Prototyping and Control of an Automatic Ceramic Tableware Finishing Device

Mariano José Alvarez - a49193

Work carried out under the guidance of

Prof. Dr. José Gonçalves

Prof. Dr. João Paulo Coelho

Prof. Dr. Guillermo Gutierrez

Master in Industrial Engineering

2021-2022

Prototyping and Control of an Automatic Ceramic Tableware Finishing Device

Dissertation submitted to the School of Technology and Management of Bragança to obtain the Master Degree in Industrial Engineering

Mariano José Alvarez - a49193

2021-2022

Dedication

I dedicate this work to my parents, Willam Alvarez and Nora Vaudagna, who were always there to support me and help me move forward. Without them I could not be where I am today. I am eternally grateful for their support.

Acknowledgment

I hereby want to thank all those who in one way or another have contributed to the development of this project.

Firstly, to my supervisors at Instituto Politécnico de Bregança (IPB), Professor Dr.José Gonçalves and Professor Dr. João Paulo Coelho, for all the support, guidance and advice during the course of this work. I am very grateful for all your help!

To my supervisor at the Universidad Tecnológica Nacional (UTN), professor Dr. Guillermo Gutierrez, for his support and predisposition.

To my home university, UTN, for the opportunity to participate in the Dual Diplomacy program. I would never have had the opportunity to meet such nice people from all over the world without it.

To my girlfriend, Daniela Francisco, who despite the distances always supported me and was by my side.

To my family and friends, who were always encouraging and motivating me. Nothing could be done without them.

Resumo

Garantir a qualidade da etapa de acabamento em produtos cerâmicos é uma questão fundamental em qualquer indústria de fabricação de louças. Essa qualidade de acabamento é um desafio a ser realizado por um sistema de automação quando as peças apresentam geometrias muito irregulares. Este trabalho foca-se neste problema e tem como objetivo descrever as etapas de prototipagem e teste de um sistema mecatrónico a ser utilizado no processo de acabamento na empresa GRESTEL - Produtos Cerâmicos S.A., onde atualmente é feito manualmente. Este trabalho foi desenvolvido sob os auspícios do projeto STC 4.0 HP e envolve o controle, em malha fechada, da força de contato e velocidade de rotação de uma esponja úmida que será utilizada no processo de alisamento das peças cerâmicas antes de serem submetidas à processo de cozimento.

A estrutura mecânica da solução projetada foi prototipada usando uma tecnologia baseada em FDM. Uma impressora 3D foi utilizada para a fabricação das peças estruturais para suportar a esponja rotativa e os sensores de medição. Além disso, um controle baseado em PID é usado para controlar o sistema. Uma vez que o protótipo foi projetado e montado, uma série de testes e medições foram realizados levando à conclusão de que a abordagem proposta é adequada para atender aos requisitos de projeto.

Palavras-chave: Grés Louça Cerâmica, Acabamento Automatizado, Controle de malha fechada.

Abstract

Ensuring the quality of the finishing stage in ceramic products is a fundamental question in any tableware manufacture industry. This finishing quality is a challenge to be carried out by an automation system when parts present highly irregular geometries. This work focus on this problem and aims to describe the prototyping and testing stages of an mechatronic system to be used in the finishing process at the GRESTEL - Produtos Cerâmicos S.A. company, where is presently done manually. This work was developed under the auspices of the STC 4.0 HP project and involves the control, under closed loop, of the contact force and rotation speed of a wet sponge that will be used in the smoothing process of the ceramics parts before being submitted to the cooking process.

The mechanical structure of the devised solution was prototyped using a FDM based technology. A 3D printer was used for the manufacturing of the structural parts to support the rotating sponge and measurement sensors. In addition a PID based control is used to control the system. Once the prototype has been designed and assembled a series of tests and measurements were carried out leading to the conclusion that the proposed approach is adequate to meet the design requirements.

Keywords: Stoneware Tableware Ceramics, Automated Finishing, Closed-Loop Control.

Contents

List of Figures

Acronym

ADC Analog to Digital Converter.

DC Direct Current.

IPB Polytechnic Institute of Bragança.

LVDT Linear Variable Differential Transformers.

PI Proportional-Integral.

PID Proportional–Integral–Derivative.

PLA Polylactic Acid.

PPR Pulse Per Revolution.

PV Process Variable.

PWM Pulse Width Modulation.

SCC Signal Conditioning Circuit.

SP Set-Point.

TTL Transistor-Transistor Logic.

USB Universal Serial Bus.

Chapter 1

Introduction

In this chapter the objectives, the motivation of the work and the structure of the document are presented. A brief overview will be made to what industry 4.0 is in relation to the production of ceramic's tableware.

