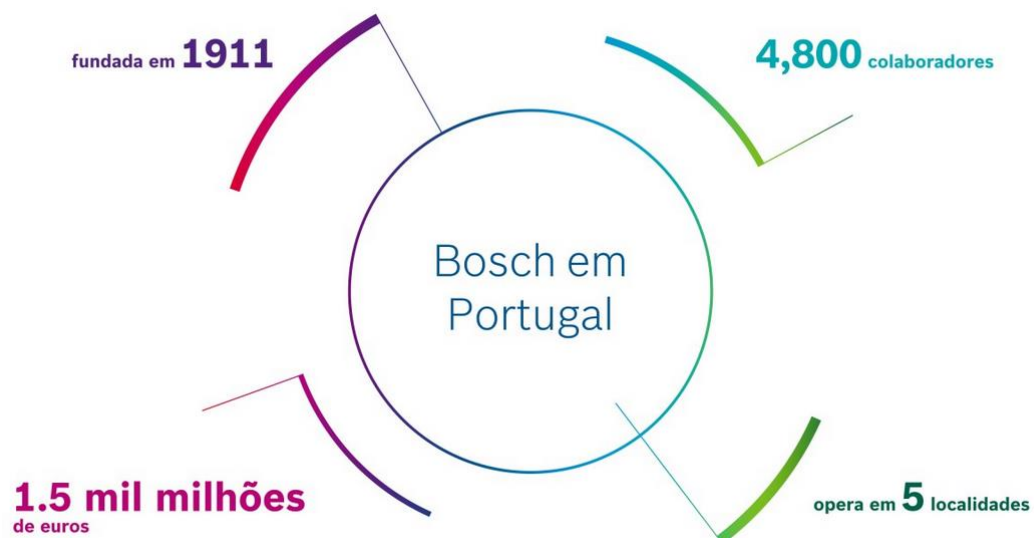


Figure 5 - 2017 Bosch data in Portugal (URL4).

1.4 Project Goals

The main goal of this project is to design and validate a five degrees of freedom robotic arm with the purpose to fulfil the Bosch Car Multimedia optomechanical laboratory needs for product tests and validation. The robotic arm must be able to integrate and position, with the correct stability and accuracy, different measuring devices such as spectrometers, power-meters or image acquisition cameras. The robotic arm development was supported by open source projects.

To achieve the main goal, the project was divided in the following steps:

- Design and components validation of the robotic arm;
- Hardware/electronics and software/firmware implementation;
- Test motion control of the prototype.

1.5 Methodology

To develop a good prototype, it is necessary to get as much information as possible about the topic to be worked on. Therefore, the first step of the project was a literature review on robotics. A brief presentation on the history since the beginning until the present, the different type of robots and the parameters that characterize them were addressed, as well as the technologies and processes to design and build the parts necessary to create a prototype. A study on available open source projects that could suit as base for the project was also performed.

The second step of the project consisted in CAD redesign and validation of all parts and components to properly build the prototype. After this, it was necessary to fabricate the parts and test the electronics.

The final step was the testing of the prototype motion control to validate and obtain results regarding its performance.

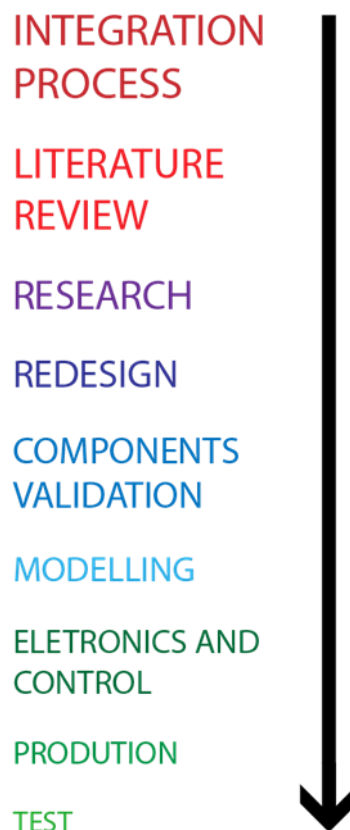


Figure 6 - Flowchart of the developed project.

1.6 Structure

Additionally to this chapter, this thesis is organized with five more chapters:

- Chapter 2: Literature Review. This chapter presents an introduction on robotics, covering history and important topics on robotics and an introduction on the 3D printing process.
- Chapter 3: Development. This chapter describes all the development process, from parts redesign to components selection and validation.
- Chapter 4: Motion Control. This chapter addresses the motion control theme, the kinematic model, electronics and control software.
- Chapter 5: Implementation. This chapter describes the production processes for the parts, as well as the assembly and testing. It also addresses future improvements.
- Chapter 6: Conclusions and Future Work. This chapter presents the conclusions on the project, the internship and future working perspectives.
- Appendix: Description of other work developed during the internship.

2 Literature Review

This chapter enlightens about robotics and the production process used for some parts in this project. Robotics are introduced, as well as its covering history and important topics, the mechanical structure and robots motion, planning and control. It is also made an introduction on the 3D printing process.

2.1 Robotics

The field of robotics can be recognized as an interdisciplinary subject concerning the cultural areas of mechanics, control, computers, and electronics (Siciliano et al. 2009). It studies machines that can replace human beings in several tasks, either physical or decision-making. This is an old ambition, as it is possible to state in old tails, such as the legend of Titan Prometheus who built humans from clay or Frankenstein, in modern times. In the Industrial Age, automaton, a mechanical creature, has been assigned with the task of performing human labor duties; the Czech playwright, Karel Capek, first introduced this concept when he used the word robot, *robota* in Slav languages, which meant forced labor. The concept of robot as a mechanical artifact, without feelings, started in 1940s with Isaac Asimov, a Russian science fiction writer; the behavior of the robot was controlled by a “positronic” brain programmed by humans to satisfy certain rules of ethical conduct. Asimov introduced the term Robotics as the science which studies robots and it was based on three fundamental laws:

- First law: A robot must not injure a human being or, through inaction, allow a human being to come to harm;
- Second law: A robot must obey the orders given by human beings, except where such orders would conflict with the First Law;
- Third law: A robot must protect its own existence, as long as such protection does not conflict with the First or Second Laws.

These laws are the base to design robots that since then have the connotation of an industrial product, which is developed by engineers and specialized technicians. In a scientific interpretation, a robot is a machine that can modify the operative environment in which it operates. This is accomplished by performing actions that are conditioned by the rules of behavior of the machine, as well as data that the robot collects on its status and on the environment. Robotics is commonly defined as the science that studies the intelligent connection between perception and action (Siciliano et al. 2009).

A robotic system is a complex system, represented by multiple subsystems. The essential component is the mechanical system, probably with a locomotion mechanism (wheels, crawlers, mechanical legs) and a manipulation mechanism (mechanical arms, end effectors, artificial hands). The concretization of such a system refers to the framework of design of articulated mechanical systems and materials choice.

The ability of performing an action, both by locomotion and manipulation, is produced by an actuation system, which animates the robot’s mechanical components. The design of such a

system refers to the context of motion control, dealing with servomotors, stepper motors, drives and transmissions.

The ability of perception is dependent of a sensory system, which can acquire data on the mechanical system internal status, as well as the external status of the environment. The design of such a system refers to the context of materials properties, signal conditioning, data processing and information retrieval.

The ability of connecting action to perception in an intelligent fashion is provided by a control system, which can command the execution of an action in respect to the goals set by a task planning technique, as well as by constraints of the robot and the environment. The design of such a system follows the same feedback principle devoted to control of human body functions, possibly exploiting the description of the robotic system's components (modelling).

Nowadays, the industrial robot sales are growing worldwide, by 16% to 294.312 units, supported mainly by the automotive and electrical/electronics industry. In 2016, these industries represent a 66% share of the total supply. The bigger growth was in the electrical/electronics industry, but the automotive is still the major customer regarding this type of technology, with a share of 35% of the total supply (Kutzbach 2017a). This data is shown in Figure 7.

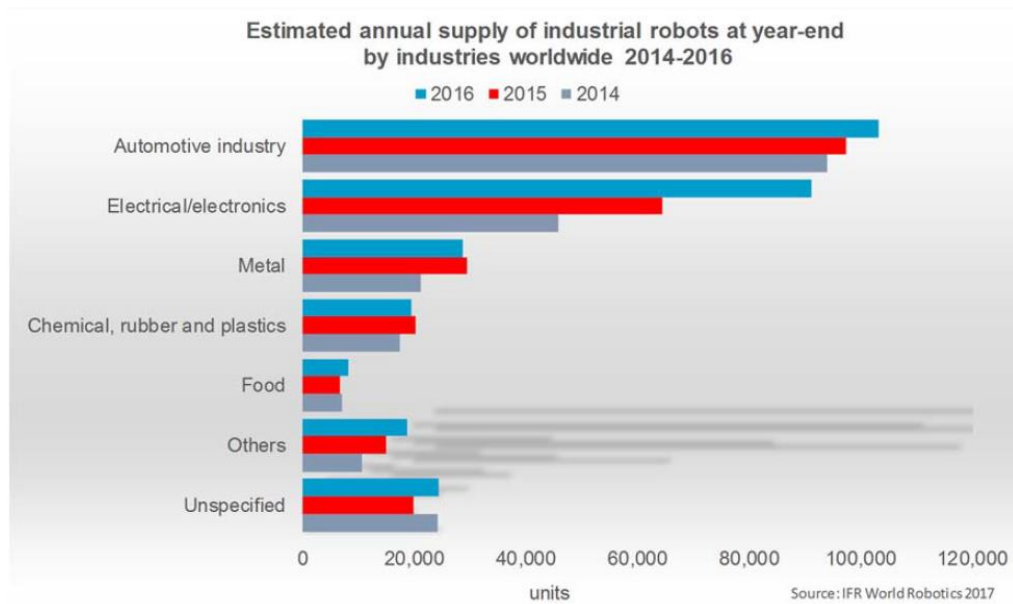


Figure 7 - Estimated annual supply of industrial robots at year-end by industries worldwide (Kutzbach 2017a).

Regarding service robots, the number of sales and sales value, in 2016, have increased 24% to 59.706 units and 2% to 4.7 thousand million dollars (Figure 8). The reason behind this rather low increase rate is the considerable decrease of high valued defense robots. Around 285,000 service robots for professional use, since 1998, have been counted in these statistics. It is not possible to estimate how many of these robots are still working due to the diversity of these products, resulting in varying utilization times (Kutzbach 2017b).

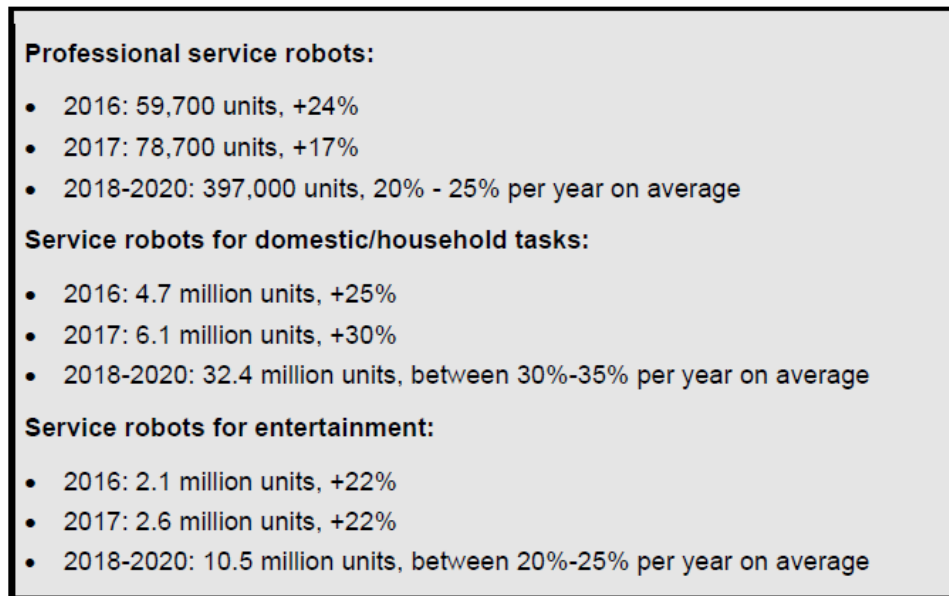


Figure 8 - Projections for 2017 and 2018-2020 service robots (Kutzbach 2017b).

2.1.1 Industrial Robotics

Industrial robotics is the discipline related with robots design, control and applications for industry (Figure 9); the resulting products have reached a level of mature technology. An industrial robot operates in a structured environment with usually well-known geometrical and physical characteristics and it has relevant versatility and flexibility characteristics. According to Robot Institute of America, “a robot is a reprogrammable multifunctional manipulator designed to move materials, parts, tools or specialized devices through variable programmed motions for the performance of a variety of tasks” (Spong and Vidyasagar 2008).

The primordial industrial robots were developed in 1960s and the subsequent decades; popularity increased with the possibility to create automated manufacturing systems. Comparing with its precursors, the robot manipulators are characterized by their versatility, adaptability and repeatability. These systems improved human operator ergonomics, productivity, product quality and decreased the manufacturing costs (Siciliano et al. 2009). Automation is the key to have an efficient industrial robot. The goal is to replace humans in a manufacturing process, regarding not only the physical operations but also the intelligent processing of information. There are three levels of automation, according to the goal.

- Rigid automation: Used when the goal is the mass production of the same type of products. The need to produce a large quantity of parts, with high productivity and quality standards, demands the use of fixed operational sequences; these operations are performed on the work piece by special purpose machines.
- Programmable automation: Used when the goal is to manufacture low-to-medium batches of products of different types. A programmable automated system allows to easily change the sequence of operations executed on the work pieces, to vary the range of products. The machines employed are more versatile and capable of manufacturing different objects belonging to the same group of technology. Most of the products available on the market today are manufactured by programmable automated systems.
- Flexible automation: Represents the evolution of programmable automation. It is used when the goal is to allow manufacturing of variable batches of different products, by minimizing the time lost in reprogramming the sequence of operations and the number of machines employed to pass from one batch to the next. The use of a flexible manufacturing system demands strong integration between computer technology and industrial technology.

Other possible application for industrial robots, besides handling and manipulating material in a manufacturing process is to perform measurements to evaluate a product quality. The ability to position the end effector, with a measurement device, in a 3D space, allows the robot to be used to detect manufacturing imperfection, contour defects and so on (Siciliano et al. 2009).



Figure 9 - Industrial robots on automotive industry (URL5).

2.1.2 Advanced Robotics

Advanced robotics is the discipline related with the study of robots with autonomy characteristics, operating in badly structured or unstructured environments, with geometrical and physical characteristics not known a priori. This field is still in its youth, but there are many motivations to learn and study more about the technologies associated. The need to find an alternative to explore or perform activities that are not safe or possible for a human being (field robots), or develop products to improve life quality (service robots) are some markets that motivate the research for developments in this robotics discipline (Siciliano et al. 2009).

1. Field Robots

Field robots (Figure 10) are used to perform activities that human beings are not able to perform or could endanger us. Typical scenarios are exploration of deep oceans, space or contaminated areas. When a catastrophe occurs, fires in tunnels or earthquakes for example, human rescue teams can operate together with robots. In the military field, autonomous aircrafts and missiles are being used (Siciliano et al. 2009).



Figure 10 - Mars Science Laboratory (URL6).

2. Service Robots

A service robot, according to the International Federation of Robotics, is a robot “that performs useful tasks for humans or equipment excluding industrial automation application.” (DIS 2012). Field robots could be included in this definition, but a distinction between them is made. Nowadays it is possible to transform prototypes into commercial products; due to this, service robotics technologies are growing at a quick pace and replacing human’s everyday tasks. There are products that can improve the elderly and impaired quality of life through autonomous wheelchairs, mobility aid lifters or rehabilitation robots. Surgery assistance systems for medical applications are now a reality, as well as autonomous vacuum cleaner or lawn mowing robots (Siciliano et al. 2009).

2.2 Mechanical Structure

The robot most important feature is the mechanical structure. Ordinarily robots are classified as:

- Robot manipulators, those with a fixed base;
- Mobile robots, those with a mobile base.

2.2.1 Robot Manipulators

A robot manipulator mechanical structure consists on a sequence of rigid bodies, or links, connected by an articulation, or joint. An arm that guarantees mobility, a wrist that confers dexterity, and an end effector that allows the robot to perform the required task, defines a manipulator (Craig John 1989).

A kinematic open chain can be described as a chain where there is only one sequence of links connecting the base and the end effector. The open kinematic chain is the manipulator fundamental structure. When a sequence of links forms a loop, the kinematic chain is closed.

Joints guarantee the manipulator’s mobility and can be classified in different categories, according to the movement that two links connected by that joint are able to perform. As displayed in Figure 11, joints can be revolute, prismatic or spherical. In an open kinematic chain, each joint provides the structure with an additional degree of freedom (Santos 2004).

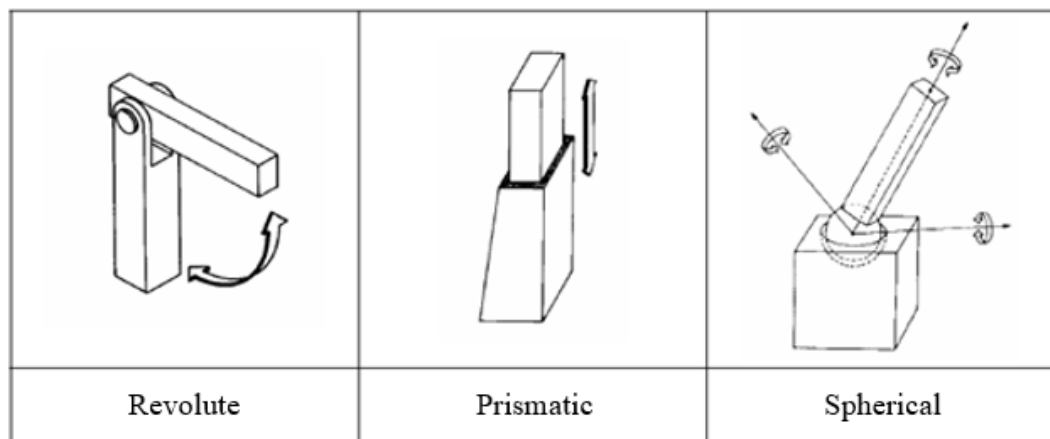


Figure 11 - Different joints for manipulators (adapted from (Santos 2004)).

The number of degrees of freedom (DOF) equals the number of independent positions variables that a manipulator possesses and that has to be specified to locate all parts of the mechanism. To perform the given task in a proper way, it is necessary to distribute properly the degrees of freedom. In case of a typical industrial robot, the number of joints equals the number of DOF, because a manipulator is usually an open kinematic chain and each joint is usually defined with

a single variable. Generally, we describe the position of the manipulator by giving a description of the tool frame, attached to the end effector, relative to the base frame, attached to the nonmoving base of the manipulator (Craig John 1989).

The workspace is the volume of space where the end effector can be placed by the manipulator. When a robot is classified by its kinematic structure only the main joints are used. There are five types of structures:

- Cartesian: accomplished by three prismatic joints whose axis usually are mutually orthogonal. Each degree of freedom corresponds to a Cartesian space variable and performs straight motions in space. The workspace is a rectangular parallel-piped (Figure 12). With this type of structure, it is possible to move objects with big dimensions and heavy weight, but it has low dexterity structure, because all the joints are prismatic. The motors actuating on the joints of a Cartesian manipulator are normally electric and occasionally pneumatic.

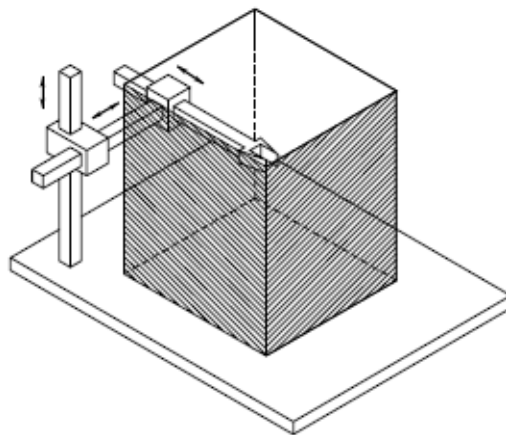


Figure 12 - Cartesian manipulator work space (Siciliano et al. 2009).

- Cylindrical: the difference between cylindrical and Cartesian structures is that the first prismatic joint is replaced by a revolute joint. If the task to perform is defined in cylindrical coordinates, each degree of freedom corresponds to a Cartesian space variable. The workspace is a portion of a hollow cylinder (Figure 13). Cylindrical manipulators are used to carry objects of big dimensions; in such a case, the use of hydraulic motors is preferred to electric motors.

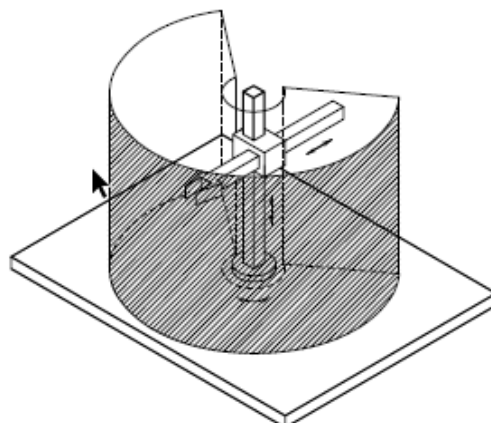


Figure 13 - Cylindrical manipulator work space (Siciliano et al. 2009).