1.1 Overview

Portugal has a long tradition in the casting of ceramics ranging from the manufacture of products for civil construction to household and ornamental or decorative pieces. Indeed, more than 300 companies are currently active on the production of those types of products and, in overall, this industry has a very positive footprint on the Portuguese trade balance [1].

Besides sustainability and by-products reduction goals, modern ceramics has a constant pressure to manufacture increasingly complex products, with tight quality standards, operating in a strongly globalized world. The previously refereed conditions force ceramic production companies to push the envelope toward the development of alternative manufacturing processes and the integration of cutting-edge technologies into the fabrication loop. For example, the use of virtual and augmented reality technologies [2], or machine vision for quality control [3] have been used to help achieve this goal. In addition, steps are being given toward a full industry digitalization, which will be able to interconnect

the machines between the physical world and the virtual digital domain. Once in the virtual realm, big data and machine learning algorithms will enable the prediction of the manufacturing chain behaviour leading to an increase in both production quality and plant's productivity [4].

Besides those high level improvements in the manufacturing chain management and quality control, ceramic industry must increase its automation level and replace human operators attached to repetitive tasks. Such as, the removal of excess material that derives from the casting process. However, due to the fragile nature of the produced ceramics, replacing human labour by robots is a very complex task. Specially in activities that deals with soft materials with irregular shapes. In this context, tactile and force feedback actuation must be included in any robotic manipulator in order to control the amount of pressure exerted in the material while adapting to disturbances arriving from slight differences in the production batch due to random changes in the operating conditions.

The integration of tactile capacity in a robot is instrumental for achieving high adaptability, since this type of sensors are able to provide local information that is not accessible by other types of sensing methods such as cameras or lasers. Since the sense of touch is fundamentally 3D, it is an essential asset when dealing with non-systematic environments due to all the uncertainty that surround the manipulated objects. Within any ceramic industry frame of reference, the ability for robotic systems to be able of handling ductile objects is not only desirable, but mandatory.

At the present, state-of-the-art cobots are able to measure forces and torques on all its joints and on flexible parts of its mechanical structure. However, for tackling delicate tasks such as the ones already enumerated, tactile sensors must be added to the manipulator. Currently, those types of sensors consists on a matrix of several sensing elements scattered along the manipulator working area in order to be able to produce a pressure map [5].

In the market there are some commercial solutions that seek to solve these series of problems, such as the finishing of cups and bowls [6], or for ceramic pieces with irregular shapes through manipulation with collaborative robots [7]. In addition, there are different articles related to the problem, such as, the control of the applied force and prediction of

deformation [8], or the calculation of the trajectory in the manipulation of objects through simulation [9]. However, in these examples you can see partial solutions or related to other applications. It is for this reason that this work seeks to find a solution to the finishing of ceramic pieces together with the STC 4.0 HP project. At the same time this work continues with the article published in the 15th APCA International Conference on Automatic Control and Soft Computing (CONTROLO 2022) [10].

1.2 The STC 4.0 HP Project

The STC 4.0 HP project (New Generation of Stoneware Tableware in Ceramic 4.0 by High Pressure Casting Robot work cell) aims to establish the production conditions that will allow GRESTEL to worldwide launch a range of innovative products in fine stoneware tableware, as a result of the combination of different Ceramics 4.0 technologies of the industrial value chain, with robotic finishing and artificial vision control, supported by new knowledge and technological developments at industry 4.0 level. This project aims to automate an industrial process that, until recent time, would have been almost unthinkable. With the advent of collaborative robots and the advancement of fundamental research in industrial robotics, today it is possible to approach the robotization of all forming processes, which include high pressure pressing, drying and subsequent finishing.

The project consortium is composed by the industrial company GRESTEL, by two entities of the scientific Portuguese system, the Polytechnic Institute of Bragança (IPB) and the Polytechnic institute of Leiria, and also the Research Group from Limerick School of Art & Design of the Limerick Institute of Technology (Ireland).

Currently in GRESTEL facility, the production process of ceramic pieces begins with their planning, followed by modeling through molds. Then the piece is allowed to dry up to a certain percentage, to begin the finishing process. The finishing process of the ceramic pieces, is carried out by a system that uses a series of wet rotating sponges which are pressed against the ceramic pieces (Figure 1.1b), removing the excess material. Once this process is done, the pieces are finished drying and go to the decoration and

firing stage. In the Figure 1.3 you can see a flowchart of the complete process for the production of a ceramic piece in the company GRESTEL. But this finishing process it can only be implemented for ceramics with regular circular shapes, it is for this reason that the authors seek to develop a solution for irregular pieces, which requires the control of force and rotation speed.

(a) (b)

Figure 1.1: a) Suction grips. b) Finishing system implemented in GRESTEL

(a) (b)

Figure 1.2: a) Before and b) after being fired in the oven.

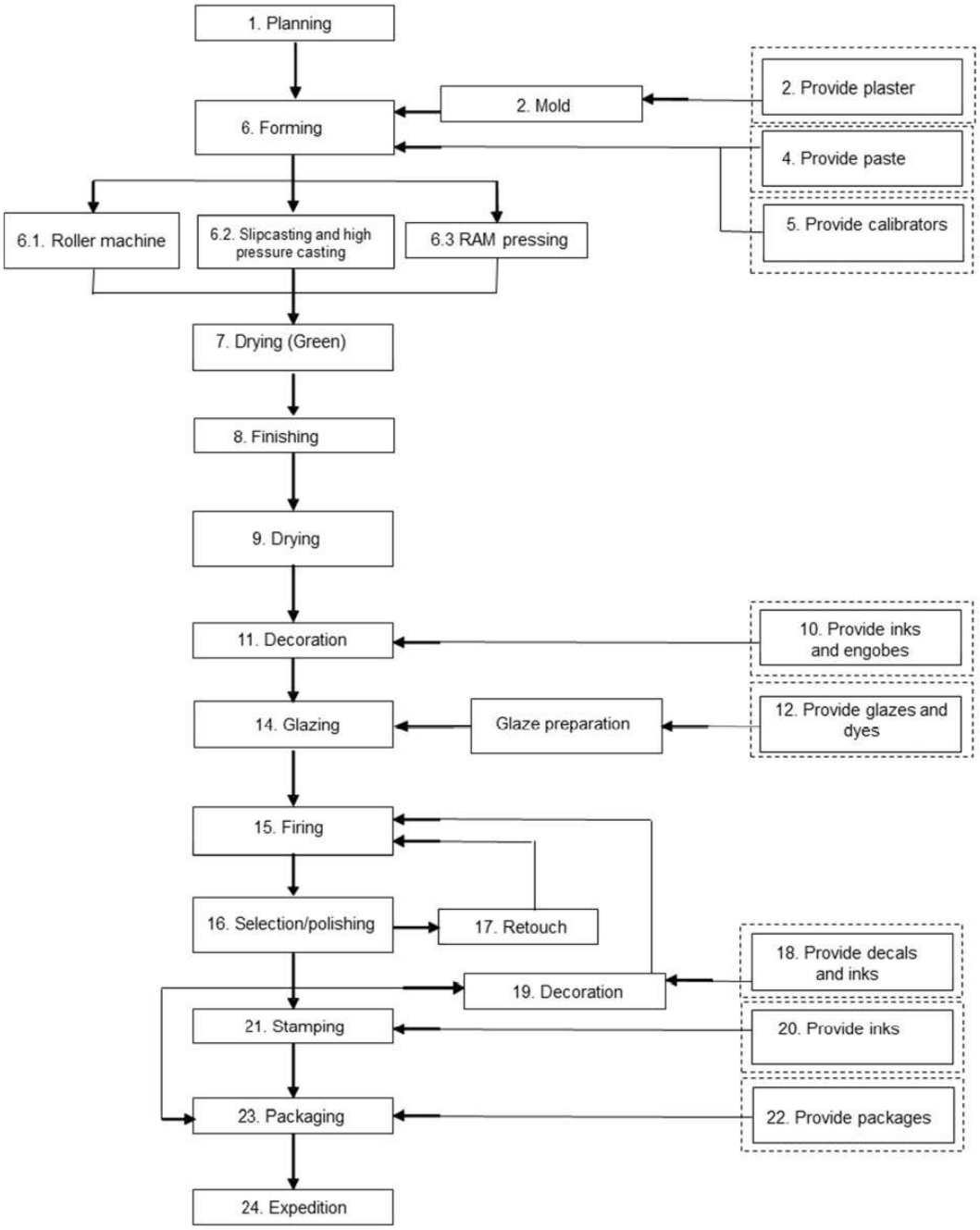


Figure 1.3: Flowchart of the production of ceramic pieces.

1.3 Objectives and Goals

The goal for this prototype is to obtain a system capable of performing a homogeneous finish with a high degree of repeatability, focused on ceramic pieces with irregular shapes. Eliminating the interaction of man in this process.

For this, the system must be able to control, in closed loop, a DC motor that allows to maintain constant the speed of rotation of a sponge, and a stepper motor that is in charge of controlling the position of the structure that supports the sponge. As well as measuring the current consumed by the DC motor, by means of a current sensor based on a hall effect sensor, allowing an approximation of the force applied to the ceramic piece.

It is expected with these solutions to get a first look at the possible approaches to accomplish this task.

1.4 Document structure

In this document it will be presented the first prototype of a polishing machine, that will interact with a collaborative manipulator (that will handle a ceramic part that needs to be finished). The described prototype was developed on the scope of STC 4.0 HP project and it will be described in the next chapters, describing its requirements, mechanical design, its control and finally some conclusions and future work are presented.