- Spherical: the difference between cylindrical and spherical structures is that the second prismatic joint is replaced by a revolute joint. If the task to perform is defined in spherical coordinates, each degree of freedom corresponds to a Cartesian space variable. The

workspace is a portion of a hollow sphere (Figure 14). Spherical manipulators are mainly used for machining. Electric motors are typically used to actuate the joints.

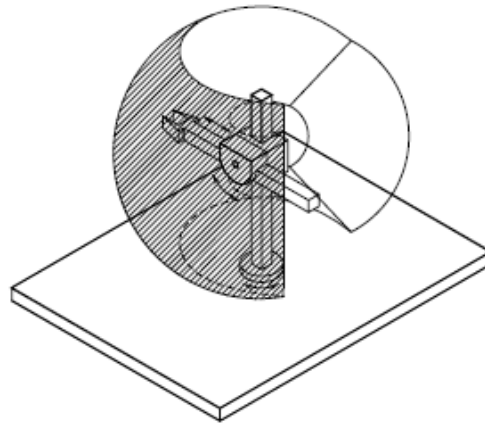


Figure 14 - Spherical manipulator work space (Siciliano et al. 2009).

- SCARA (selective compliance assembly robot arm): this type of structure is accomplished by disposing two revolute joints and one prismatic joint in such a way that the axes of motion are parallel. The correspondence between the DOF and Cartesian space variables is maintained only for the vertical component of a task, described in Cartesian coordinates. The workspace is described in Figure 15. SCARA manipulators are used to carry small objects. Electric motors are typically used to actuate the joints.

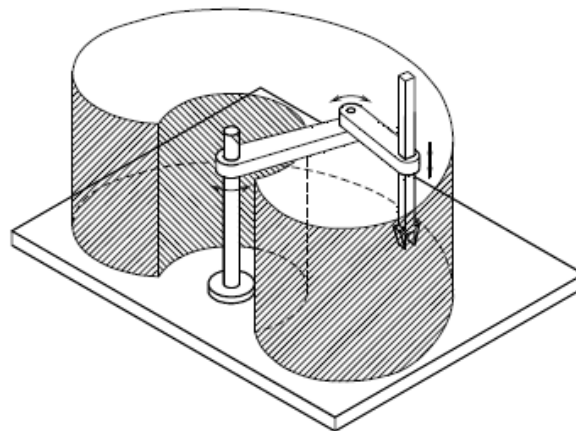


Figure 15 - SCARA manipulator work space (Siciliano et al. 2009).

- Articulated or anthropomorphic: this type of structure is accomplished by disposing three revolute joints. This structure is the most dexterous, since all the joints are revolute. The workspace is approximately a portion of a sphere (Figure 16). Electric motors are usually used to actuate the joints. Anthropomorphic manipulators are widely used in industry, in different applications.

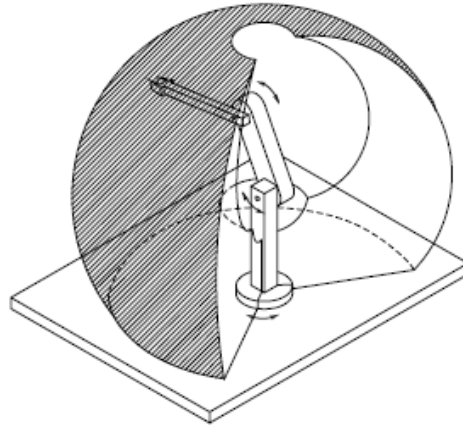


Figure 16 - Articulated or anthropomorphic workspace (Siciliano et al. 2009).

All these manipulators have an open kinematic chain, but in some cases closed chain kinematic manipulators are necessary, mostly when the payload has a larger weight (Santos 2004).

2.2.2 Mobile Robots

Comparing with a robot manipulator, a mobile robot's major feature is the ability to move freely in the environment, due to the mobile base. This type of robots are mainly used in service applications, where autonomous and extensive motion capabilities are required (Siciliano et al. 2009). Currently mobile robots are used in the industry (Figure 17), especially to move products with self-moving vehicles that do not need any kind of structure to guide them, like AGV's.



Figure 17 - Autonomous mobile robot (URL7).

2.3 Robots Modelling, Planning and Control

2.3.1 Modelling

Kinematics is the science of motion, which in this subject describes the motion of a robot mechanical structure in respect to a fixed reference Cartesian frame, ignoring the forces and torques. Regarding a robot manipulator, kinematics characterizes the analytical relationship between joints and end effector positions and orientation (Craig John 1989). To study a robot manipulator kinematics, there are two important topics to address:

- Direct kinematics: determination of the end effector position and orientation as a function of the joint variables with linear algebra tools;

- Inverse kinematics: determination of the joints variables as function of the end effector position and orientation with linear algebra tools.

The solution of both is crucial to guarantee the desired motion of the end effector. Differential kinematics characterizes the analytical relationship between joint and end effector motion in terms of velocities (Siciliano et al. 2009).

To properly choose the actuators, control and simulate the manipulator motion it is necessary a dynamic model; for the dynamic analysis, the motion of the manipulator takes in account the forces and torques acting on it.

2.3.2 Planning

The trajectory planning of robot manipulator (Figure 18) is less complex comparing with mobile robots, because of the constraints that the wheels generate. For a manipulator it is possible to assign a motion at the joints or directly at the end effector, depending if the purpose of the robot is, for example, handling objects or machining. The main goal of trajectory planning is to generate the timing laws for the relevant variables, joint or end effector, with an accurate description of the wanted motion (Siciliano et al. 2009).

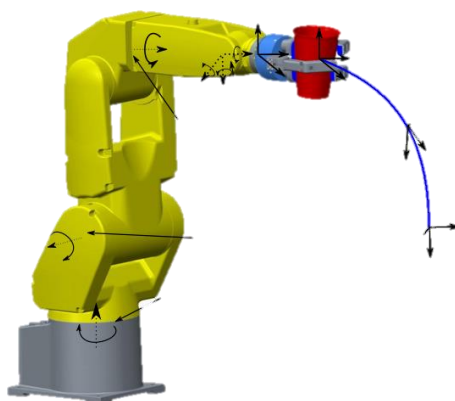


Figure 18 - Robot manipulator trajectory planning (URL8).

2.3.3 Motion Control

Accomplishing the intended motion by control laws requires the application of actuators, sensors and a control system that implement those laws. The robot manipulator control, as mentioned before, is a complex problem because the motion of one link influences the motion of other links and it is necessary to find the time behavior of the actuators to ensure the intended motion. Motion equations reveal the presence of coupling dynamic effects among the joints. The synthesis of the joint forces and torques should not be made considering only the knowledge of the dynamic model, since this does not completely describe the real structure. Therefore, manipulator control is entrusted to the closure of feedback loops; by computing the deviation between the reference inputs and the data provided by the proprioceptive sensors, a feedback control system is capable of satisfying accuracy requirements on the execution of the intended motion (Siciliano et al. 2009). Several control systems have different capabilities and prices, depending on the goal and needs.

1. Arduino

Arduino (Figure 19) is a cheap programmable microcontroller; it processes inputs and outputs between the device and external components connected to it. These boards are made of an Atmel AVR microprocessor, a crystal or oscillator and a 5V voltage regulator. Depending on

what type of Arduino you have, it may also have a USB socket to enable the connection to a computer to upload or retrieve data. The board exposes microcontroller's I/O (input/output) pins to enable you to connect those pins to other circuits or to sensors, etc.

The Arduino hardware and software are both Open Source; so, the code, schematics, design, etc., are all open for anyone to take and change if they want. Indeed, it is possible to purchase the appropriate components and make an Arduino on a breadboard or on a homemade PCB (printed circuit board). The Arduino can also be extended with the use of shields, which are circuit boards containing other devices, for example steppers, LCD displays, Ethernet modules, etc.; that can be simply connected to the top of the Arduino board to get extra functionalities. Shields also extend the pins (the places on the Arduino where it is possible to output or input data) to the top of a circuit board and still have access to all of them. Instead of a shield, it is possible to use a breadboard.

To program the Arduino, it is used the Arduino IDE (Integrated Development Environment), a free software that enables the programming in the language that the Arduino board understands. The language is based on C/C++ and can even be extended through C++ libraries. With IDE, it is possible to write step-by-step instructions and upload them. The Arduino will then carry out those instructions and interact with whatever is connected to it (McRoberts 2013).



Figure 19 - Arduino MEGA 2560 board, used to control the robot.

2. RAMPS 1.4

RAMPS 1.4 shields were designed to fit all the electronics needed for RepRap, in a small and cheaper kit. It interfaces with an Arduino Mega and all his capabilities. The RAMPS design includes plug in stepper drivers and extruder control electronics on an Arduino Mega shield for easy assembly, part replacement, upgrade-ability and expansion. Version 1.4 (Figure 20) uses surface mount capacitors and resistors to assemble more components (URL20).

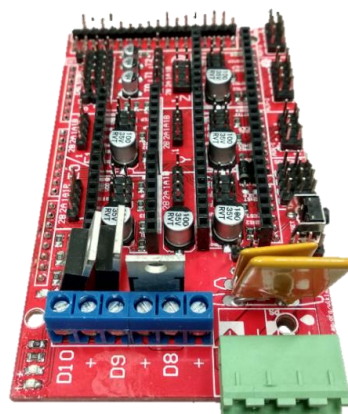


Figure 20 - RAMPS 1.4 shield, used to simplify the interface between hardware.

3. TB6560 Driver

TB6560 driver (Figure 21) is a micro stepping axis stepper driver that uses the TOSHIBA TB6560 chip, based on pure-sine current control. With that and self-adjustment technology, the motors work with smaller noise, lower heating, smoother movement and have better performances at high speed, when compared with other drivers in the market.

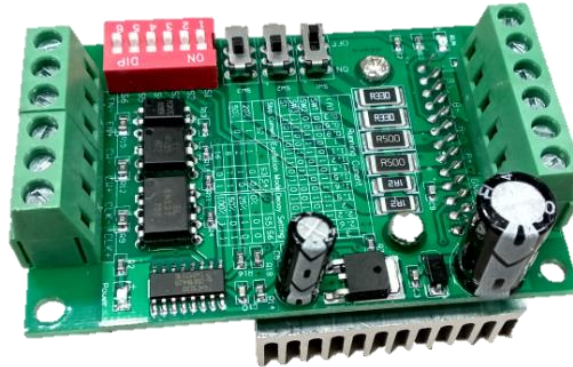


Figure 21 - TB6560, used to drive a stepper motor.

2.4 3D Printing

The industrial revolution and therefore traditional manufacturing, which usually are subtractive technologies, have boosted the world, as we know; as there are some limitations with these technologies, regarding final shape, cost, velocity of production and need for specialized operators, new approaches were developed. To produce fast and cheaper, complex products additive technologies, based on adding material layer by layer, were proved a good alternative. 3D printing is one of the best options comparing with other additive technologies because of the cost of the equipment and the easy integration with CAD software. (Berman 2012) The prototype can be printed using different 3D printing technologies and materials that differentiate from one another, based on several characteristics. As usually there is not an ideal process and material, it is necessary to evaluate what is going to be the final goal of the prototyped piece. Some of the factors that affect this choice are mechanical properties, aesthetics, time to produce and cost. The flowchart to produce a 3D printed object is presented in Figure 22.

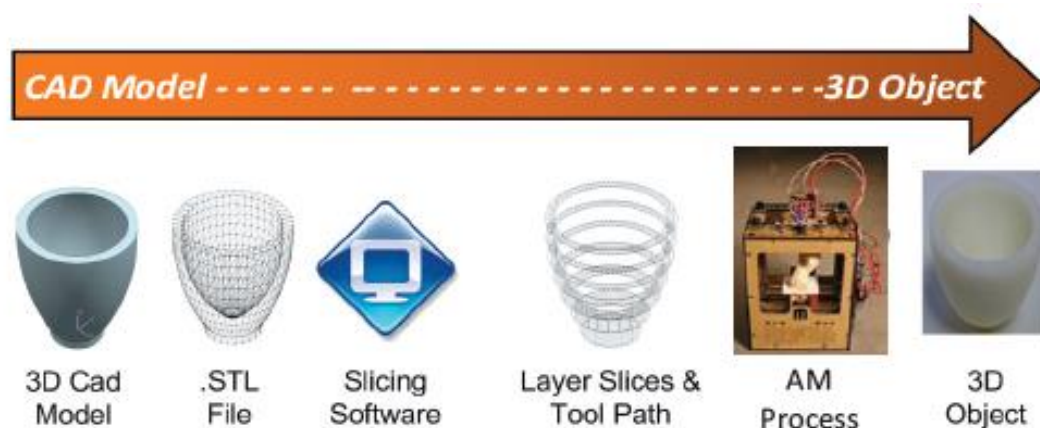


Figure 22 - 3D printing flow chart (Campbell et al. 2011).

2.5 Summary

The introduction made in this chapter, regarding robotics, was crucial to understand the different type of robots, their applicability and how motion is controlled and performed. The 3D printing process, which will be used to produce some parts, is also addressed, to understand the reason behind its use.

3 Development

This chapter presents some research on open source projects, the design of the prototype and the calculations made to validate some components.

3.1 Assumptions

A robotic manipulator is an essential device for the optomechanical lab that works for the car industry. With the growth of the Bosch Car Multimedia Portugal development department, the existing robot, a Stäubli TX series 60, is not able to perform all the required tasks, due to lack of time. To perform less demanding tasks, the challenge was to design and validate a five degrees of freedom robotic arm, based on open source projects. The idea of building a robotic arm was not new and some of the components were already available, so one of the requirements was to use those components. Other available resources were the 3D printer in the department, an Objet Prime 30 and DONE lab, an additive manufacturing laboratory at Minho University (a partnership with Bosch Car Multimedia Portugal) that gives the possibility to use some of the best technologies, regarding rapid prototyping.

3.2 Open Source Projects

The goal was to choose an open source project and make the needed modifications to perform the intended tasks. After some online research, there were four projects taken in consideration.

3.2.1 *Open-Source*

GitHub is the largest open source community in the world, with millions of projects. It is mostly used for code and it is possible to share, have access and modify the shared code (URL9). With the success of open source concept regarding software, some hardware projects started to apply this model and this helped consumers that want low cost alternatives to have access to them. One of the largest communities regarding open source hardware is hackday platform (URL10).

For a project to be considered open source, its license must qualify as “Open Source”. This means that the license must permit non-exclusive commercial exploitation of the licensed work, make available the work source code and permit the creation of derivative works from the work itself. Each of these principles is expressed in the Open Source Definition (Laurent 2004).

In 2007, a report from the Standish group stated that adoption of ‘open source’ has caused a drop in revenue to the proprietary software industry by about 60 billion dollars per year, which represents around 6% of the world market for software (Rubinstein 2007).

3.2.2 *BCN3D Moveo*

The BCN3D Moveo (Figure 23) is a robotic arm designed by BCN3D Technologies, a Fundació CIM project, in collaboration with the Departament d’Ensenyament from the Generalitat de

Catalunya. Its structure is entirely printed, using additive manufacturing technologies, and the software Arduino controls it is electronics.

This project was developed with an educational propose and to make a move for the robotics community progress. An Open Source, adaptable and low cost reproducible robotic arm could embrace several educational topics and be accessible for students at far lower price when compared with industrial equipment.

BCN3D Technologies shares all developed products on GitHub; therefore, all BCN3D Moveo files are available for everyone. It is possible to find there a bill of material (BOM), CAD files that can be modified, STL files ready to print, firmware and a manual to assemble and upload software. This allows anyone to build a robotic arm at home (URL11).



Figure 23 - BCN3D Moveo project (URL11).

3.2.3 Thor

The main purpose of this project was to create a robotic arm that could be used in universities and schools to teach robotics, instead of using simulation software or low accurate models. Thor (Figure 24) is a printable robotic arm with six degrees of freedom. Its configuration (yaw-roll-roll-yaw-roll-yaw) is the same one that is used on most manipulator robots, existing in the market. In its upright position, Thor has about 625mm and it can lift objects up to 750 grams. Having this in mind, the final prototype had to be affordable and, of course, Open Source (URL12).



Figure 24 - Thor robot project (URL12).

3.2.4 AR2

The AR2 (Figure 25) is a small desktop robot that is modeled like an industrial six axis robot. This is a DIY project that can be built from 3D printed components or from machined aluminum components, using low cost stepper motors and an Arduino controller (URL13).



Figure 25 - AR2 robot project (URL13).

3.2.5 6DOF Robotic Arm

This six axis robotic arm was built with servomotors. It also has a simple mechanism to use different tool heads on it and exchange them easily. Most parts can be fabricated almost without support material; so, they can be printed with many 3D printers. The movements will be done with instructions from the hardware controller, an Arduino due, and a computer (URL14).



Figure 26 - 6DOF robotic arm project (URL14).

3.2.6 Comparison

Table 1 indicates the pros and cons that led to the chosen project, according to six criteria and their importance.

- Available drawings: this takes in consideration if there were CAD files of the whole project, or only the stl files of the printable parts;
- Available components: there were some components that were already available (stepper motors and Arduino MEGA boards); since one of the goals was to use them, this criteria takes that in consideration;
- Aesthetics: this topic was taken in consideration because in the future, the robot will be used in the new laboratory;

have an overview of the prototype to understand the final concept and a representation of each module location.

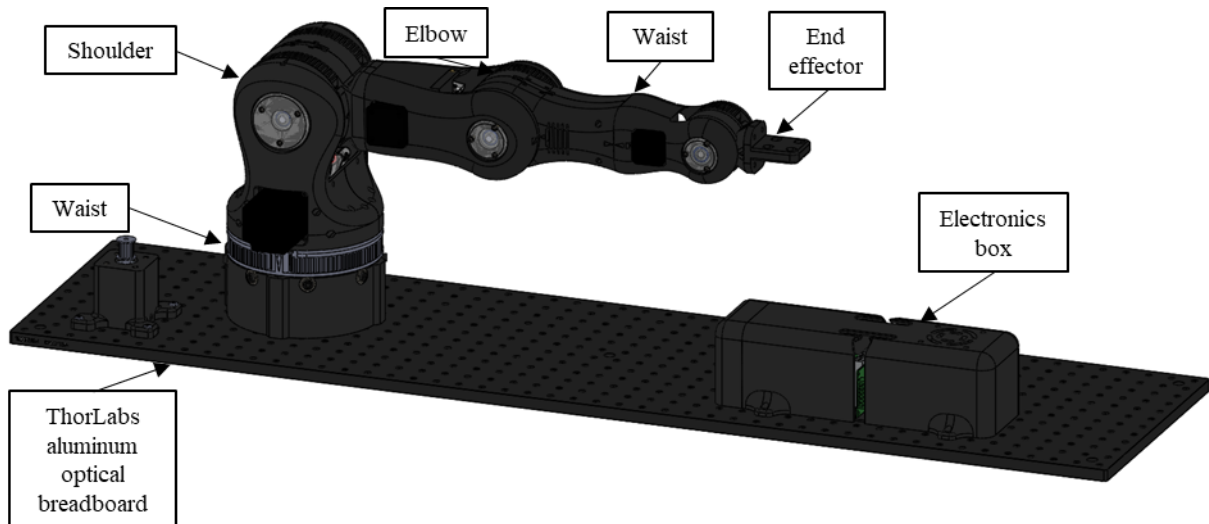


Figure 28 - Prototype drawing with modules identification.

3.3.1 Module Waist

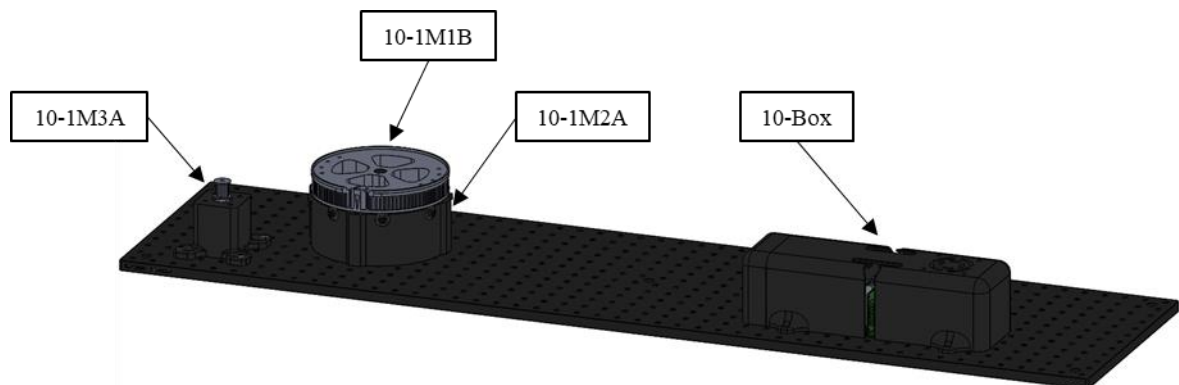


Figure 29 - Module base with printed parts identified.

Figure 29 presents all parts on module base. The robotic arm has to assemble on a Thorlabs aluminum optical breadboard. To do that, it was necessary to change the base (10-1M2A), regarding screwing. After consulting the ThorLabs Solid Aluminum Optical Breadboards specifications (Figure 69), we realized that the breadboard has M6 tapped holes with a spacing of 25 mm in both perpendicular directions. Figure 30 shows the new base design allowing the correct screw driving to the optical breadboard. The box for the actuator (10-1M3A), that allows the waist rotation, and the box for the electrical components (10-Box), were also adapted to screw to the breadboard.