This document is organized in 5 chapters, where the current chapter 1 presents the context, objectives and proposal of this work.

Chapter 2 presents a problem statement, where a concise description of the issue to be addressed and the condition to be improved upon, will be made.

Chapter 3 presents the development of the modeling and assembly of the structural parts and the analysis of the devices to be used, along with a description of the interconnections of the subsystems. The designs of the implemented controllers and the design of the firmware are also explained.

Chapter 4 tests and measurements of the systems are carried out separately and together, analyzing the results and data obtained. Such as the settings of the controller parameters Proportional-Integral (PI), an analysis of the system step response, its response to load disturbances and some speed control and position control tests.

Finally, in chapter 5 a general conclusion about the prototype and its performance is shown, along with some considerations to continue working on future projects.

Chapter 2

Problem statement

The finishing of irregular ceramic pieces in the GRESTEL company should be a process that can be fully automated, improving the quality of the finish together with a reduction in production times and costs. As well as a reduction in the personnel necessary to carry out the work, thus being able to dedicate them to other tasks.

However, with the current finishing system, it becomes impossible to perform the task on this type of pieces, since the irregular shape does not allow the correct operation of the current system, requiring control of the position of the sponge as the piece rotates. This forces the company to carry out this task by hand, making the process more inefficient and with less precision in its result. Resulting in lower production and higher spending on labor and materials. As well as a lower quality in the final product, damaging the image of the brand.

This is why the implementation of an automatic system with control of the force applied to the ceramic pieces is proposed, this is possible thanks to the use of closed loops to control the speed of rotation of the sponge and its position. Keeping the speed constant allows a more homogeneous result throughout the entire production run, and the possibility of adjusting the position allows the pressure exerted to be constant, avoiding damaging the piece. In this way productivity increases and waste is reduced.

The steps followed to deal with this problem will be explained below.

Chapter 3

Material and Methods

As already mentioned, the finishing of ceramic pieces is a process that requires great precision and care. It is for this reason that the main feature of this prototype is to measure and control the pressure that is exerted on the ceramic. To achieve this, different approaches were analyzed, of which it was decided to apply a closed-loop control of the rotation speed of the sponge. The aim is to keep the speed of rotation constant in order to have a finish that is as homogeneous as possible. A complementary approach applied was that from the measurement of the current of the Direct Current (DC) motor, it is possible to obtain an estimate of the applied force, since they are proportional to each other, and once the value of the force is known, with a position control, it is possible to adjust the location of the machine to vary the applied pressure.

This chapter presents the different stages of project development, such as mechanical setup, measurement and actuators, and controller and firmware design.

3.1 Mechanical setup

The preliminary design began by making a list of the different devices to be used, analyzing the different options and choosing the most suitable for this application. Once the elements to be used were chosen, a preliminary model of the structure was made, taking the measurements of the devices to be used as reference.

The design of the structure was based on a strategy already deployed in GRESTEL facility (Figure 3.1) where a platform moves forward and backward, pressing the rotating sponge on the piece to be finished. With the new prototype, although it can also work in this way, the forward and backward movement is made to make a fine position adjustment, in order to keep the applied force constant despite disturbances.

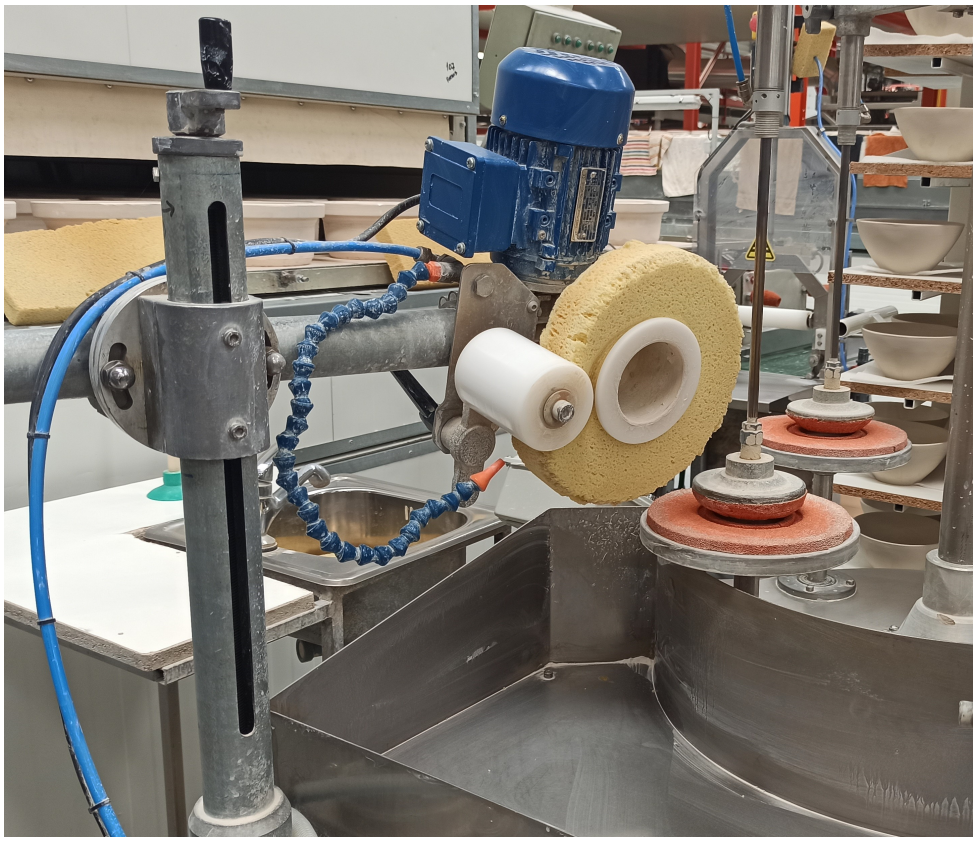


Figure 3.1: Currente system implemented in GRESTEL.

For this prototype, the parts were designed and printed on a 3D printer. The printing material is Polylactic Acid (PLA). This prototype is exclusively for concept testing in the laboratory as it is not yet adapted for use in industry.

3.1.1 Design

The system is attached to the platform, an acrylic (PMMA) sheet of 550x300x3 mm, which has rails mounted on it (Figure 3.2). The structure that supports the sponge

moves thanks to a lead screw that is driven by a stepper motor. The chosen actuator for this was a stepper motor with a rated current of 0.64 A and a voltage of 24 V [11]. It was chosen because has enough torque to move the structure without losing any step. The motor has an angular resolution of 1.8^{o} and together with the screw has a resolution of about 0.04 $mm/step$, and a speed of 26 mm/s.

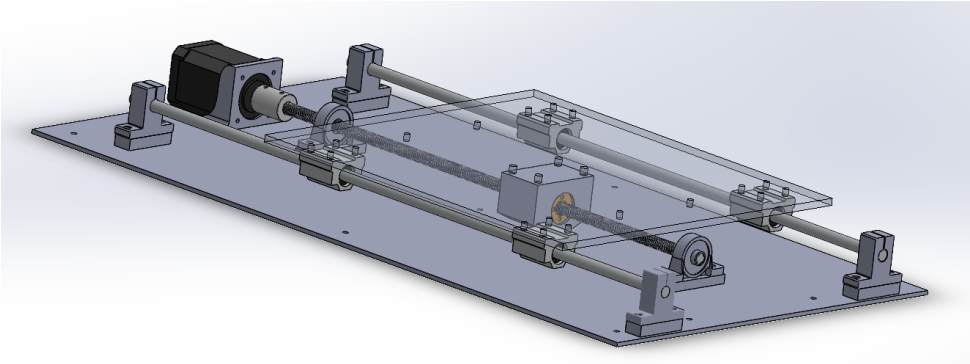
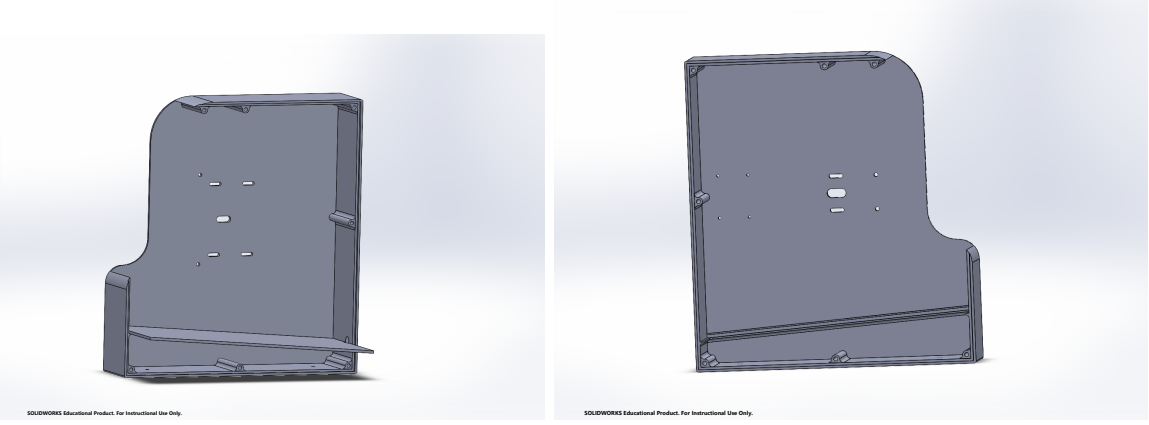


Figure 3.2: Platform with rails and stepper motor.