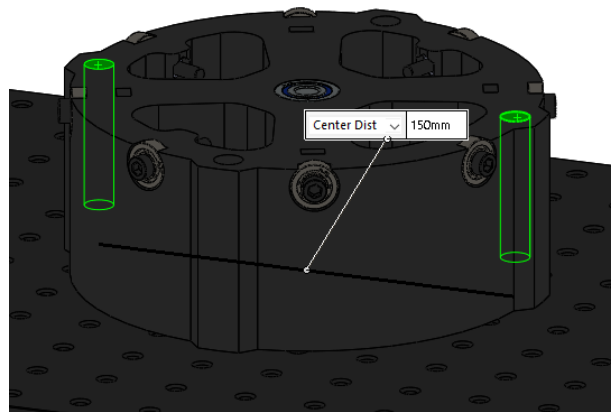


Figure 30 - Base (10-1M2A) assembled on ThorLabs Solid Aluminum Optical Breadboards.

The rotational movement of the waist robot is performed with the transmission of power from an actuator to the rotary plate (10-1M1B), through a T5 belt. As shown in Figure 31, the rotary plate rotates on top of the base due to use of bearings system, a cheap and low friction option. The rotational movement is guided by a screw, which acts as a shaft, and bearings, to reduce the friction.

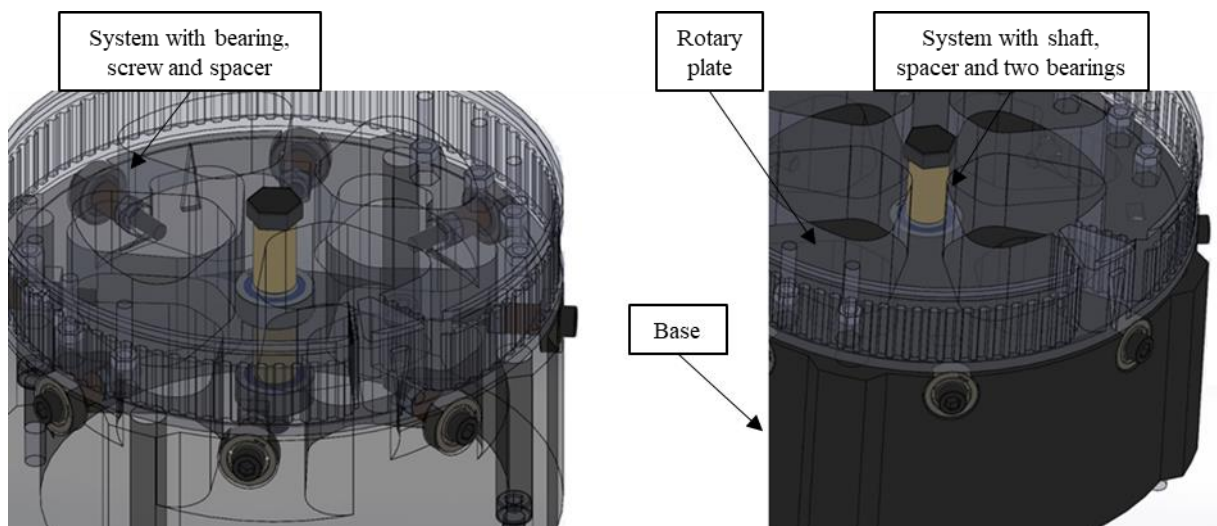


Figure 31 - Robot waist with description of used systems.

3.3.2 Module Shoulder

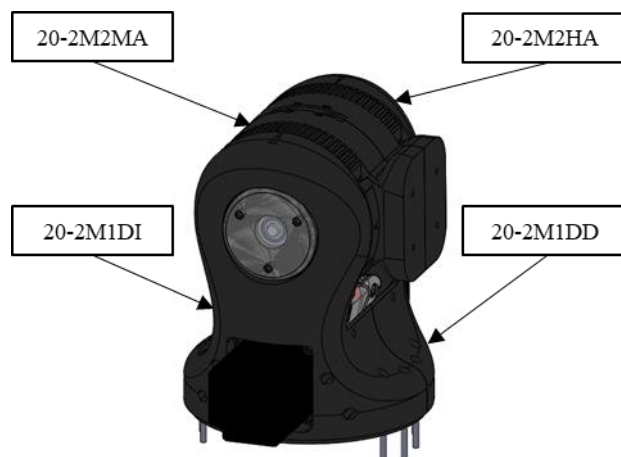


Figure 32 - Module shoulder drawing.

Figure 32 presents all parts on module shoulder. At first, the 3D printed part should be fabricated in the department 3D printer, with a maximum building size of 294 x 192 x 148.6 mm; but as one part was bigger than that, it was necessary to divide it in two (Figure 33). The solution was to attach both parts (20-2M1DI and 20-2M1DD) with four screws and locknuts and to fixate the part to the rotary plate that had a boss, to guide the two parts, using the same technique. With the implementation of these changes it is possible to assure that the robot will work as planned. This module is screwed to the rotary plate; the connection to module elbow is explained after.

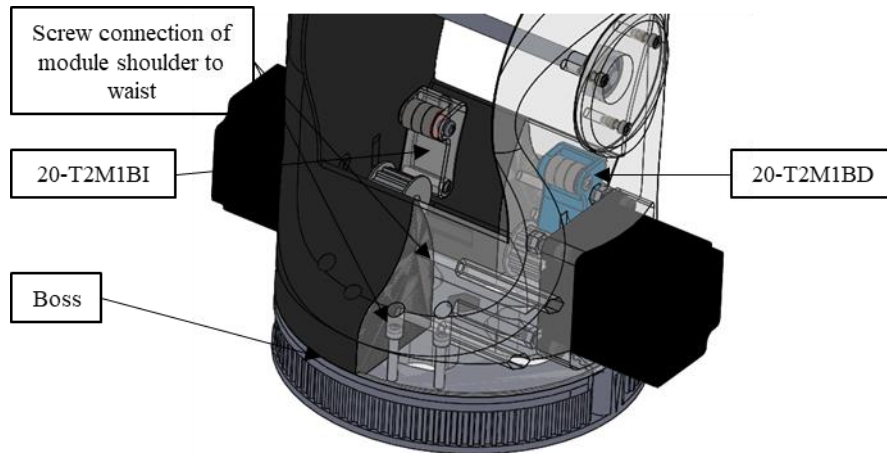


Figure 33 - Module shoulder with the two parts divided and respective connections.

To perform the necessary rotational motion, it is necessary to transfer the shaft movement, which is accomplished with a two pulleys system, one on the stepper and the other in the printed part, which rotates, being both connect by a belt, as shown in Figure 34. It is also used adjustable tensors (20-T2M1BI and 20-T2M1BD) to maintain the pulley stretched, so that clearances are reduced. This type of system is easier to use and it wears it down less, when compared with a gear system.

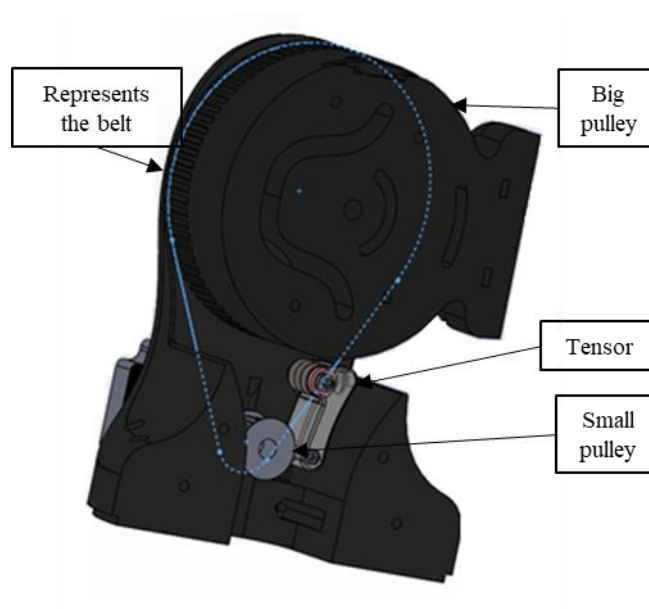


Figure 34 - Transmission system drawing, with the parts identified.

3.3.3 Module Elbow

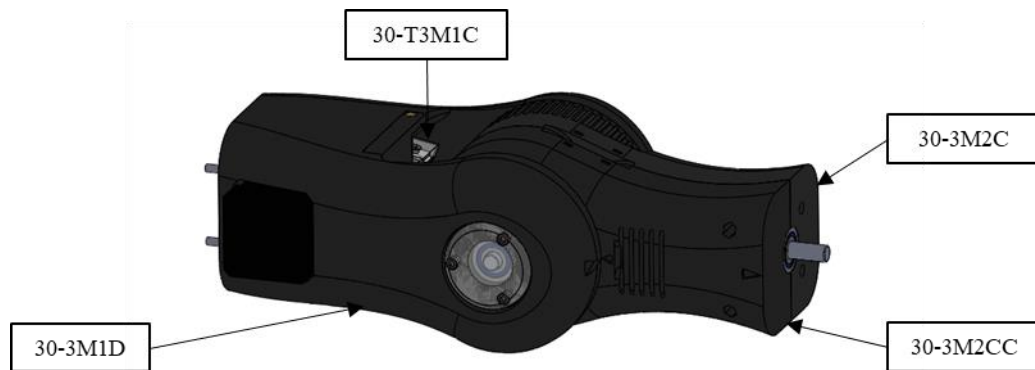


Figure 35 - Module elbow drawing.

Figure 35 presents all parts on module elbow. The rotational movement between both parts is accomplished using the same transmission system of the module shoulder, two pulleys, an adjustable tensor (30-T3M1C) and a T5 belt. In this case, the actuator has a 5:1 gearbox to improve the available torque at the bigger pulley. This module is attached to module shoulder, using screws and locknuts. The connection to module wrist is described after.

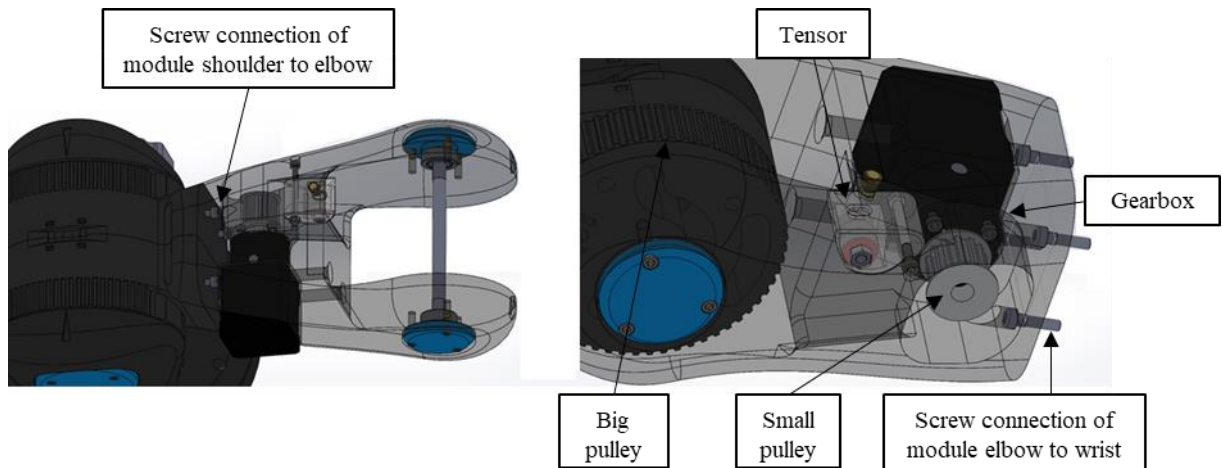


Figure 36 - Module elbow connections.

In Figure 37, the printable parts to build a 5:1 planetary gearbox are presented. It is also necessary a bolt to use as shaft, screws and nuts to screw the bearings and planets to the carrier.

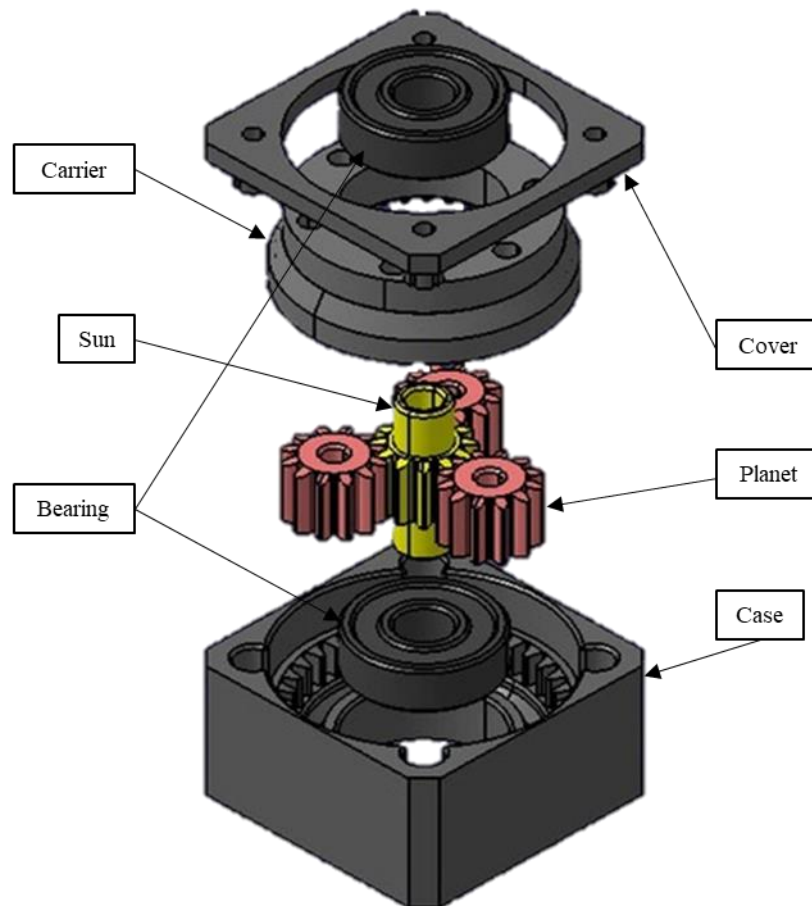


Figure 37 - 5:1 gearbox parts to 3D print.

3.3.4 Module Wrist

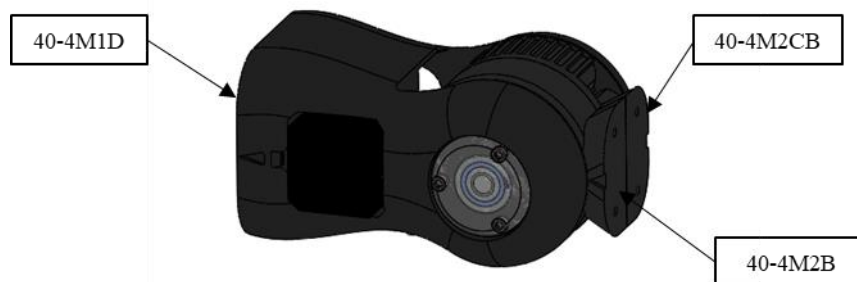


Figure 38 - Module wrist drawing.

Figure 38 presents all parts on module wrist. The transmission system used to rotate this module was a coupling system due to the need of transferring power from the stepper shaft to a larger shaft (Figure 39). The deterrent of using directly the stepper shaft was its length. This system uses two bearings to reduce the friction and a locknut to prevent the displacement of the bearings and shaft. The actuator is assembled inside module elbow. The rotational movement between both parts is accomplished using the same transmission system of the module shoulder, two pulleys, a tensor and a T5 belt.

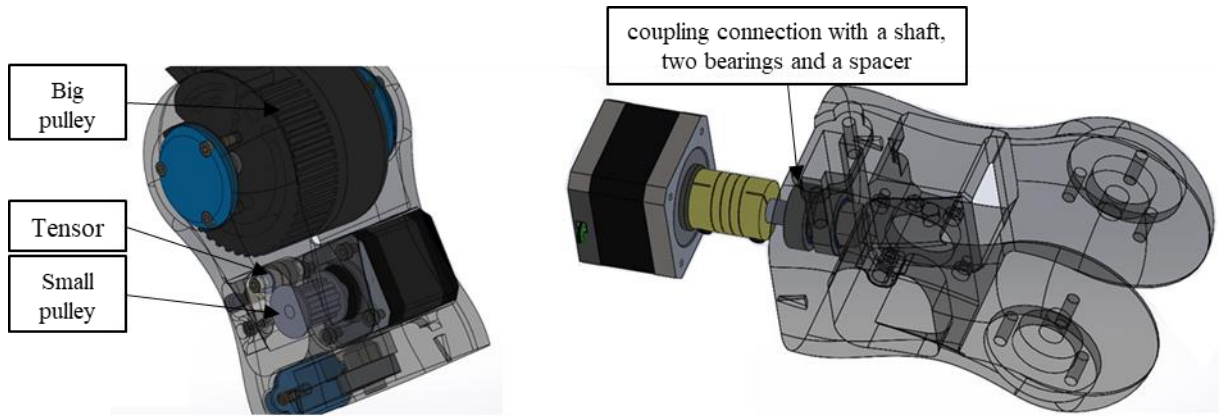


Figure 39 - Module wrist transmission systems.

3.3.5 End Effector

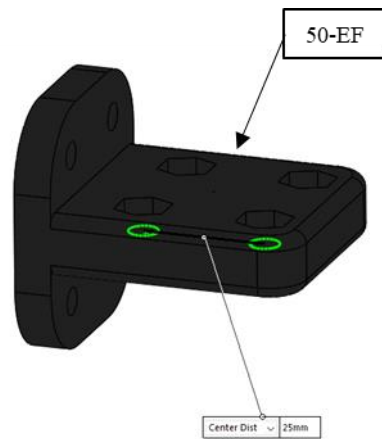


Figure 40 - End effector base drawing.

Figure 40 presents the end effector where the measuring device is attached. The last modification on the design was the end effector base (50-EF). Previously, it had a gripper and needed a change to be possible to perform the intended tasks. The main goal was to perform optical measurements; so, it was necessary to assemble some optical devices (Thorlabs equipment) to a compatible base; in order to do that, it is necessary to screw a base with tapped holes, with a spacing of 25 mm in both directions.

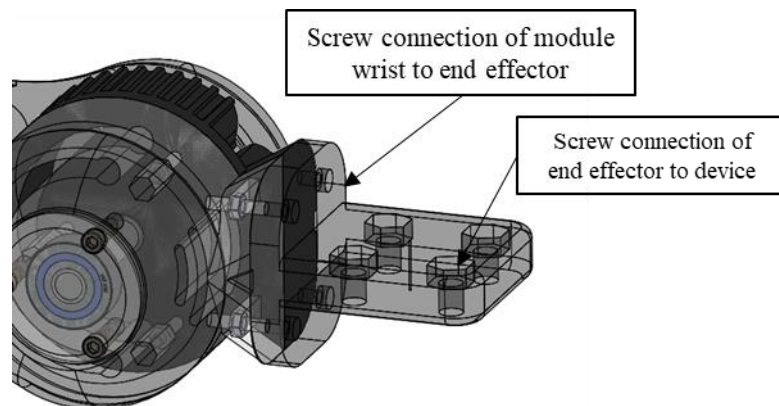


Figure 41 - End effector base connections.

3.4 Calculations

With the robotic arm designed, it was necessary to assess if the stepper motors, already available, could perform the required tasks, regarding torque and dimensions. It was also needed to order the missing stepper motors and power supply. The following calculations validate all the chosen equipment.

3.4.1 Stepper Motors

Stepper motors are DC motors that move in equal steps. The motor shaft rotates when a phase is energized; multiple coils, organized in groups constitute a phase (URL16). If the goal is accurate positioning, speed control or low speed torque are a good option; however, they have low efficiency, no feedback and limited high-speed torque.

Motion in this project is accomplished with six stepper motors, chosen according to the calculations made and the available stock. We had two Lin Engineering 4118M-01RO and two Nanotec ST5709S1208-B, already available, that needed validation. The remaining stepper motors needed to be bought.

1. Stepper Motors Torque

A stepper or steppers transfer torque at a joint to overcome the resistance to move, around the rotational axis, of the subsequent links (Serway and Jewett 2018). That resistance can occur due to gravity and inertia effects. To perform the calculations, the following assumptions have been made:

- The weight of each link is located along the center axis;
- The load is evenly distributed around the center;
- The length of each segment is measured from one center point to the next;
- The actual material properties, moment of inertia and mass of each part are estimated using SolidWorks “mass properties” evaluation.

Regarding gravity effect, it is an experimental evidence that all bodies fall with the same acceleration in a uniform gravitational field. Due to the fact that all bodies in planet Earth are subjected to that effect, every part of the robot should be analyzed, considering that the robot should be able to overcome this effect (Hartle 2003).

Torque is defined as the turning or twisting “force” and is calculated using the relation:

$$\tau_g = F \times L \quad (3.1)$$

where: τ_g is the torque, F the force that acts on an arm point and L the perpendicular distance between the rotation point and the point of application of F .

The force F represents and is defined as:

$$F = m \times g \quad (3.2)$$

where: F is the force that acts on an arm point, m the mass and g the gravitational acceleration.

The resistive torque on a joint due to gravity effect depends on the robot’s position; so it is necessary to assume the worst-case scenario. The highest torque occurs when the arm is completely stretched out horizontally (Corke 2017).

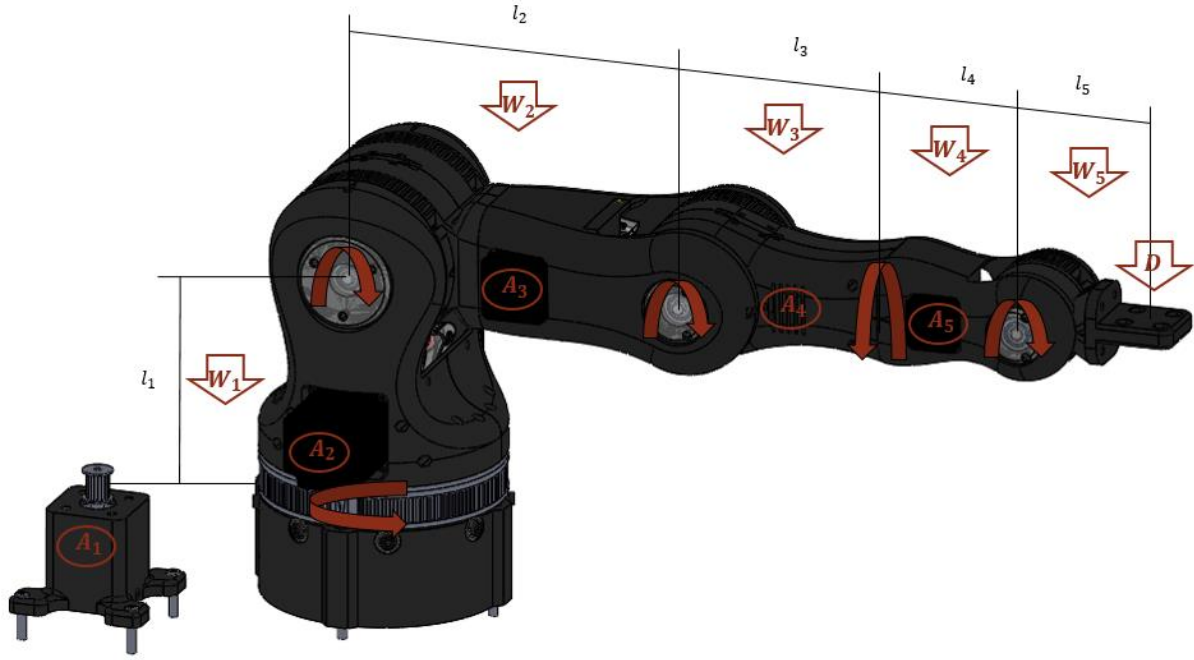


Figure 42 - Robotic arm completely stretched out. A_i -represents the actuators; W_i -represents the weight of each link; l_i -distance between joints; D -represents the weight of a device.

In Figure 42 it is possible to see the position of each actuator, numbered according to the joint that they move (each actuator has an associated weight), the distance between joints, the link's weights and the weight of the device attached on the end effector. The joints torque calculation was made assuming a load of 500 g at the end effector; that value is ten times higher than the weight for the typical measuring device used in the optomechanical laboratory, a Flir Flea3 camera (Figure 72 in Appendix A). For the links weights we resort to SolidWorks "mass proprieties" evaluation (Appendix C); distances were also evaluated in SolidWorks.

The calculation of required torque for each joint due to gravity effect is as follows:

$$\tau_{5g} = \frac{l_5}{2} \times W_5 + l_5 \times D \quad (3.3)$$

$$\tau_{4g} = 0 \quad (3.4)$$

$$\begin{aligned} \tau_{3g} = & \frac{l_3}{2} \times W_3 + \frac{l_3}{2} \times A_4 + \left(l_3 + \frac{l_4}{2}\right) \times W_4 + \left(l_3 + \frac{l_4}{2}\right) \times A_5 \\ & + \left(l_3 + l_4 + \frac{l_5}{2}\right) \times W_5 + (l_3 + l_4 + l_5) \times D \end{aligned} \quad (3.5)$$

$$\begin{aligned} \tau_{2g} = & \frac{l_2}{2} \times W_2 + \frac{l_2}{2} \times A_3 + \left(l_2 + \frac{l_3}{2}\right) \times W_3 + \left(l_2 + \frac{l_3}{2}\right) \times A_4 \\ & + \left(l_2 + l_3 + \frac{l_4}{2}\right) \times A_5 + \left(l_2 + l_3 + l_4 + \frac{l_5}{2}\right) \times W_5 \\ & + (l_2 + l_3 + l_4 + l_5) \times D \end{aligned} \quad (3.6)$$

$$\tau_{1g} = 0 \quad (3.7)$$

where: τ_{ig} is the required torque for the joint i , l_i the link i length, W_i the link i weight, D the end effector device weight and A_i the actuator(s) i weight.

Table 2 summarizes the necessary data and presents the results regarding the necessary torque at each joint due to gravity.

Table 2 - Torque due to gravity.

Joint	l_i - Length (mm)	W_i - Weight (N)	A_i - Load (N)	τ_{ig} - Torque (N·m)
1	152.750	17.484	13.734	0.000
2	221.330	19.225	19.620	11.648
3	127.000	7.001	5.886	3.815
4	95.510	3.064	2.894	0.000
5	98.500	2.114	2.894	0.587

The inertia effect is the tendency of an object to resist any attempt change of its rotational velocity (Serway and Jewett 2018). The torque, due to this effect, is calculated using the following equation:

$$\tau_I = I \times \alpha \tag{3.8}$$

where: τ_I is the torque, I the moment of inertia and α the angular acceleration.

The moment of inertia measures the resistance of an object to change its rotational motion (Serway and Jewett 2018); in this case, the values were estimated through SolidWorks “mass properties” evaluation that uses a specific axis at the mass center. As the rotational movement is made around the joint rotational axis; so it was needed to calculate the moments of inertia for those axes in the mass center; this was made using the parallel-axis theorem (Figure 43):

$$I = I_{CM} + m \times h^2 \tag{3.9}$$

where: I is the moment of inertia about the rotation axis, I_{CM} the moment of inertia about an axis through the center of mass, m the mass and h the perpendicular distance between axis.

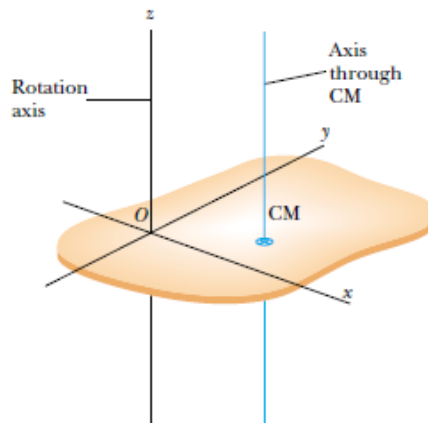


Figure 43 - Perspective drawing showing the z axis (the axis of rotation) and the parallel axis through the mass center (Serway and Jewett 2018).

The calculation of moment of inertia for each link, after analyzing the correct axis of rotation, is as follows:

$$I_1 = I_{xx1} + m_1 \times h_1^2 \tag{3.10}$$

$$I_2 = I_{yy2} + m_2 \times h_2^2 \tag{3.11}$$

$$I_3 = I_{yy3} + m_3 \times h_3^2 \tag{3.12}$$

$$I_4 = I_{xx4} + m_4 \times h_4^2 \tag{3.13}$$

$$I_5 = I_{zz5} + m_5 \times h_5^2 \tag{3.14}$$

where: I_i is the link i moment of inertia, I_{CMi} the link i moment of inertia about the axis through the center of mass, m_i the link i mass and h_i the link i distance between axis.

Table 3 - Moment of inertia.

Joint	I_{xx} - Inertia (g·mm ²)	I_{yy} - Inertia (g·mm ²)	I_{zz} - Inertia (g·mm ²)	m_i - mass (g)	h_i - distance (mm)	I_i - Inertia (g·mm ²)
1	505091052.4	31613387.8	485007633.6	9402.5	240.000	1046676780.4
2	5929707.5	174361142.2	173633385.5	5117.7	250.000	494218017.2
3	1953742.6	37124613.8	37211566.7	3058.3	170.000	125509772.8
4	1020791.9	7139050.8	7089030.5	1874.4	120.000	28011575.9
5	326133.5	926337.9	1020747.3	1221.1	75.000	7889378.5

The angular acceleration for every joint that we chose was 30°/s² to fulfil the needs.

Table 4 - Torque due to inertia.

Joint	α - Angular acceleration (°/s ²)	α - Angular acceleration (rad/s ²)	I_{CM} - Inertia (g·mm ²)	τ_I -Torque (N·m)
1	30.000	0.524	1046676780.430	0.548
2	30.000	0.524	494218017.160	0.259
3	30.000	0.524	125509772.840	0.066
4	30.000	0.524	28011575.900	0.015
5	30.000	0.524	7889378.530	0.004

The total torque required at each joint will be the sum of both torques, due to gravity and inertia. The result is presented in Table 5.

Table 5 - Total torque at each joint.

Joint	τ_g - Torque due to gravity (N·m)	τ_I - Torque due to inertia (N·m)	τ - Total torque (N·m)
1	0.000	0.548	0.548
2	11.648	0.259	11.907
3	3.815	0.066	3.880
4	0.000	0.015	0.015
5	0.587	0.004	0.591

2. Steppers Transmission

To provide motion to the joints it is necessary a transmission to transfer the power from the actuators. To achieve that, we will resort to a coupling, to connect a 5 mm shaft to an 8 mm, for joint 4. In the other four joints, the transmission will occur, using belts with pulleys with different dimensions, as shown before. The use of a transmission system increases the torque, but decreases the velocity. Figure 44 shows a flat belt transmission scheme, with the important variables to analyze it correctly.

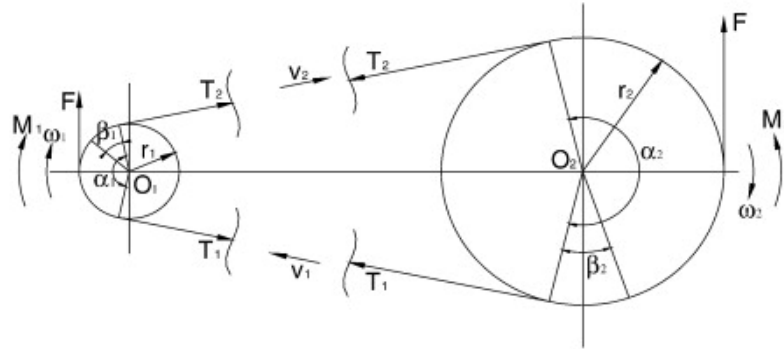


Figure 44 - Scheme of flat belt transmission (Della Pietra and Timpone 2013).

The “Euler model”, published in 1762, is the simplest theoretical model regarding belt drives (Della Pietra and Timpone 2013). It states that the tensions at the ends of a stretched rope wrapped around a body in sliding limit conditions satisfies the following law:

$$\frac{T_1}{T_2} = e^{\mu\alpha} \quad (3.15)$$

where: T_1 is the tension at the rope end, T_2 the tension at other the rope end, μ the friction coefficient and α the contact angle.

The contact angle is calculated using the following equation:

$$\alpha = \pi - 2 \times \sin^{-1} \left(\frac{r_1 - r_2}{C} \right) \quad (3.16)$$

where: α is the contact angle, r_1 the radius of pulley 1, r_2 the radius of pulley 2 and C the distance between centers.

Rubber coated nylon flat belts running on pulleys display, in practice, friction coefficients between 0.3 and 0.8 (Childs 1980); so, we used friction coefficient of 0.5. Table 6 presents all the needed dimensional data and the ratio between the diameters of pulleys connected by a belt.

Table 6 - Data for transmission calculations.

Transmission	Ratio $\left(\frac{r_1}{r_2}\right)$	r_1 (mm)	r_2 (mm)	C (mm)	μ	α_1 (rad)
1	10.533	7.500	79.000	184.560	0.500	2.346
2	5.455	11.000	60.000	101.800	0.500	2.137
3	4.318	11.000	47.500	112.610	0.500	2.481
5	4.667	7.500	35.000	57.350	0.500	2.141

The arcs $r_1 \times \beta$ and $r_2 \times \beta$ on the booth pulleys are called active arcs, because friction forces occur along them, giving rise to the change in belt tension, which corresponds to the transmitted torque. The torque applied to both pulleys, placing $F = T_1 - T_2$, is calculated as follows:

$$M_1 = (T_1 - T_2) \times r_1 \tag{3.17}$$

$$M_2 = (T_1 - T_2) \times r_2 \tag{3.18}$$

where: M_i is the torque of pulley i , T_i the tension at the rope end and r_i the radius of pulley i .

The torque on the stepper motors already available at Bosch Car Multimedia is known from the data sheet, so they only needed validation. The rest of the stepper motors were chosen according to the torque needs and dimensions. Considering equation 3.17 and 3.18, the torque available in the bigger pulley (M_2) was calculated for each joint.

The result and the model of the stepper motors for each joint is presented in Table 7.

Table 7 - Torque transmission.

Joint	Quantity	Stepper model	M_1 (N·m)	M_2 (N·m)
1	1	42BYGHW811	0.480	5.056
2	2	ST5709L1108-B	2.700	14.727
3	1	4118M-01 w/5:1 PG	2.200	8.636
4	1	103H5205-5240	0.265	0.265
5	1	4118M-01	0.440	2.053

The comparison between the available torque and the required one is presented in Table 8. It is possible to ensure that with the choices we made. we will be able to perform the required tasks.

Table 8 - Required and available torque comparison.

Joint	Quantity	Stepper	Required Torque (N·m)	Available Torque (N·m)	
1	1	42BYGHW811	0.548	5.056	
2	2	ST5709L1108-B	11.907	14.727	
3	1	4118M-01	3.880	8.636	
4	1	103H5205-5240	0.015	0.265	
5	1	4118M-01	0.591	2.053	

3.4.2 Power Supply

The power supply is responsible to deliver electric power to the robot components. To choose correctly, it is necessary to calculate the power of each component and the necessary current. The necessary current for each component is in the data sheet (Appendix C). In Portugal, the standard voltage for electrical networks is 220 V; therefore, the power supply needs to convert 220 VAC to 24 VDC, which was our choice for the output voltage on the power supply.

The needed power is calculated using the following equation (Alexander and no Sadiku 2000):

$$P = V \times I \quad (3.19)$$

where: P is the power, V the voltage and I the current.

Table 9 presents the data and results, regarding the needed power for each and the total components.

Table 9 - Components needs, regarding power supply.

Component	Quantity	<i>I</i> - Current (A)	<i>P</i> - Power (W)
4118M-01	2	1,7	40,80
ST5709L1108-B	2	1,1	26,40
42BYGHW811	1	2,5	60,00
103H5205-5240	1	1	24,00
Arduino Mega	1	0,5	12,00
Total	7	9,6	230,40

We have selected a 10 A, 24 V power supply with 250 W (Figure 45). Since the Arduino Mega microcontroller maximum input tension is 20 V, it was necessary a power supply that fulfilled that need; so, we used another power supply, an adjustable one, with a maximum output voltage of 16 V. The option of selecting a 24 V power supply instead of a 12 V was not to lose performance regarding the stepper motors.



Figure 45 - Power supply used in this project.

3.5 Summary

This chapter describes the process used to choose the base project for the prototype redesign and the actual redesign to fit the requirements and constraints. The use of an open source project allowed a fast development because of the available resources. The validation and choice of the components that provide motion was also performed and the available material at Bosch Car Multimedia was approved.

4 Motion Control

In this chapter the motion of the prototype is analyzed. The kinematics and dynamic model, as well as the control and electronics are, presented and simulated.

4.1 Kinematics

4.1.1 Direct Kinematics

As stated by kinematics is the science of motion without regard to the forces that cause it (Craig John 1989). The Denavit-Hartenberg notation is the most common systematic method to describe a serial-link mechanism geometry; Jacques Denavit and Richard S. Hartenberg proposed it in 1955. Due to the complex geometry of a manipulator, we will affix frames to the different parts of the mechanism and then describe the relationships between them. The study of manipulator kinematics involves, among other things, how the locations of the different frames change as the manipulator articulates. We will follow a method to compute the position and orientation of the manipulator's end-effector relative to the base of the manipulator as a function of the joint variables. Usually, frames can be arbitrarily defined, as long as they are attached to the link they are referred to. Nevertheless, it is convenient to follow some rules for the definition of the link frames (Siciliano et al. 2009).

1. The Standard Denavit-Hartenberg Notation

The relation between two coordinate frames has six parameters, three translations and three rotations, but with the Denavit-Hartenberg notation, it is possible to use only four parameters, due to coordinate frames positions. The first rule is that the x-axis from frame i needs to intercept z-axis from frame $i - 1$; the second constrain is that the x-axis from frame i is perpendicular to the z-axis from frame $i - 1$ (Corke 2017).

Regarding the standard Denavit-Hartenberg notation, a manipulator with N joints (numbered from 1 to N) has $N + 1$ links (numbered from 0 to N); 0 is the base and N is the end effector. Joint i connects the link $i - 1$ to the link i , so joint i moves link i . As mentioned before, a link can be defined by two parameters; its length a_i and its twist α_i . Joints can be defined also by two parameters; the link offset d_i , is the distance from one link coordinate frame to the next, through the axis of the joint, and the joint angle θ_i , that is the rotation of one link coordinate frame with respect to the next, through the joint axis (Corke 2017).

Table 10 - Denavit-Hartenberg parameters.

θ_i (°)	The angle between x_{i-1} and x_i axes about the z_{i-1} axis
d_i (mm)	The distance from the origin of frame $i - 1$ to the x_i and z_{i-1} axis interception along the z_{i-1} axis
a_i (mm)	The distance from z_{i-1} and x_i axis interception to the frame i origin along x_i axis
α_i (°)	The angle from z_{i-1} axis to z_i axis about x_i axis

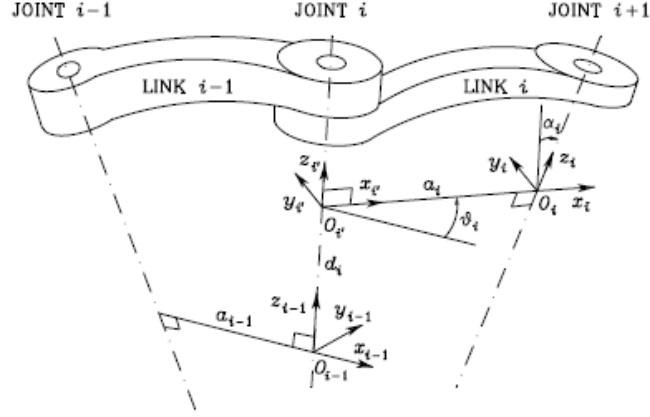


Figure 46 - Standard Denavit-Hartenberg link parameters (Corke 2017).

As shown in Figure 46, the coordinate frame i is attached to the far end of link and the i joint axis is aligned with the z -axis. These links and joint parameters are known as Denavit-Hartenberg parameters.

The transformation from link coordinate frame $i - 1$ to frame i is defined in terms of elementary rotations and translations as ${}^{i-1}A_i(\theta_i, d_i, a_i, \alpha_i) = T_{Rz}(\theta_i)T_z(d_i)T_x(a_i)T_{Rx}(\alpha_i)$; it can be expanded as:

$${}^{i-1}A_i = \begin{pmatrix} \cos\theta_i & -\sin\theta_i\cos\alpha_i & \sin\theta_i\sin\alpha_i & a_i\cos\theta_i \\ \sin\theta_i & \cos\theta_i\cos\alpha_i & -\cos\theta_i\sin\alpha_i & a_i\sin\theta_i \\ 0 & \sin\alpha_i & \cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (3.20)$$

For an open-chain manipulator with $i + 1$ links and i joints, where link 0 is the base fixed to the ground, it is assumed that each joint provides one degree of freedom to the structure, corresponding to the joint variable.

The construction of an operating procedure for the computation of direct kinematics, naturally, derives from the typical open kinematic chain of the manipulator structure. Since each joint connects two consecutive links, it is reasonable to consider first the description of kinematic relationship between consecutive links and then to obtain the overall description of manipulator kinematics in a recursive way. To achieve this, it is worth defining a coordinate frame attached to each link, from link 0 to link N . Then, the coordinate transformation describing the position and orientation of frame N with respect to frame 0 is given by:

$$T_N^0(q) = A_1^0(q_1)A_2^1(q_2) \dots A_N^{N-1}(q_N) \quad (3.21)$$

As requested, the computation of the direct kinematics function is recursive and is obtained in a systematic manner, by simple products of the homogeneous transformation matrices $A_i^{i-1}(q_i)$ (for $i = 1, \dots, N$), each of which is a function of a single joint variable.

With reference to the direct kinematics equation, the actual coordinate transformation describing, the position and orientation of the end effector frame with respect to the base frame, can be obtained as:

$$T_e^b(q) = T_0^b T_N^0(q) T_e^N \quad (3.22)$$

where: T_0^b is the position and orientation of frame 0 with respect to the base frame and T_e^N the position and orientation of the end-effector frame with respect to frame N .

2. Prototype Denavit-Hartenberg Parameters

As stated before, to properly develop and control a robotic manipulator it is important to determinate the Denavit-Hartenberg parameters. The first step to achieve that is to do a correct frame assignment. The frame assignment for the prototype is represented in Figure 47. The process to assign the frames was the following:

- Establish the base frame. Set the origin on the support base, z_0 axis coincident with joint 1 axis. The x_0 and y_0 axes are chosen conveniently to form a right-hand frame.
- Establish joints axis. Axis z_1 coincident with joint $i + 1$ axis.
- Establish the frame i origin. The origin of frame i is in the interception between z_j and z_{i-1} axis. If z_i and z_{i-1} axis are parallel, the origin is in the interception of z_i axis and the common normal of z_{i-1} and z_i axis.
- Establish x_i . Along the common normal between z_{i-1} and z_i through O_i .
- Establish y_i to complete a right-hand frame.
- Establish the end-effector frame. Establish x_i , so that it is perpendicular to z_{i-1} (intercepting it). If the end effector joint is revolute, align z_i with z_{i-1} . Set y_i conveniently to form a right-hand frame.
- Perform these steps for each frame.