The platform, were designed using SOLIDWORKS® 3D CAD design software version 2020. The design of the structure that holds the sponge was also made with this CAD software. It is a box with an opening that covers the sponge to avoid accidents and so that the water used to moisten the sponge does not splash around it.



(a) (b)

Figure 3.3: a) Left side of the box that supports the sponge. b) Right side of the box.

At first it was thought to make a system where the axis of the sponge could move a

few millimeters, in order to measure the displacement of the sponge, and to be able to calculate, by means of springs and a LVDT, the force that is applied to the ceramic piece, but this idea was discarded due to the lack of greater precision in the machining and the available materials. Instead, the motor that rotates the sponge, which is a gear motor with encoder, is attached to one side of the box (Figure 3.4), coupled to the sponge shaft, at the other end of the shaft it rests on a bearing that allows it to rotate freely .

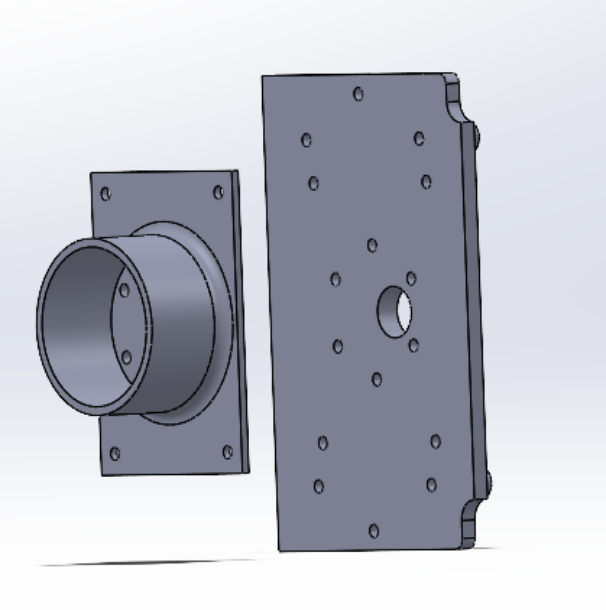


Figure 3.4: DC motor support

Once the structures were designed, they were printed and assembled together with the other devices.

3.1.2 Assembly

First, the height adjusters were attached to the acrylic platform. These are custom made so that the axis of the stepper motor is aligned with the screw.

Figure 3.5: Platform with height adjusters.

Then the rails and screw were added.

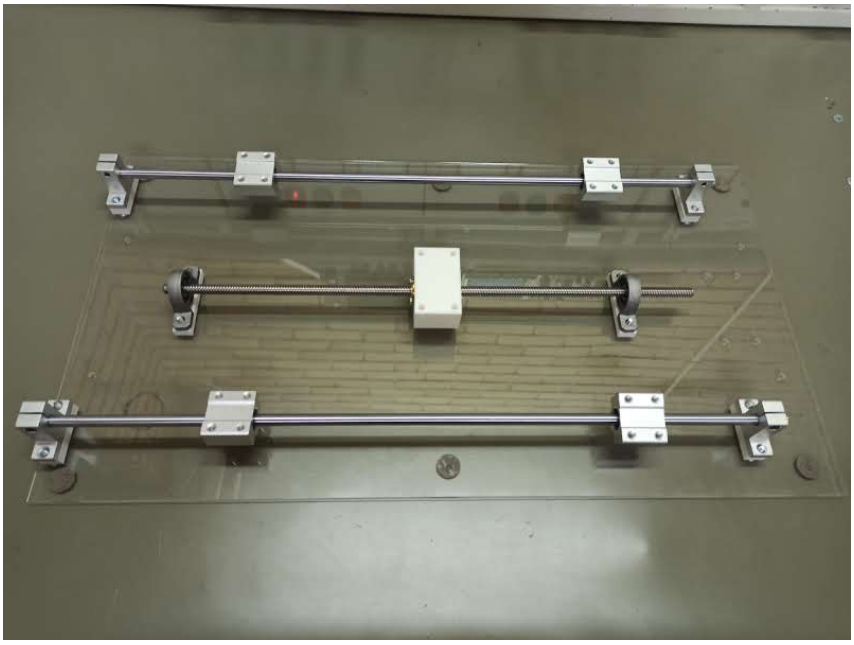
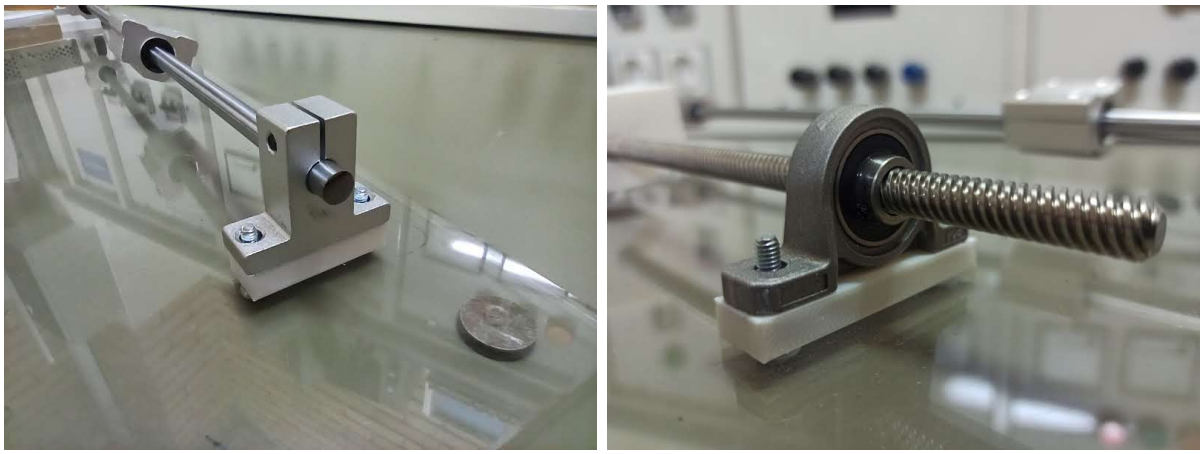