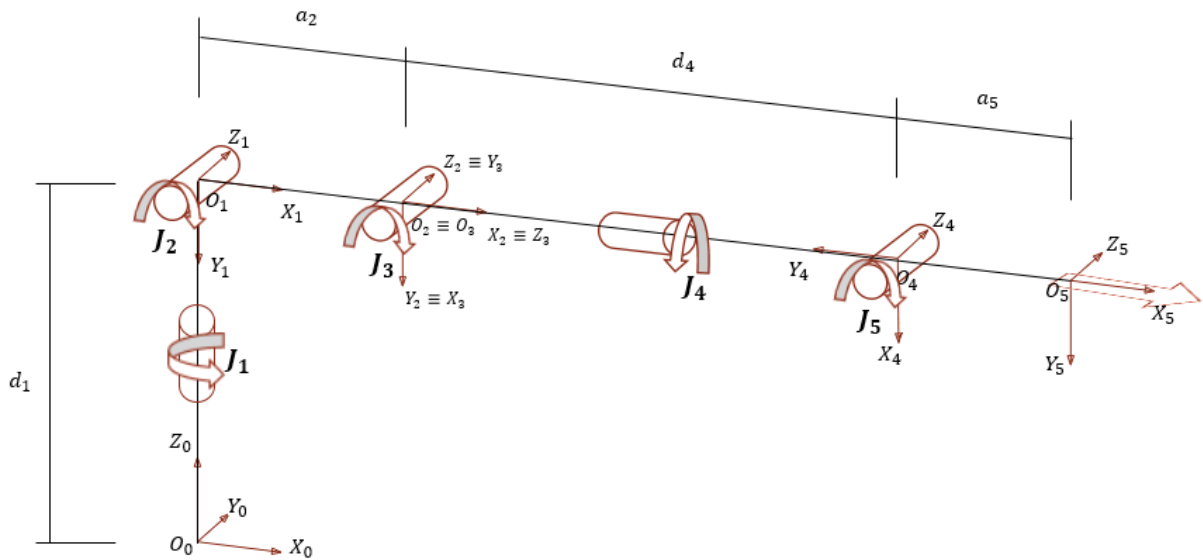


Figure 47 - Frame assignment for the prototype.

According to Table 10, the Denavit-Hartenberg parameters for the prototype were determined and presented on Table 11.

Table 11 - Prototype D-H parameters.

Joint	θ_i (°)	d_i (mm)	a_i (mm)	α_i (°)
1	$0+\theta_1$	d_1	0	-90
2	$0+\theta_2$	0	a_2	0
3	$90+\theta_3$	0	0	90
4	$0+\theta_4$	d_4	0	-90
5	$-90+\theta_5$	0	a_5	0

4.1.2 Inverse Kinematics

After solving the direct kinematics problem, it is necessary to work out the inverse kinematics problem, the determination of the joints variables corresponding to a given end effector position and orientation. The solution to this problem is crucial, in order to transform the motion specifications, assigned to the end effector in the operational space, into the corresponding joint space motions that allow execution of the required motion (Siciliano et al. 2009).

Regarding the direct kinematics, once the joint variables are known, the end effector position and rotation matrix are computed in a unique way. On the other hand, the inverse kinematics problem is more elaborated because of the following reasons:

- The equations to solve are in general nonlinear, and thus it is not always possible to find a closed-form solution.
- Multiple solutions may exist.
- Infinite solutions may exist, e.g., in the case of a redundant manipulator.
- There might be no admissible solutions, in view of the manipulator kinematic structure.

The only way to guarantee a solution is if the given end effector position and orientation belong to the manipulator dexterous workspace.

Regarding the problem of multiple solutions, it depends not only on the number of degrees of freedom, but also on the number of non-null Denavit-Hartenberg parameters; usually, the greater the number of non-null parameters, the greater the number of admissible solutions. Such occurrence demands some criterion to choose among admissible solutions. The existence of mechanical joint limits may, eventually, reduce the number of admissible multiple solutions for the real structure.

Computation of closed-form solutions requires, either, algebraic intuition to find those significant equations containing the unknowns or geometric intuition to find those significant points on the structure, with respect to which it is convenient to express position and/or orientation, as a function of a reduced number of unknowns. If a closed-form solution is hard to find, it might be appropriate to resort to numerical solution techniques; these, clearly have the advantage of being applicable to any kinematic structure, but, generally, they do not allow computation of all admissible solutions. Another solution is to use suitable algorithms, utilizing the manipulator Jacobian to solve the inverse kinematics problem (Siciliano et al. 2009).

4.1.3 Differential kinematics

The goal of the differential kinematics is to find the relationship between the joint velocities and the corresponding end-effector linear and angular velocities. A matrix, termed geometric Jacobian, describes this mapping, depending on the manipulator configuration. If the end-effector pose is expressed with reference to a minimal representation in the operational space, it is possible to compute the Jacobian matrix through differentiation of the direct kinematics function with respect to the joint variables. The resulting Jacobian, termed analytical Jacobian, in general differs from the geometric one. The Jacobian is one of the most important tools for manipulator characterization; in fact, it is useful for finding singularities, analyzing redundancy, determining inverse kinematics algorithms, describing the mapping between forces applied to the end-effector, resulting torques at the joints (statics), deriving dynamic equations of motion and designing operational space control schemes (Siciliano et al. 2009).

The relationship between joint velocity and Cartesian velocity is calculated as:

$$v_e = J(q)\dot{q} \quad (3.23)$$

where: v_e is the Cartesian velocity, $J(q)$ the velocity Jacobian and \dot{q} the joint velocity.

4.2 Dynamic model

With the kinematics studies we focused in static positions, static forces and velocities, but it is also necessary to take in consideration the necessary forces to cause motion; dynamic models describe this relationship. The derivation of the dynamic model of a manipulator is important to simulate the motion, analyze the manipulator structure and the design of control algorithms. To simulate the manipulator motion allows control strategies and motion planning techniques that can be tested without the need of a physical system (Siciliano et al. 2009).

There are several methods for generating the dynamic model equations. A conceptually simple and systematic method is the Lagrange formulation. Another method is the Newton-Euler formulation, which is computationally more efficient, since it exploits the typically open structure of the manipulator kinematic chain. Both formulations allow the computation of the relationship between joint torques and the motion of the structure (Siciliano et al. 2009).

The advantages of the Lagrange formulation are the following:

- It is systematic and of immediate comprehension;
- It provides the equations of motion in a compact analytical form containing the inertia matrix, the matrix in the centrifugal and Coriolis forces, and the vector of gravitational forces. Such a form is advantageous for control design;
- It is effective if we wish to include mechanical effects with more complexity, such as flexible link deformation.

The advantage of the Newton–Euler formulation is the following:

- It is an inherently recursive method that is computationally efficient.

Regarding the dynamics of a manipulator, it is important to find a solution to two kinds of problems, computation of direct dynamics and inverse dynamics.

4.3 MATLAB Simulation

To graphically simulate the prototype kinematics and work space we have chosen MATLAB. This allowed the possibility of using toolboxes that are focused in particular topics, such as robotics. We resort to a toolbox developed by Peter Corke (Corke 2017), which provides many functions, useful to study and simulate classical arm-type robotics, such as kinematics,

dynamics, and trajectory generation. This toolbox is based on a general method of representing the kinematics and dynamics of serial-link manipulators, the Denavit-Hartenberg notation. (Corke 2017).

4.3.1 Direct Kinematics

To simulate the direct kinematics, the end effector position and orientation as function of the joints variables, and create a kinematic model, it is necessary to determinate the Denavit-Hartenberg parameters. The parameters for this prototype have already been determined, and Table 12 presents them.

Table 12 - Denavit-Hartenberg parameters for the MATLAB simulation.

Joint	θ_i (°)	d_i (m)	a_i (m)	α_i (°)
1	$0 + \theta_1$	0.232	0	$-\frac{\pi}{2}$
2	$0 + \theta_2$	0	0.221	0
3	$\frac{\pi}{2} + \theta_3$	0	0	$\frac{\pi}{2}$
4	$0 + \theta_4$	0.223	0	$-\frac{\pi}{2}$
5	$-\frac{\pi}{2} + \theta_5$	0	0.099	0

The steps followed in MATLAB were:

- Create a Denavit-Hartenberg parameters matrix (Figure 48);

```
>> dh = [
0 0.232 0 -pi/2
0 0 0.221 0
pi/2 0 0 pi/2
0 0.223 0 -pi/2
-pi/2 0 .099 0
]

dh =

     0     0.2320         0    -1.5708
     0         0     0.2210         0
  1.5708         0         0     1.5708
     0     0.2230         0    -1.5708
 -1.5708         0     0.0990         0
```

Figure 48 - Denavit-Hartenberg parameters matrix command window.

- Create a robot object, r , and put it into the workspace variable using the toolbox function, $r = SerialLink(dh)$, with the Denavit-Hartenberg parameters matrix, resulting a serial link object (represents the robot arm) created in the workspace (Figure 49);

```
>> r = SerialLink(dh)
r =
noname (5 axis, RRRRR, stdDH, fastRNE)
```

j	theta	d	a	alpha	offset
1	q1	0.232	0	-1.571	0
2	q2	0	0.221	0	0
3	q3	0	0	1.571	0
4	q4	0.223	0	-1.571	0
5	q5	0	0.099	0	0

```
grav = 0 base = 1 0 0 0 tool = 1 0 0 0
      0      0 1 0 0      0 1 0 0
      9.81 0 0 1 0      0 0 1 0
              0 0 0 1      0 0 0 1
```

Type of joints

Figure 49 - $r = SerialLink(dh)$ command window.

- Create qz matrix that represents the zero joint angles Figure 50, the robot position, as shown in Figure 42;

```
>> qz = [
0 0 pi/2 0 -pi/2
]
qz =
      0      0  1.5708      0 -1.5708
```

Figure 50 - Zero joint angles matrix.

- With function $r.plot(qz)$ it is possible to plot the robot to its zero angle pose, to graphically visualize it (Figure 51);

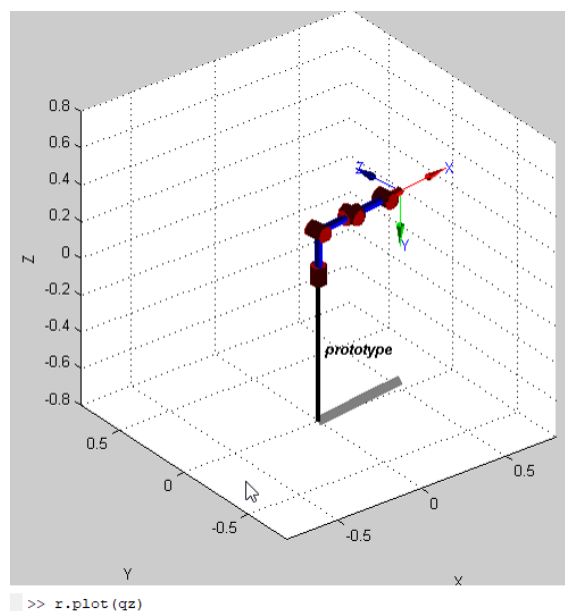


Figure 51 - Prototype plot for the zero joint angles.

- With function $r.teach$, it is possible to simulate a different position, by changing the joint angles (Figure 52);
- With function $r.fkine$, it is possible to obtain the homogeneous transformation matrix that represents the pose of the end effector, editing θ_i (Figure 52).

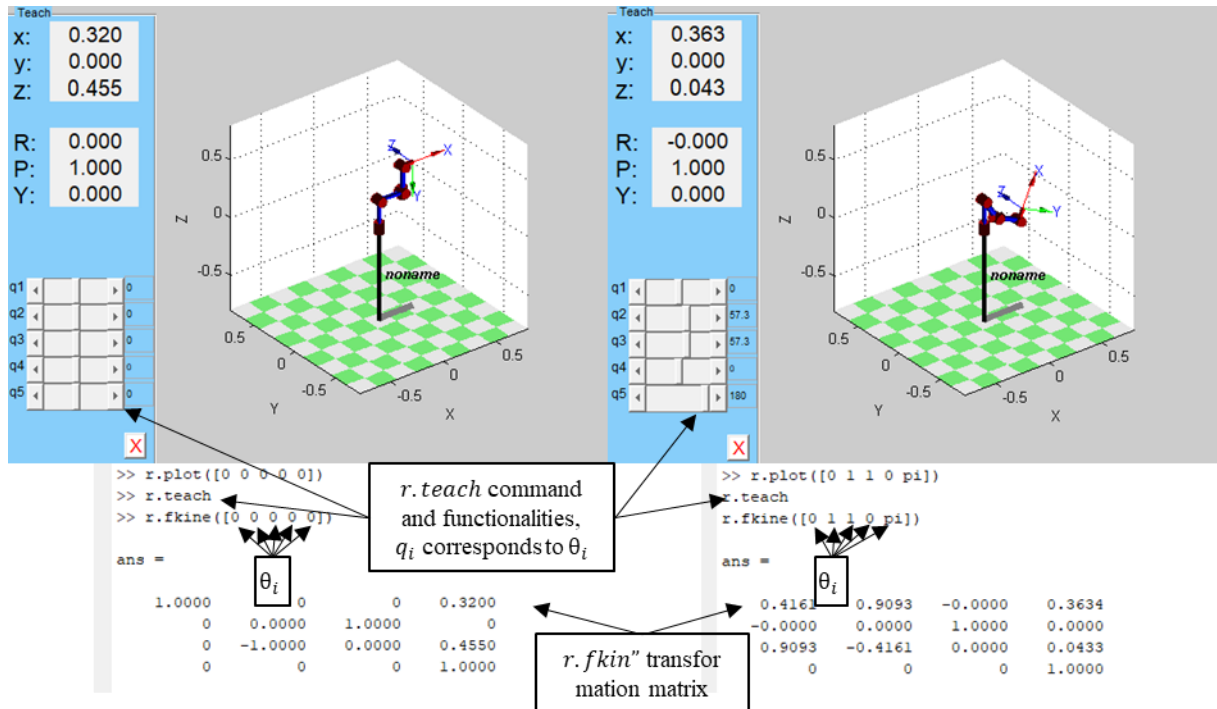


Figure 52 - Different positions simulation.

4.3.2 Inverse Kinematics

To simulate the inverse kinematics, the joints variables as function of the end effector position and orientation, and continue the code made for the direct kinematics, the steps followed were:

- To create the T pose by translating and rotating the end effector position (Figure 53);

```
>> T = transl (0.1, 0.05, 0) * trotx(pi)

T =

    1.0000         0         0    0.1000
         0   -1.0000    -0.0000    0.0500
         0    0.0000   -1.0000         0
         0         0         0    1.0000
```

Figure 53 - End effector position matrix.

- To compute the joint angles required to achieve this end effector configuration. The robot is under actuated, so we need to specify a mask matrix to indicate the degree of freedom that we want to change. We will choose: x, y, z and the orientation of z (Figure 54);

```
>> q=r.ikine(T, qz, [1 1 1 0 0 1])
Warning: Initial joint configuration results in a (near-)singular configuration, this may slow convergence
> In SerialLink.ikine at 156
Warning: solution diverging at step 3, try reducing alpha
> In SerialLink.ikine at 260
Warning: ikine: iteration limit 1000 exceeded (row 1), final err 0.003922
> In SerialLink.ikine at 179

q =

    1.3529    0.4492   -2.9868    1.8892    2.7478
```

Figure 54 - ikine command.

- To check the inverse kinematics, by performing the forward kinematics for these joints. It is possible to check that the translational components are equal (Figure 55).

```
>> r.fkine(q)

ans =

    0.8520    0.2210    0.4747    0.1000
   -0.2092   -0.6874    0.6955    0.0500
    0.4800   -0.6919   -0.5394   -0.0000
         0         0         0         1.0000
```

Figure 55 - Inverse kinematics check.

4.4 Control

4.4.1 Electronics

1. Electronic assembly

Figure 56 presents the schematics of the electronic components and their connections. For the interface between Arduino Mega microcontroller and the stepper motors, we used a RAMPS 1.4 shield with a TB6560 stepper driver that has an output current up to 3A and a supply voltage up to 32V. These components were chosen to easily integrate with the steppers motors; this equipment is widely used in 3D printers.

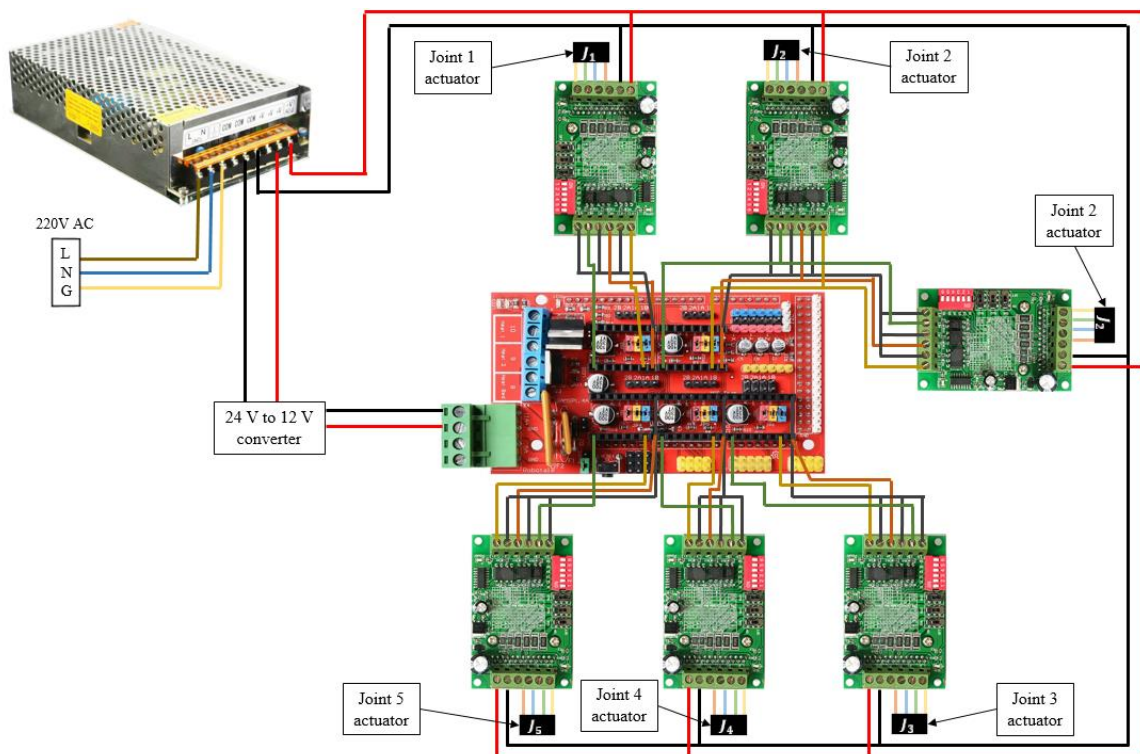


Figure 56 - Electronic scheme.

2. TB6560 configuration

TB6560 driver has four configurable settings (Figure 57):

- Running current: controls how running current goes to the stepper motors;

- Stop current: controls how much current the stepper motors should get when they do not run or when they stay standstill; they need current to hold the position and do not get out of position;
- Excitation mode: controls the micro stepping; a higher micro stepping gives higher resolution but a lower holding torque;
- Decay setting: fast mode allows the current, when the engine stops, to go directly to the power supply. This allows the magnetic field to collapse quickly and the motor will charge afterwards when it stops; this means faster motor speeds. In a slow decay setting, the current from the engine is fed back to itself, this means the magnetic field collapses slowly and the motor stops instantly; this configuration is good not to lose accuracy, but decreases the motor speed.

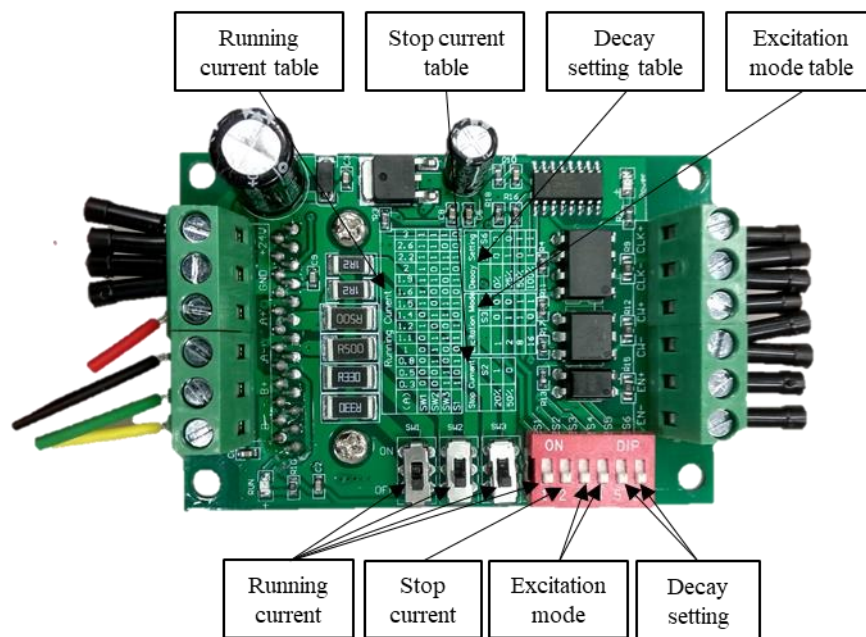


Figure 57 - TB6560 settings.

Table 13 presents the configuration for each TB6560 driver.

Table 13 - TB6560 configuration.

Joint	Stepper	Running current			S1	S2	Excitation mode		Decay setting	
		SW1	SW2	SW3			S3	S4	S5	S6
1	42BYGHW811	1	0	0	0	1	1	0	0	0
2	ST5709L1108-B	0	1	1	1	1	1	0	0	0
3	4118M-01	1	0	0	0	1	1	0	0	0
4	103H5205-5240	0	1	0	0	1	1	0	0	0
5	4118M-01	1	0	0	0	1	1	0	0	0

4.4.2 Firmware

A firmware is a computer software that controls the prototype stepper motors and is installed in the Arduino Mega microcontroller. The Marlin firmware was chosen, because it is an open source firmware that is used in several commercial 3D printers. The control concept is similar for this prototype robot.

It can be used with several microcontrollers and it is possible to change the code, to match the required goals. The code configuration is made in Arduino IDE platform. The control language is a derivative of G-Code, that sends commands regarding positioning to the robot. In a computer with suitable software and an USB connection, it is possible to have an online control; the software used to test the motion of the joints was Pronterface. With it, is possible to move all axis individually through g code.

The steps to establish communication between the Marlin firmware uploaded in the Arduino microcontroller and the Pronterface were:

- Load the adapted Marlin firmware (Figure 58) to the Arduino Mega 2560 microcontroller;

```

//
// Bepop firmware based on Sprinter and gbl.
// Copyright (C) 2011 Camiel Gubbels / Erik van der Zalm
//
// This program is free software: you can redistribute it and/or modify
// it under the terms of the GNU General Public License as published by
// the Free Software Foundation, either version 3 of the License, or
// (at your option) any later version.
//
// This program is distributed in the hope that it will be useful,
// but WITHOUT ANY WARRANTY; without even the implied warranty of
// MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
// GNU General Public License for more details.
//
// You should have received a copy of the GNU General Public License
// along with this program. If not, see <http://www.gnu.org/licenses/>.
//
//
// This firmware is a mashup between Sprinter and gbl.
// (http://sdlbld.com/wiki/index.php/Sprinter)
// (http://sdlbld.com/gbl/index.php)
//
// It has preliminary support for Matthew Roberts advance algorithm
// http://rpspro.com/robotmail/roboter-dev/2011-06v/003223.html
//
//
// All the implementation is done in *.cpp files to get better compatibility with avr-gcc without the Arduino IDE.
// Use this file to help the Arduino IDE find which Arduino libraries are needed and to keep documentation on gCode.
//
#include "Configuration.h"
#include "pins.h"

#ifdef ULTRA_LCD
  #if defined(LCD_I2C_TYPE_PCF8575)
    #include <I2C.h>
    #include <LiquidCrystal_I2C.h>
  #elif defined(LCD_I2C_TYPE_MCP23017) || defined(LCD_I2C_TYPE_MCP23008)
    #include <I2C.h>
    #include <LiquidCrystal_I2C.h>
  #elif defined(DDSI2C)
    #include <I2C.h> // library for graphics LCD by Oli Kraus (https://code.google.com/p/rgbifib/)
  #else
    #include <LiquidCrystal.h> // library for character LCD
  #endif
#endif
    
```

Figure 58 - Marlin_BCN3D_Moveo firmware opened in IDE platform.

- Establish communication, with the right parameters, between Pronterface (Figure 59) and the Arduino Mega 2560 microcontroller; with this software we can test the stepper motors motion, by moving the axis;

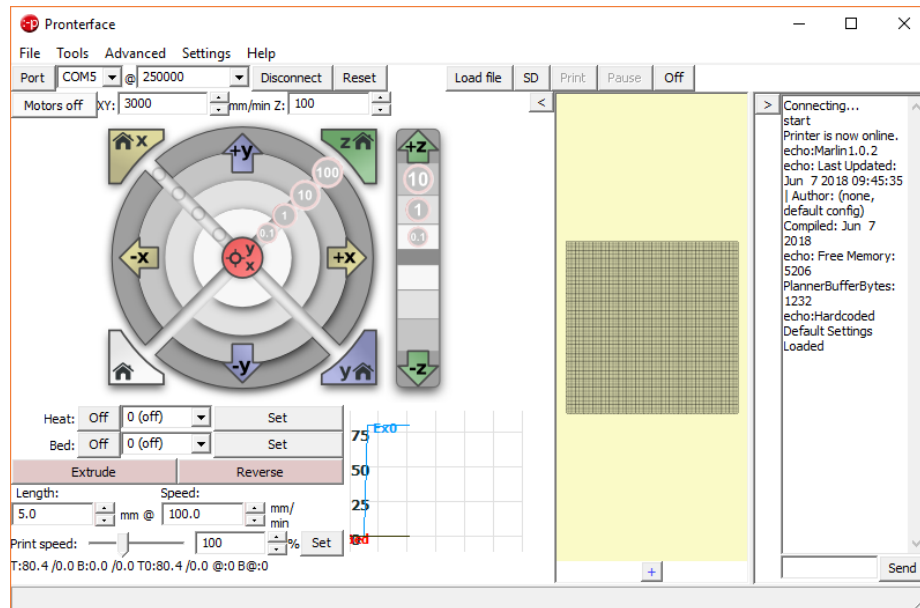


Figure 59 - Pronterface, the software used to control the robot stepper motors.

4.5 Summary

The kinematic and dynamic model development and simulation allowed a better understanding of the motion and physical constraints of the prototype. The electrical and control implementation will grant the possibility to test each joint actuator pre-assembly.

5 Implementation

This chapter covers the implementation and production of the developed prototype. It also addresses the pre-assembly and post-assembly testing of the prototype, as well as possible improvements.

5.1 3D Printing

One of the most important aspects to consider when building a robot prototype is to 3D print the designed parts; this allows keeping a low cost and a faster production time. There were two criterias regarding the choice of the technology and the material, the available equipment to perform this task and the desired mechanical proprieties. To print the parts we used two technologies, PolyJet and FDM.

5.1.1 PolyJet

Some parts were printed using a desktop 3D printer available at Bosch Car Multimedia, an Objet30 Prime (Figure 60) that uses PolyJet technology.



Figure 60 - Objet30 Prime.

The polyjet process (Figure 61) creates prototypes a bit like inkjet document printing, moving the print head along the x-axis. The difference is that, instead of jetting ink drops onto paper, the print head jets microscopic layers of liquid photopolymer onto a build tray. After building each layer, an UV light instantly cures and hardens the layer. This step eliminates the post curing, required by other technologies (Singh 2011).

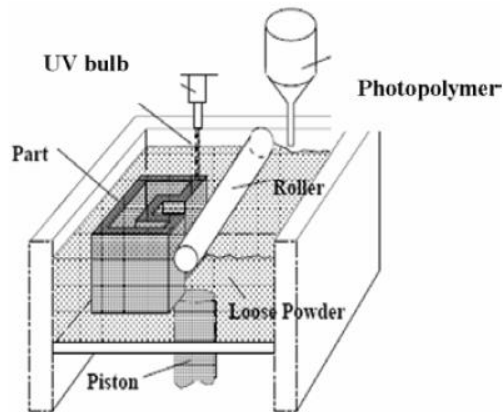


Figure 61 - Polyjet process (Singh 2011).

The printer specifications (Figure 70), the maximum build size, and the available time allowed printing only small parts of the robot. The parts printed present good aesthetics and surface quality. The material used was the VEROCLEAR RGD810 (Figure 71), due to the available stock. To produce the parts it was necessary to provide the stl files and then, with the Objet Studio software, to configure the printing parameters, according the required needs and to optimize the process. (Singh 2011)After printing, the parts need to be cleaned to remove the support material.

5.1.2 FDM

The rest of the printed parts were fabricated in DONELAB, at Minho's University, using Fused Deposition Modeling (FDM) technology (Figure 62). This is one of the most widely used forms of 3D printing at the consumer level. FDM 3D printers build parts, by extruding thermoplastic filament, through rollers at constant speed, melting it afterwards, due to the heated nozzle. The print head deposits each layer on a mobile building platform.

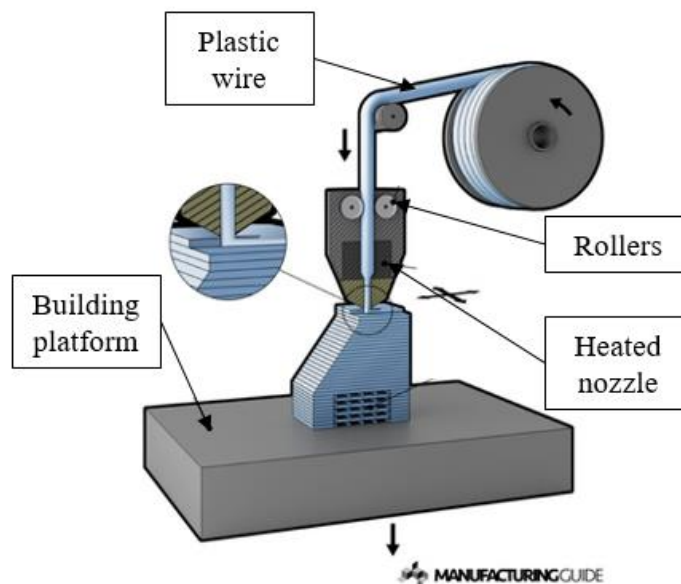


Figure 62 - FDM process (URL17).

FDM technology works with a range of standard thermoplastics, such as ABS (Acrylonitrile Butadiene Styrene), PLA (Polylactic Acid) and their various blends. This technology has the ability to produce durable parts, ready for end-use applications (URL18). The chosen material was ABS.

5.1.3 Printed parts

Some parts of the robot prototype were 3D printed, using the techniques previously referred. The location of some 3D printed parts is shown in Figure 63.

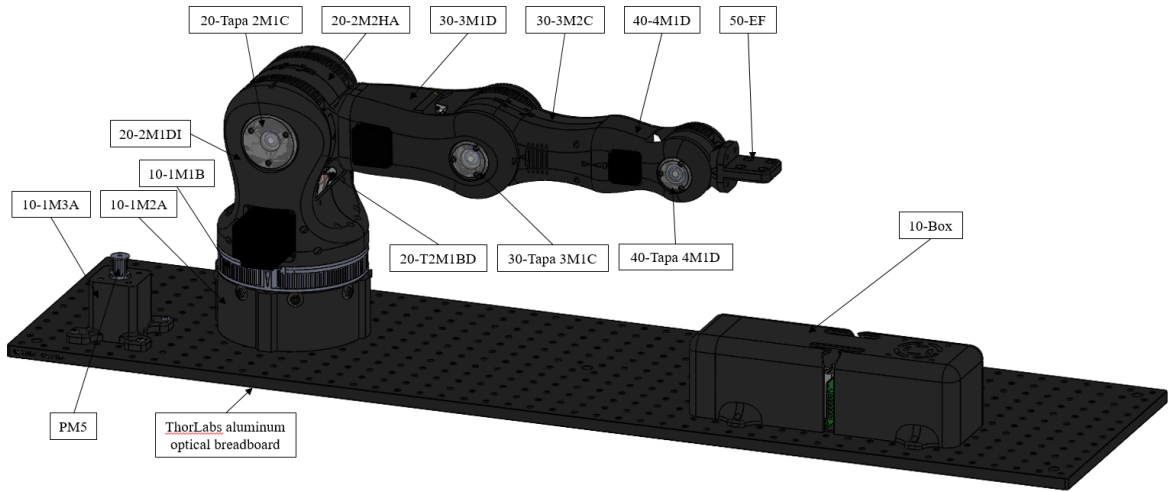































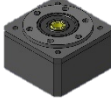
Figure 63 - Prototype drawing with the location of 3D printed parts.

A comparison between 3D models and produced part is presented in Table 14.

Table 14 - Drawing and 3D printed parts.

Module	Part Number	Quantity	Process	3D Model	3D Printed
Waist	10-1M3A	1	FDM		
	10-1M2A	1	FDM		
	10-1M1B	1	FDM		
	10-Box	1	FDM		
Shoulder	20-2M1DD	1	FDM		
	20-2M1DI	1	FDM		
	20-Tapa 2M1C	2	PolyJet		
	20-T2M1BD	1	PolyJet		

	20-T2M1BI	1	PolyJet		
	20-2M2HA	1	FDM		
	20-2M2MA	1	FDM		
Elbow	30-3M1D	1	FDM		
	30-T3M1C	1	PolyJet		
	30-Tapa 3M1C	2	PolyJet		
	30-3M2C	1	FDM		
	30-3M2CC	1	FDM		
Wrist	40-4M1D	1	FDM		
	40-Tapa 4M1D	2	PolyJet		
	40-TBC	1	PolyJet		
	40-T4M1E	1	PolyJet		
	40-4M2B	1	FDM		
	40-4M2CB	1	FDM		
End effector	50-EF	1	FDM		
Pulley	PM5	2	PolyJet		

	PM8	3	PolyJet		
Gearbox	5:1 PG	1	PolyJet		

5.2 Assembly and testing

The assembly of the robot was divided in different phases because, the components and printed parts were not all available at the same time. Before assembling the robotic arm, the electrical part and the steppers were tested. The bill of materials for this project is presented in Table 18. The assembly and testing sequence was the following:

- Assembly of electrical components (Arduino Mega, RAMPS V1.4, stepper drivers TB6560, power supply, power converter and steppers);
- Test of electrical part assembled;

5.3 Performance

Some of the most important features to evaluate the robot performance (Figure 64) are:

- Repeatability: the ability of the robot to return to a point previously visited (difference between attempts);
- Precision: difference between an achieved position (real position) and the desired position; this feature is related with the resolution;
- Resolution: the smaller incremental movement of a joint, usually detected by an encoder. Since we are not using such a control device, the resolution is calculated according to the transmission system for each joint;
- Amplitude: how many degrees the joint can rotate without collision;
- Reachable distance: the distance that the robotic arm can reach in different positions; this feature, together with the number and type of joints, provides information about the workspace.

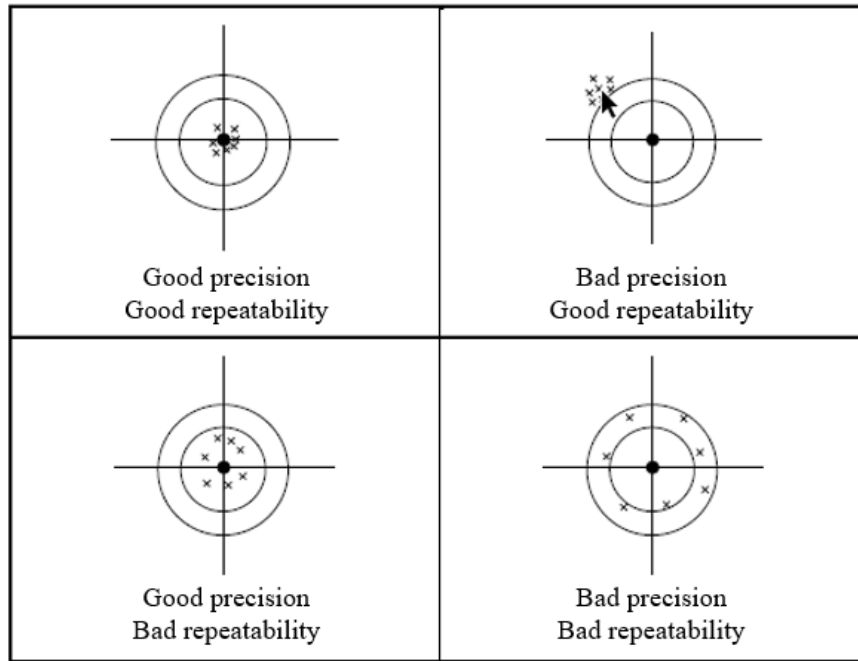


Figure 64 - Precision and repeatability performance feature (adapted from (Santos 2004)).

The prototype features regarding reachable distance (Figure 65) and amplitude were evaluated in the development phase. On the other hand, the repeatability and precision results were measured with the prototype assembled. The resolution of each joint is adjustable in the TB6560 driver, once it is a driver with micro stepping; we have set a suitable resolution for each joint, not to compromise other performance aspects, such as available torque and velocity.

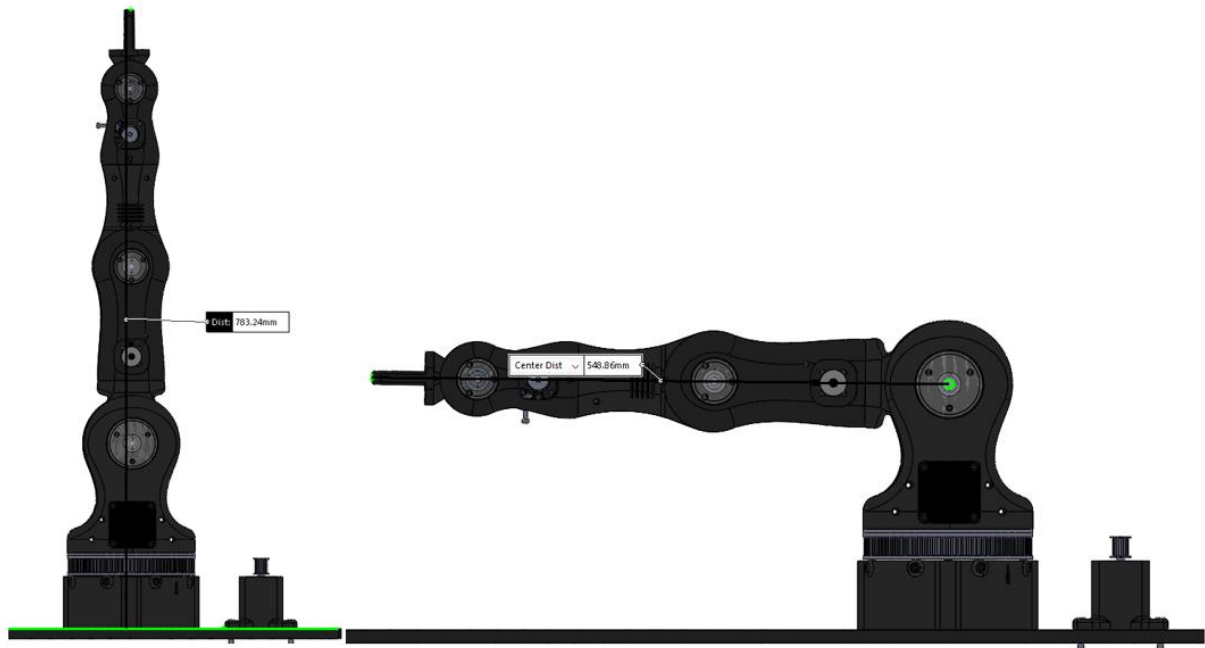


Figure 65 - Prototype reachable distance.

The performance results are displayed in Table 15 and Table 16.

Table 15 - Performance results.

Joint	Amplitude (°)	Resolution (°)
1	180	0.9
2	180	0.9
3	225	0.9
4	450	0.9
5	235	0.9

Table 16 - Performance results.

Vertical reach (mm)	Horizontal reach (mm)
783.240	548.860

5.4 Summary

The comparison made between drawings and printed parts showed some dimensional dissimilarities inherent to the process. The electrical pre-assembly tests allowed the possibility of detecting control defects, easier to correct in this stage of the assembly.

6 Conclusions and Future Work

6.1 Final Conclusions

At a personal level, the four months of internship at Bosch Car Multimedia, S.A. were of constant learning. The possibility to support in the development of products for reputed brands, to understand the problems that a development team has to deal with, regarding not only the development in itself, but also the support to production were of great value. The need to work with different departments and the possibility to visit suppliers and witness several production processes, as well as all the knowledge received from my colleagues improved even more my experience.

The project was challenging. The initial goal of designing and validating a five degrees of freedom robotic arm prototype was achieved, following the flowchart defined in the beginning of the project.

The use of an open source project as a base to start the development was an important decision to be more efficient, considering the available time. The design of the robotic arm allowed the possibility to test, in a virtual environment, possible constraints and changes that needed to be made, as well as to check if the available components were dimensionally adequate.

The available components needed not only dimensional validation, but also regarding performance. Mathematical validations, due to the mechanical solicitations and electrical needs, to access if they were able to perform the required task and to choose the lacking components, were also made. These calculations were all made assuming the worst-case scenario.

The kinematic model developed, resorting to the Denavit-Hartenberg notation, gave the possibility to understand and study the motion of the prototype and its limitations. The determination of the Denavit-Hartenberg parameters for this robot was obtained using the Denavit Hartenberg convention and allowed the possibility of simulating the motion of the robot and compute the direct and inverse kinematics matrixes, using Mat Lab toolbox.

The electronics design for this prototype gave the intended capacity to the components, regarding some performance features. The robot can reach the desired positions and has the necessary stability, due to the stepper motors capability of maintaining a fixed position. The firmware and software used for this project, initially developed for 3D printers, saved some time, since they only needed adjustments.

The use of high quality 3D printers produced parts with good mechanical properties and with good aesthetics. The printed parts needed some manual dimensional adjustment, normal in this process. The ease with which it is possible to integrate the drawings with the 3D printer's software is an advantage, regarding time saving.