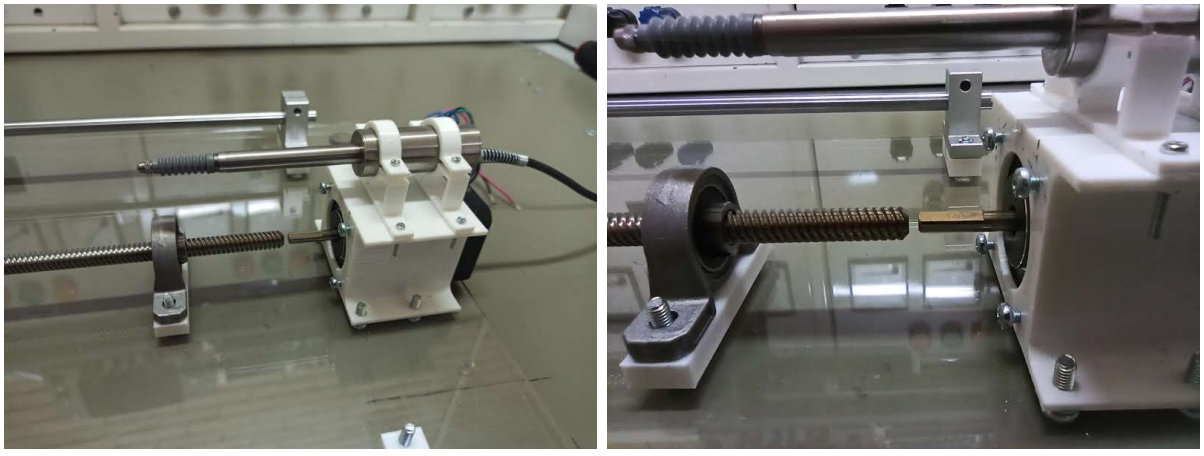
Figure 3.6: Adding rails and lead screw.



(a) (b)

Figure 3.7: a) Rail support and b) Screw bearing with height adjusters.

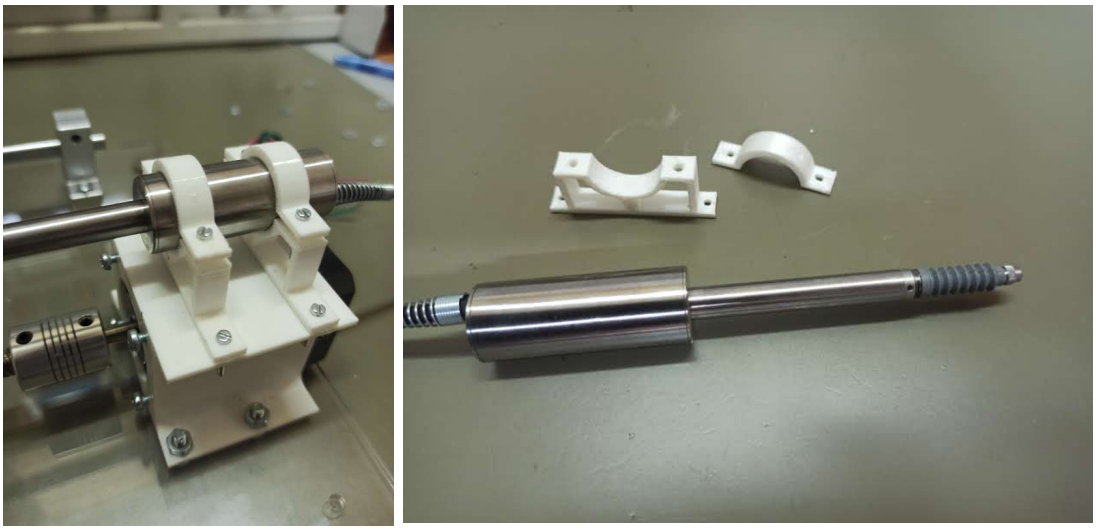
The stepper motor mount goes on one end of the deck, checks for shaft and screw alignment, and attaches with a coupler.