The tests made regarding motion control were satisfactory, and it is possible to state that with some adjustments after assembly of the prototype it will work as projected. It is expected that the printed parts might need some dimensional adjustment inherent to the prototyping process.

In short, with computational resources, some research, time and a small investment, when compared with an industrial robot, it is possible to build a low-cost robot manipulator with a good performance.

6.2 Future working perspectives

Due to lack of time, it was not possible to assemble all parts and test them, so the following step regarding this project is to build the prototype and test it.

The developed robot in this project is a prototype; therefore, there is always room for improvement. These are some changes that could improve the robot:

- Machining wearable parts: some components used in this prototype will wear out, mainly the mechanical components that have been 3D printed. Therefore, a first improvement is to change the printed pulleys for aluminum ones. Another change is to machine the 10-1M1B part and the tensors (20-T2M1BD, 20-T2M1BI, 30-T3M1C and 40-T4M1E) with better mechanical properties material;
- LabVIEW interface: create an interface app in LabVIEW to manipulate the prototype.
- Use position sensors on the stepper motors: the actuators used in this prototype were stepper motors without position control, allowing them to be low cost, but, if an interference occurs, the robot does not reach the required position. Since it will work in a closed environment, that is not a critical problem, but by installing an encoder or a resolver, it is possible to have a feedback regarding position;
- Add a new degree of freedom: to perform the required tasks, the prototype does not need an additional degree of freedom, but in future, a new task might need it. To perform this improvement, a new end effector with ability to perform a rotational motion (Figure 66) needs to be developed;



Figure 66 - End effector motion.

- Improve the robot aesthetics: replace the printed parts for machined ones and redesign the 10-1M3A, 10-1M2A and 10-1M1B to have a single part that covers the transmission system associated with that joint;

Table 17 has summary of possible improvements and their priority.

Table 17 - Future improvements.

Improvements	Importance
Machining wear parts	High
LabVIEW interface	High
Use position sensors on the stepper motors	Medium
Add a new degree of freedom	Low
Improve the robot aesthetics	Low

References

- Alexander, Charles K and Matthew no Sadiku. 2000. "Electric circuits". *Transformation* no. 135:4-5.
- Berman, Barry. 2012. "3-D printing: The new industrial revolution". *Business Horizons* no. 55 (2):155-162. <http://www.sciencedirect.com/science/article/pii/S0007681311001790>.
- Campbell, Thomas, Christopher Williams, Olga Ivanova and Banning Garrett. 2011. "Could 3D printing change the world". *Technologies, Potential, and Implications of Additive Manufacturing, Atlantic Council, Washington, DC*.
- Childs, THC. 1980. "The contact and friction between flat belts and pulleys". *International Journal of Mechanical Sciences* no. 22 (2):117-126.
- Corke, Peter. 2017. *Robotics, Vision and Control: Fundamental Algorithms In MATLAB® Second, Completely Revised*. Vol. 118: Springer.
- Craig John, J. 1989. *Introduction to robotics mechanics and control*. Vol. 2nd ed. Reading: Addison-Wesley Publishing.
- Della Pietra, Lelio and Francesco Timpono. 2013. "Tension in a flat belt transmission: Experimental investigation". *Mechanism and Machine Theory* no. 70:129-156. <http://www.sciencedirect.com/science/article/pii/S0094114X13001353>.
- DIS, ISO. 2012. "8373: 2012. Robots and robotic devices–Vocabulary". *International Standardization Organization (ISO)*.
- Groover Mikell, P. and W. Zimmers Jr Emory. 1984. *CAD/CAM computer-aided design and manufacturing*. Englewood Cliffs, NJ: Prentice-Hall.
- Hartle, James B. 2003. *Gravity: An introduction to Einstein's general relativity*. AAPT.
- Kutzbach, Nina. 2017a. *Executive Summary World Robotics 2017 Industrial Robots*.
- . 2017b. *Executive Summary World Robotics 2017 Service Robots*.
- Laurent, Andrew M St. 2004. *Understanding open source and free software licensing: guide to navigating licensing issues in existing & new software*. " O'Reilly Media, Inc."
- McRoberts, Michael. 2013. *Beginning Arduino*. Apress.
- Rubinstein, David. 2007. "Standish group report: There's less development chaos today". *Software Development Times* no. 1.
- Santos, Vítor MF. 2004. "Robótica Industrial". *Universidade de Aveiro-Departamento de Engenharia Mecânica*.
- Serway, Raymond A and John W Jewett. 2018. *Physics for scientists and engineers with modern physics*. Cengage learning.
- Siciliano, B., L. Sciavicco, L. Villani and G. Oriolo. 2009. *Robotics: modeling, planning and control*. Advanced textbooks in control and signal processing. London: Springer.

- Singh, Rupinder. 2011. "Process capability study of polyjet printing for plastic components". *Journal of Mechanical Science and Technology* no. 25 (4):1011-1015. <https://doi.org/10.1007/s12206-011-0203-8>.
- Spong, Mark W and Mathukumalli Vidyasagar. 2008. *Robot dynamics and control*. John Wiley & Sons.
- URL1, Bosch website. Accessed 05/18. <https://www.bosch.com/our-company/>.
- URL2, Bosch website. Accessed 05/18. <https://www.bosch.com/our-company/our-figures/>.
- URL3, Bosch website. <https://www.bosch.pt/a-nossa-empresa/bosch-em-portugal/braga>.
- URL4, Bosch website. Accessed 05/18. <https://www.bosch.pt/a-nossa-empresa/bosch-em-portugal/>.
- URL5, Nanalyze website. Accessed 05/18. <https://www.nanalyze.com/2018/02/4-chinese-industrial-robot-stocks/>.
- URL6, Honeybee Robotics website. Accessed 05/18. <https://www.honeybeerobotics.com/about-us/mars/>.
- URL7, RIA. Accessed 05/18. https://www.robotics.org/content-detail.cfm/Industrial-Robotics-Industry-Insights/Mobile-Robots-and-Intralogistics-the-Always-On-Supply-Chain/content_id/6163.
- URL8, USFRPAL website. Accessed 04/18. <http://rpal.cse.usf.edu/GfM.html>.
- URL9, GitHub website. Accessed 04/18. <https://github.com/open-source>.
- URL10, Hackday website. "Hackday". Accessed 05/18. <https://hackaday.io/>.
- URL11, BCN3D Moveo website. Accessed 04/18. <https://www.bcn3dtechnologies.com/en/bcn3d-moveo-the-future-of-learning/>.
- URL12, Hackday website. "Thor". Accessed 04/18. <https://hackaday.io/project/12989-thor>.
- URL13, GitHub website. "AR2". Accessed 04/18. <https://github.com/Chris-Annin/AR2>.
- URL14, Hackday website. "6DOF". Accessed 04/18. <https://hackaday.io/project/25895-6dof-robotic-arm>.
- URL15, Thingiverse website. "6DOF Moveo". Accessed 04/18. <https://www.thingiverse.com/thing:1693444>.
- URL16, Adafruit website. "Steppers". Accessed 05/18. <https://learn.adafruit.com/all-about-stepper-motors?view=all>.
- URL17, Manufacturing Guide Sweden AB website. "Fused Deposition Modeling, FDM". Accessed 05/18. <https://www.manufacturingguide.com/en/fused-deposition-modeling-fdm>.
- URL18, Formlabs website. "FDM". Accessed 05/18. <https://formlabs.com/blog/fdm-vs-sla-vs-3d-printing-how-to-choose-the-right-3d-printing-technology/>.
- URL19, FLIR website. "FlirPoint3". Accessed 04/18. <https://www.ptgrey.com/support/downloads/10136/>.
- URL20, RepRap website. "RAMPS 1.4". Accessed 05/18. https://reprap.org/wiki/Arduino_Mega_Pololu_Shield.
- URL22, ThorLabs website. Accessed 04/18. https://www.thorlabs.com/drawings/dc7006f032eb3c43-7DD42B07-9BC7-A547-D23FE8776662F4E9/MB30120_M-AutoCADPDF.PDF.

- URL23, Stratasys website. Accessed 05/18. <http://www.stratasys.com/3d-printers/objet30-prime>.
- URL24, Stratasys website. Accessed 05/18. <http://www.stratasys.com/materials/search/veroclear>.
- URL25, Farnell website. Accessed 05/18. <http://uk.farnell.com/lin-engineering/4118m-01ro/stepper-motor-bipolar-1-7a/dp/2292909>.
- URL26, Farnell website. Accessed 05/18. <http://uk.farnell.com/nanotec/st570911108-b/stepper-motor-0-9deg-2-phase/dp/4743167>.
- URL27, BoxEletronica website. Accessed 05/18. <https://www.boxelectronica.com/en/steppers/141-nema17-stepper-motors-2-phase.html>.
- URL28, Arduino website. Accessed 05/18. <https://store.arduino.cc/usa/arduino-mega-2560-rev3>.

Appendix A: Other activities

Throughout the internship, several activities, due to the full integration in the CM-CI2/ECM8 team, were performed. These tasks were related with the Ford project needs.

In this chapter, the more relevant tasks are summarized.

1. Efficiency Measurements

A combiner head up display has a variety of components that need validation and fulfillment of a variety of requirements. For the light guide, a component that, very briefly, guides the image that is projected in the display it is necessary to measure its efficiency to access, if the supplier is fulfilling the requirements.

To evaluate that, the light guide is assembled in a JIG with a LED and a power meter, and it is measured how much power is transferred from the LED to the power meter.

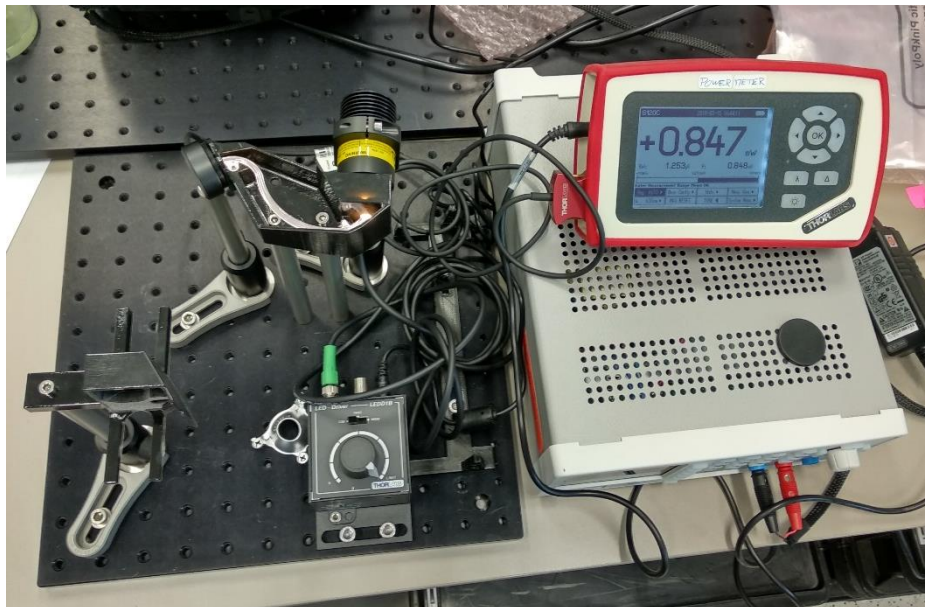


Figure 67 - JIG to measure light guide efficiency.

2. Misuse Test

The tests are not only made to the individual parts, but also to the final product. It is necessary to make a misuse test, which consists in applying a force up to a maximum of 150 N to the combiner, after subjecting it to different temperatures. The combiner is fixed, the force is applied with a dynamometer and it is verified if it breaks or unglues.

3. Torque Measurements

The three pins that are inserted in a plastic component also needed tests. It is necessary to access the torque to rotate the pin with a torque-measuring device.

4. JIG Light Guide

The first measurement JIG revealed some room for improvement. To have more accurate results two new versions of the JIG were developed, using CATIA V5 software. These new versions were projected to fabricate resorting to rapid prototyping technologies and validating them; the development was made to after validation, build an aluminum machined JIG.



Figure 68 - First version of the prototype, produced using Polyjet technology.

Appendix B: Data Sheets

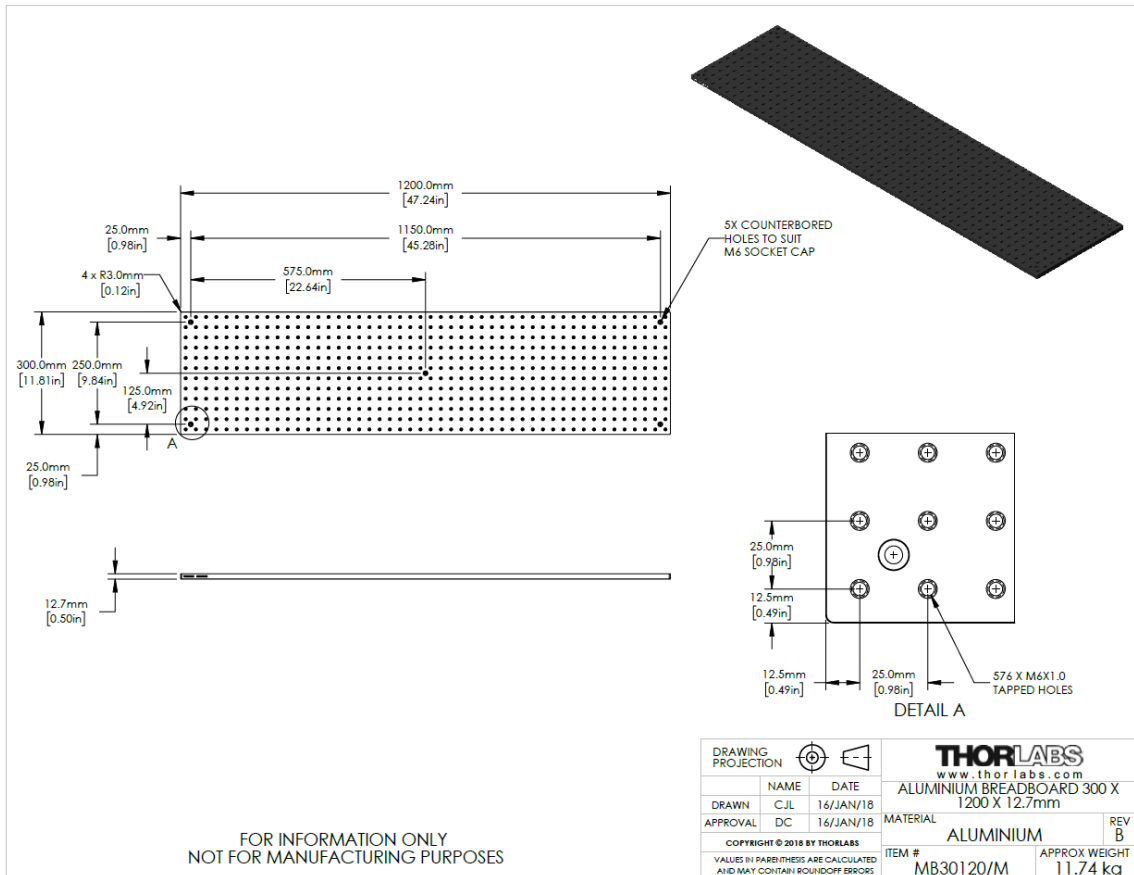


Figure 69 - Thorlabs aluminum breadboard 300 x 1200 x 12.777 (URL22).

3D PRINTER SPECIFICATIONS	
Model Materials	Rigid Opaque (VeroWhitePlus™, VeroGray™, VeroBlue™, VeroBlackPlus™) Transparent (RGD720 and VeroClear™) Simulated Polypropylene (Rigur™ and Durus™) High Temperature Rubber-like (TangoGray™ and TangoBlack™) Bio-compatible
Support Material	SUP705 (WaterJet removable) SUP706 (soluble)
Maximum Build Size (XYZ)	294 x 192 x 148.6 mm (11.57 x 7.55 x 5.85 in.)
System Size and Weight	82.5 x 62 x 59 cm (32.28 x 24.4 x 23.22 in.); 106 kg (234 lbs)
Resolution	X-axis: 600 dpi; Y-axis: 600 dpi; Z-axis: 1600 dpi
Accuracy	0.1 mm (0.0039 in.) varies depending on part geometry, size, orientation, material and post-processing method
Minimum Layer Thickness	28 microns (0.0011 in.) for Tango materials; 16 microns (0.0006 in.) for all other materials
Build Modes	Draft (36 micron); High Speed (28 micron); High Quality (16 micron)
Software	Objet Studio™ intuitive 3D printing software
Workstation Compatibility	Windows XP/Windows 7/Windows 8
Network Connectivity	Ethernet TCP/IP 10/100 base T
Operating Conditions	Temperature 18-25°C (64-77°F); relative humidity 30-70%
Power Requirements	Single phase: 100-200V; 50-60Hz; 7A or 200-240V; 50-60Hz 3.5A
Regulatory Compliance	CE/FCC/RoHS

Figure 70 - Objet30 Prime specifications (URL23).

TRANSPARENT MATERIALS					
VEROCLEAR RGD810					
	ASTM	UNITS	METRIC	UNITS	IMPERIAL
Tensile strength	D-638-03	MPa	50-65	psi	7250-9450
Elongation at break	D-638-05	%	10-25	%	10-25
Modulus of elasticity	D-638-04	MPa	2000-3000	psi	290,000-435,000
Flexural Strength	D-790-03	MPa	75-110	psi	11000-16000
Flexural Modulus	D-790-04	MPa	2200-3200	psi	320,000-465,000
HDT, oC @ 0.45MPa	D-648-06	°C	45-50	°F	113-122
HDT, oC @ 1.82MPa	D-648-07	°C	45-50	°F	113-122
Izod Notched Impact	D-256-06	J/m	20-30	ft lb/inch	0.375-0.562
Water Absorption	D-570-98 24hr	%	1.1-1.5	%	1.1-1.5
Tg	DMA, E*	°C	52-54	°F	126-129
Shore Hardness (D)	Scale D	Scale D	83-86	Scale D	83-86
Rockwell Hardness	Scale M	Scale M	73-76	Scale M	73-76
Polymerized density	ASTM D792	g/cm3	1.18-1.19		
Ash content	USP281	%	0.02-0.06	%	0.02-0.06

Figure 71 - VEROCLEAR RGD810 proprieties (URL24).

Specifications

Model	Version	MP	Imaging Sensor
FL3-U3-13E4C-C FL3-U3-13E4M-C	Color Mono	1.3 MP	<ul style="list-style-type: none"> ■ e2v EV76C560, 1/1.8", 5.3 μm ■ Global Shutter ■ 60 FPS at 1280 x 1024
FL3-U3-13S2C-CS FL3-U3-13S2M-CS	Color Mono	1.3 MP	<ul style="list-style-type: none"> ■ Sony IMX035, 1/3", 3.63 μm ■ Rolling Shutter ■ 120 FPS at 1328 x 1048
FL3-U3-13Y3M-C	Mono	1.3 MP	<ul style="list-style-type: none"> ■ On Semi VITA1300, 1/2", 4.8 μm ■ Global Shutter ■ 150 FPS at 1280 x 1024
FL3-U3-20E4C-C FL3-U3-20E4M-C	Color Mono	2.0 MP	<ul style="list-style-type: none"> ■ e2v EV76C5706F, 1/1.8", 4.5 μm ■ Global shutter ■ 60 FPS at 1600 x 1200
FL3-U3-32S2C-CS FL3-U3-32S2M-CS	Color Mono	3.2 MP	<ul style="list-style-type: none"> ■ Sony IMX036, 1/2.8", 2.5 μm ■ Rolling Shutter with Global Reset ■ 60 FPS at 2080 x 1552
FL3-U3-120S3C-C	Color	12 MP	<ul style="list-style-type: none"> ■ Sony IMX172, 1/2.3", 1.55 μm ■ Rolling shutter with Global Reset ■ 15 FPS at 4000 x 3000

Imaging Performance (EMVA 1288)	See the Imaging Performance Specification , which includes quantum efficiency, saturation capacity (full well depth), read noise, dynamic range and signal to noise ratio.
A/D Converter	12-bit (FL3-U3-13S2, FL3-U3-32S2, FL3-U3-120S3) / 10-bit (FL3-U3-13Y3, FL3-U3-13E4, FL3-U3-20E4)
Video Data Output	8, 12, 16 and 24-bit digital data
Image Data Formats	Y8, Y16, Mono8, Mono12, Mono16, Raw8, Raw12, Raw16 (all models); RGB, YUV411, YUV422, YUV 444 (color models)
Partial Image Modes	Pixel binning and region of interest (ROI) modes
Image Processing	Gamma, lookup table, hue, saturation, and sharpness
Shutter	Rolling Shutter (FL3-U3-13S2) / Global Reset (FL3-U3-32S2, FL3-U3-120S3) / Global Shutter (FL3-U3-13Y3, FL3-U3-13E4, FL3-U3-20E4); Automatic/Manual/One-Push/Extended Shutter* modes (*except FL3-U3-13Y3) See product webpage for specific model's range
Gain	Automatic/Manual/One-Push Gain modes See product webpage for specific model's range
Gamma	0.50 to 4.00, programmable lookup table
White Balance	Automatic/Manual modes, programmable via software
High Dynamic Range	Not applicable
Color Processing	On-camera in YUV or RGB format, or on-PC in Raw format
Digital Interface	USB 3.0 interface with screw locks for camera control, data, and power
Transfer Rates	5 Gbit/s
GPIO	8-pin Hirose HR25 GPIO connector for power, trigger, strobe, PWM, and serial I/O: 1 opto-isolated input, 1 opto-isolated output, 2 bi-directional I/O pins
External Trigger Modes	IIDC Trigger Modes 0, 1 (excluding FL3-U3-13E4 and FL3-U3-20E4), and 15
Synchronization	via external trigger or software trigger
Image Buffer	32 MB frame buffer
Memory Channels	2 memory channels for custom camera settings
Flash Memory	1 MB non-volatile memory
Dimensions	29 x 29 x 30 mm excluding lens holder (metal case)
Mass	Without optics: 35 g (FL3-U3-13S2, FL3-U3-32S2) / 41 g (FL3-U3-13Y3, FL3-U3-13E4, FL3-U3-20E4, FL3-U3-120S3)
Power Consumption	5-24 V via GPIO or 5 V via USB 3.0 interface, maximum <3 W
Machine Vision Standard	IIDC v1.32, USB3 Vision v1
Camera Control	Via FlyCapture SDK, CSRs, or third party software
Camera Updates	In-field firmware updates
Lens Mount	CS-mount (FL3-U3-13S2, FL3-U3-32S2) / C-mount (FL3-U3-13Y3, FL3-U3-13E4, FL3-U3-20E4, FL3-U3-120S3)
Temperature	Operating: 0° to 45°C; Storage: -30° to 60°C
Humidity	Operating: 20 to 80% (no condensation); Storage: 20 to 95% (no condensation)
Compliance	CE, FCC, KCC, RoHS
Operating System	Windows 7, Linux (32- or 64-bit)
Warranty	3 years

Figure 72 - FLIR Flea 3 datasheet (URL19).

Appendix C: Mass properties

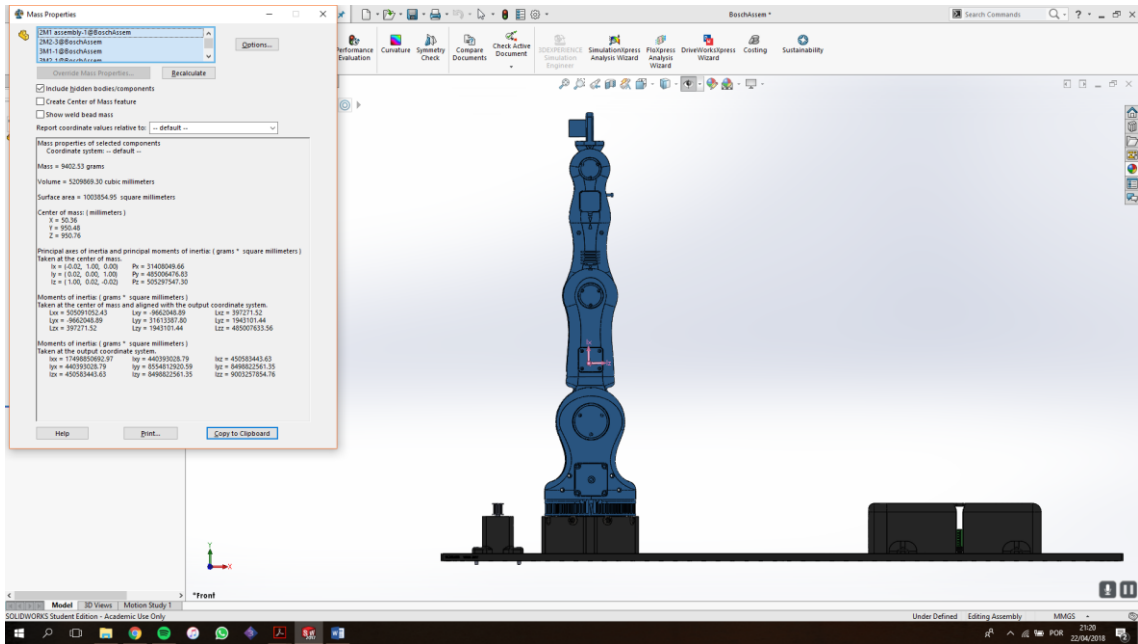


Figure 73 - Mass properties for joint 1.

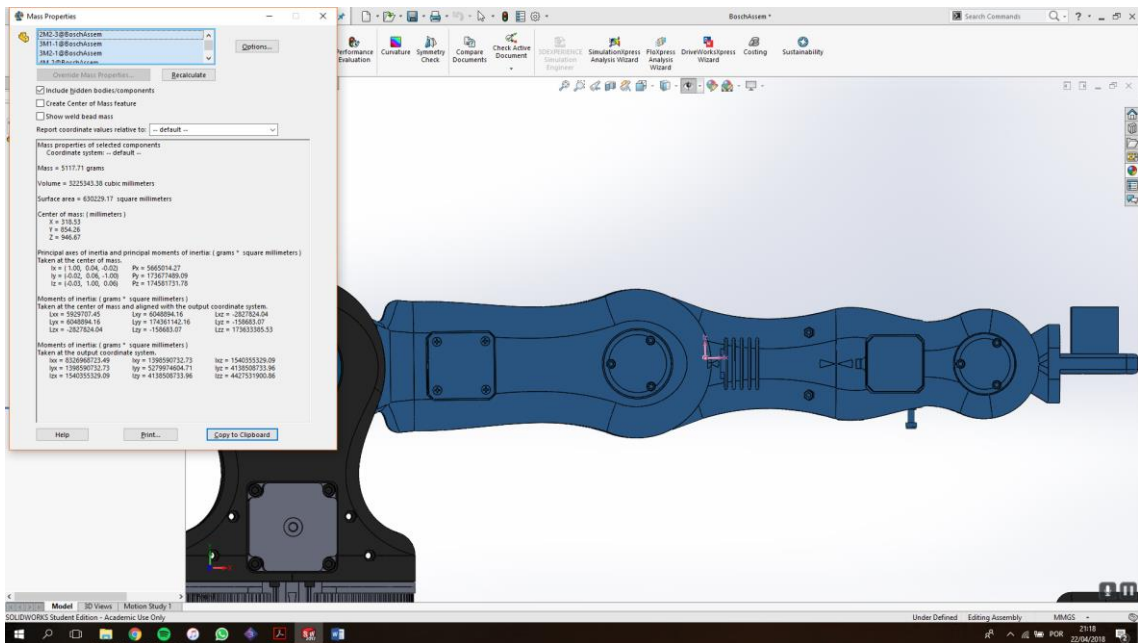


Figure 74 - Mass properties for joint 2.

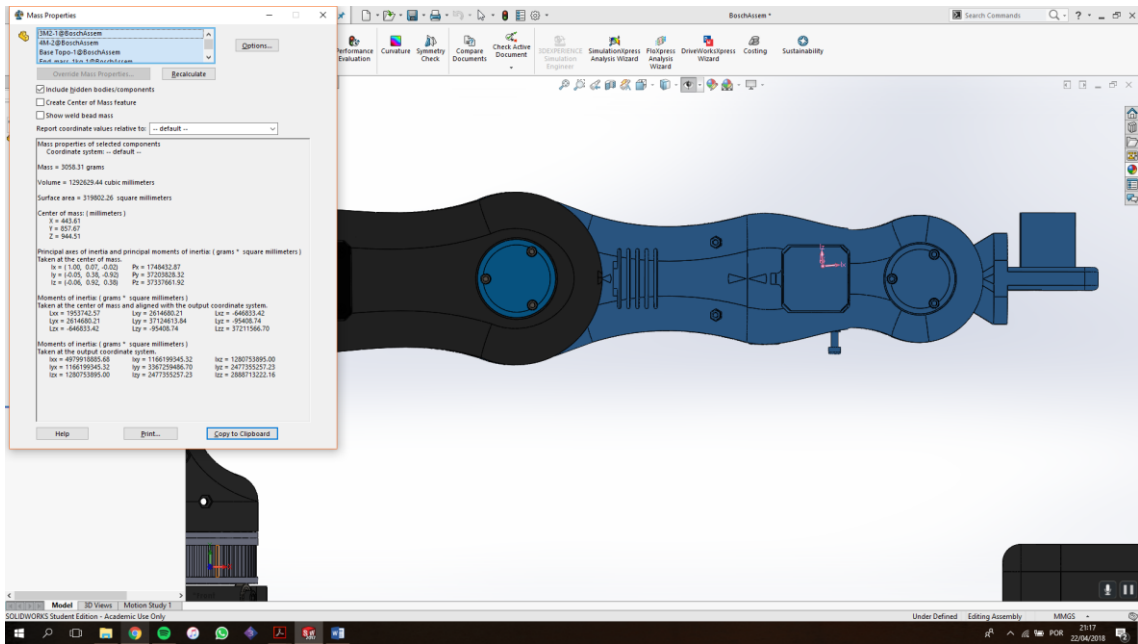


Figure 75 - Mass properties for joint 3.

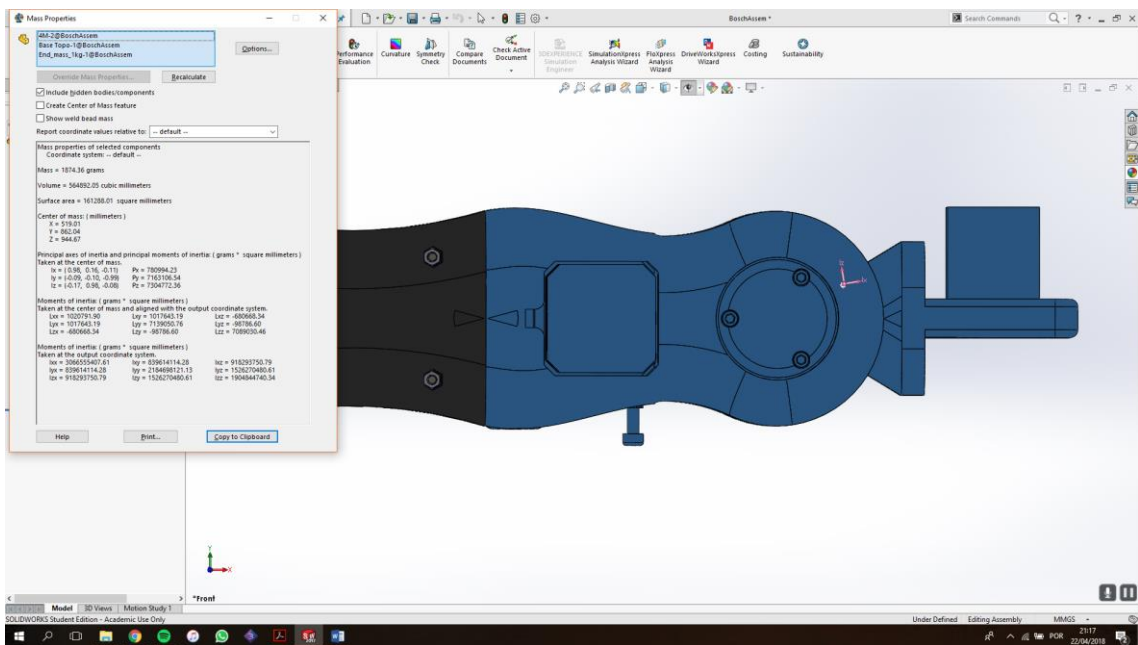


Figure 76 - Mass properties for joint 4.

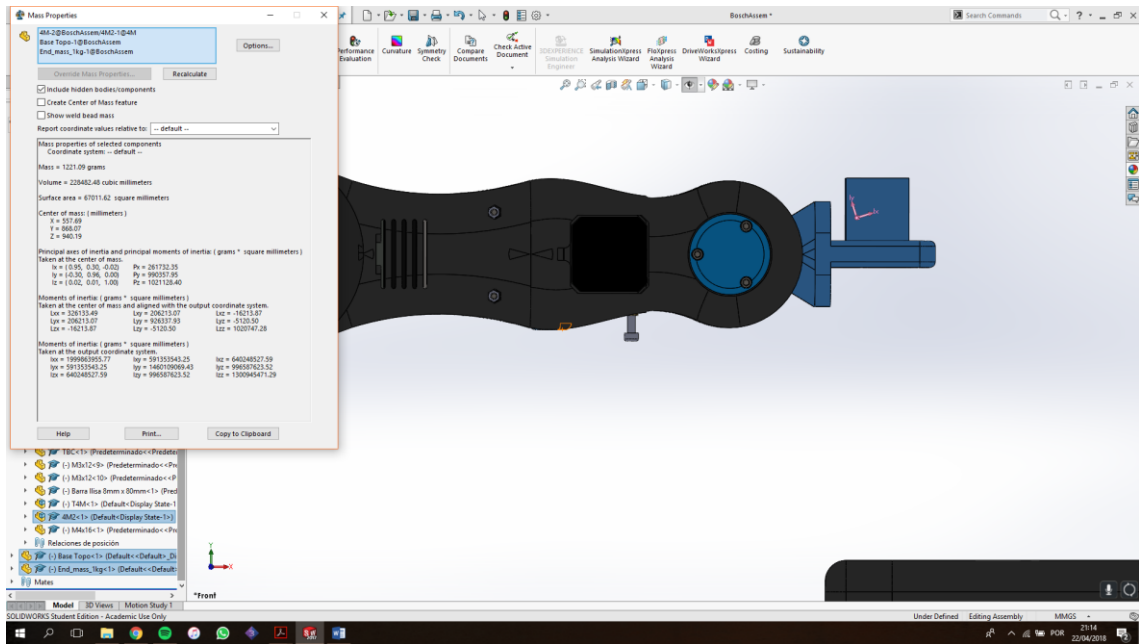


Figure 77 - Mass properties for joint 5.

Appendix D: Components datasheet

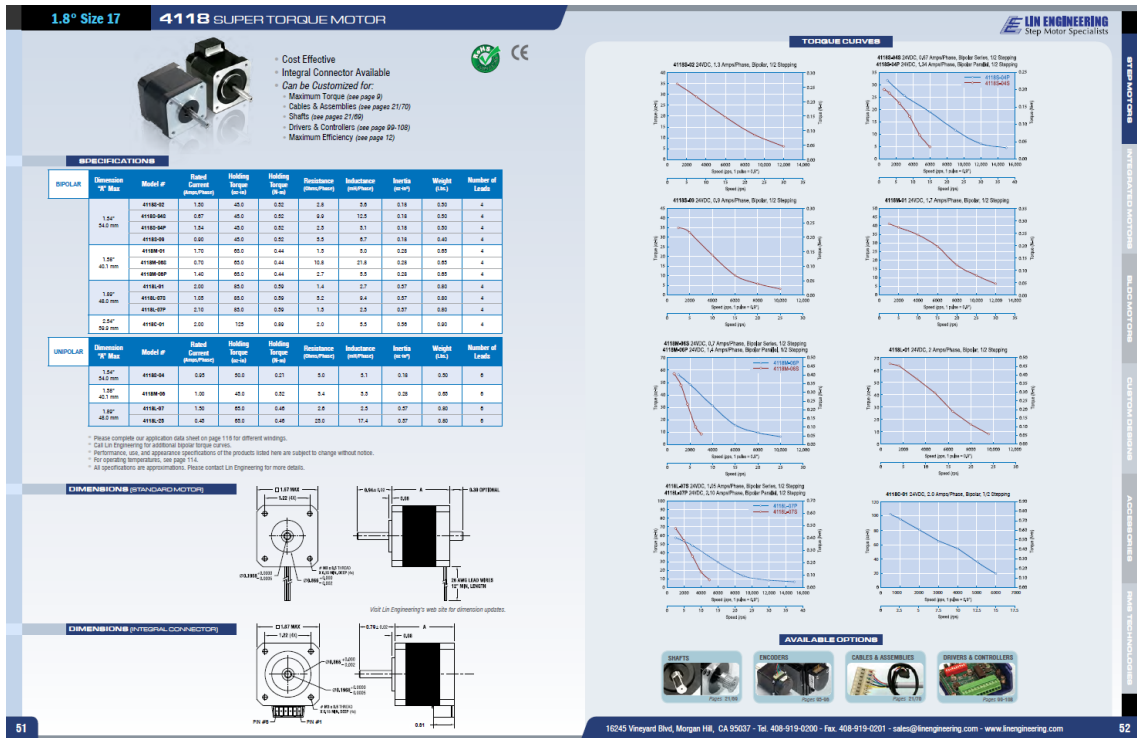


Figure 78 - LIN ENG 4118M-01-RO datasheet (URL25).

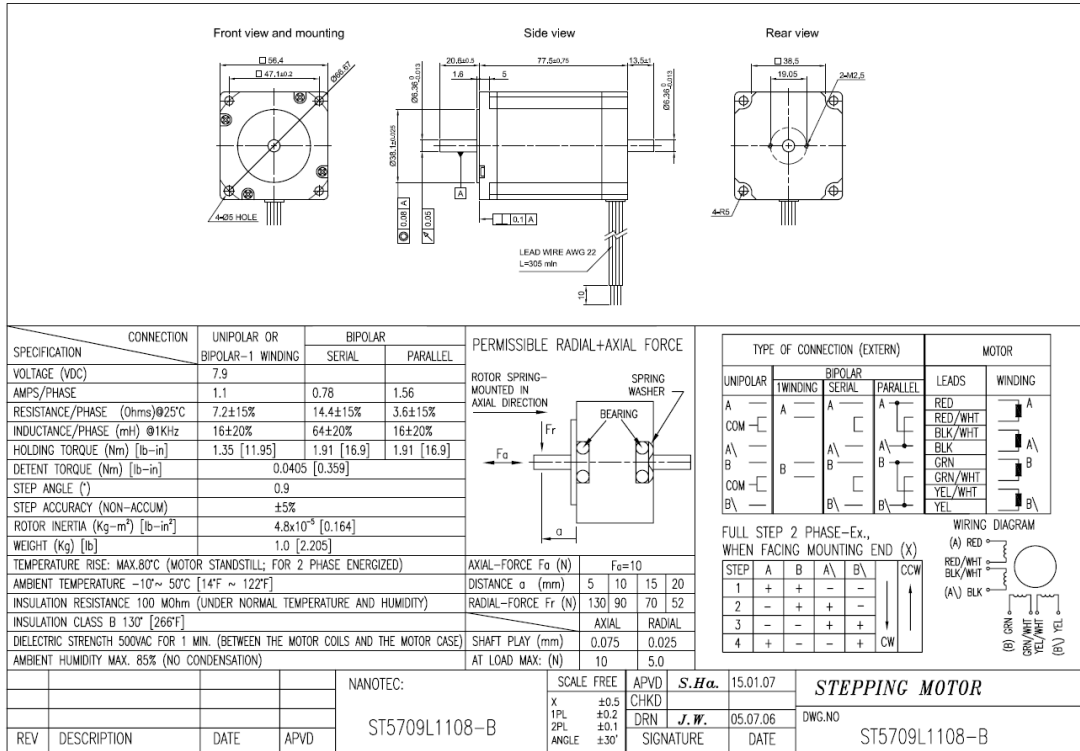


Figure 79 - ST 5709L1108-B datasheet (URL26).

- Angulo de passo (graus) : 0.9
- 2-Fases
- Tensão Nominal: 3V
- Corrente Nominal: 1.7A/Fase
- Eixo com 5mm de diâmetro
- Torque Máximo: 48N.cm
- NEMA 17

Figure 80 - 42BYGHW811 datasheet (URL27).

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	54 (of which 15 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	256 KB of which 8 KB used by bootloader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz
LED_BUILTIN	13
Length	101.52 mm
Width	53.3 mm
Weight	37 g

Figure 81 - Arduino Mega datasheet (URL28).

Appendix E: Bill of Materials

Table 18 - Project bill of materials.

Screw	M3x10	18
	M6x15	6
	M5x20	8
	M6x25	4
	M4x45	8
	M5x15	8
	M3x25	3
	M4x20	7
	M4x30	4
	M3x40	11
	M3x12	12
	M4x16	5
	M3x8	4
	M3x35	3
	M3x20	1
Autoblocant	M8	1
3D Printed	10-1M3A	1
	10-1M3B	1
	10-1M2A	1
	10-1M1B	1
	10-Box	1
	20-2M1DD	1
	20-2M1DI	1
	20-Tapa2M1C	2
	20-T2M1BD	1
	20-T2M1BI	1
	20-2M2HA	1
	20-2M2MA	1
	30-3M1D	1

	30-T3M1C	1
	30-Tapa3M1C	2
	30-3M2C	1
	30-3M2CC	1
	40-4M1D	1
	40-Tapa4M1D	2
	40-TBC	1
	40-T4M1E	1
	40-4M2B	1
	40-4M2CB	1
	50-EF	1
	PM5	3
	PM8	1
	5:1 PG	1
Bearing	5mm	8
	8mm	10
	4mm	9
	3mm	3
Bolt	M8x65	1
Brass insert	M3 BI	20
	M4 BI	4
Coupling	5 to 8	1
Lock	M5 L	8
Lock Nut	M4 LN	12
	M5 LN	8
Nut	M3 N	22
	M4 N	7
Smooth Bar	8x140	1
	8x121	1
	8x50	1
	8x80	1
Spacer	M5x10	8
	M8x20	2
Stepper	42BYGHW811	1
	ST5709L1108-B	2
	4118M-01 w/ 5:1 P G	1
	103H5205-5240	1
	4118M-01	1

Belt	T5	2m
Electronics	Arduino Mega 2560	1
	Stepper Driver tb6560	6
	RAMPS V1.4	1
	Power Supply 24V 250W	1
	Power Supply 12V 50W	1
	Cables	
	Cable USB2,0 AM7BMM	1
	Power Suplly cable	1
	Fan	1