(a) (b)

Figure 3.8: a) Stepper motor mount. b) Correct alignment.

On top of this support the LVDT is fixed with its own grips.



(a) (b)

Figure 3.9: a) LVDT mounted over the stepper motor. b)LVDT grip.

The box base is then added and bolted to the bearings and nut housing.

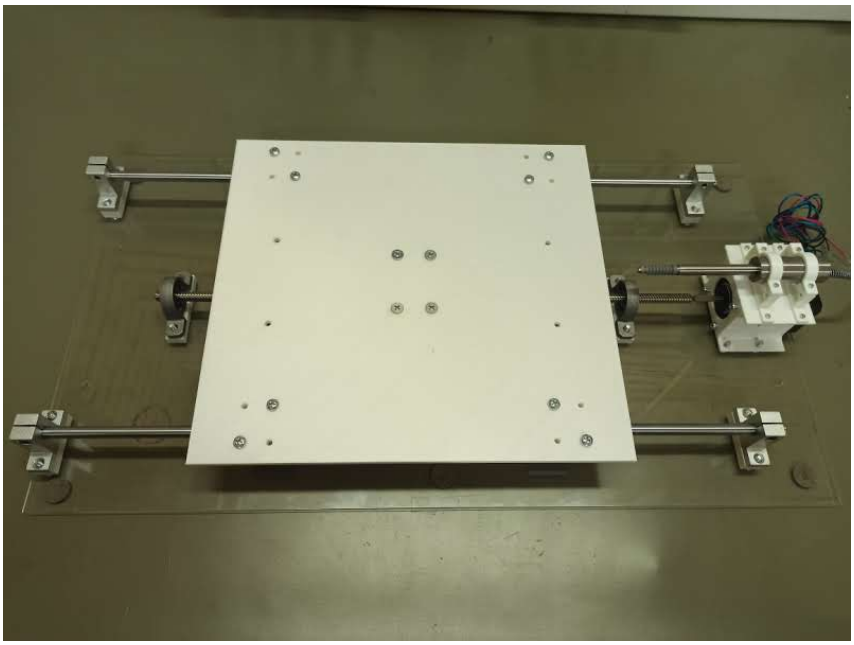
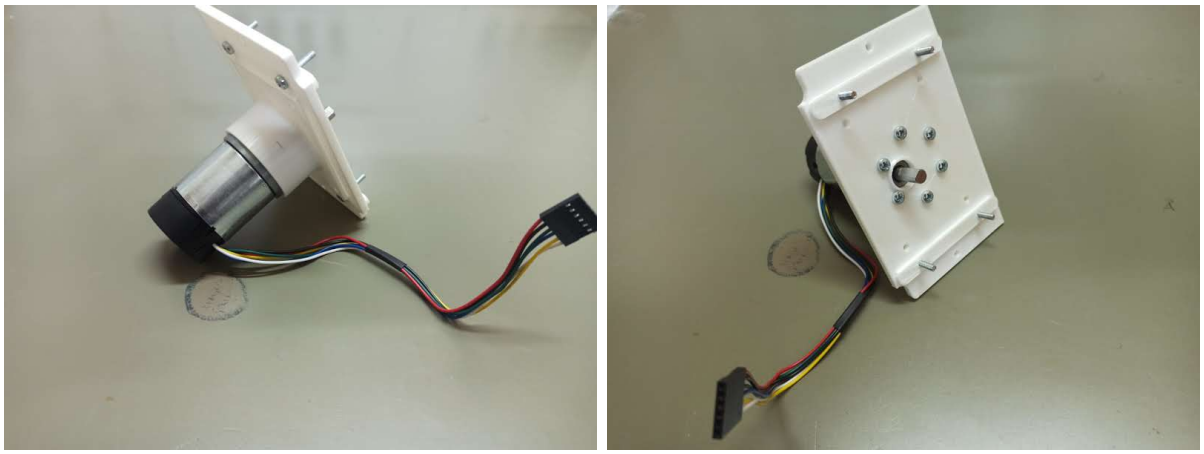


Figure 3.10: Base mounted.

On the other hand, the DC motor is attached to one of the sides of the box,

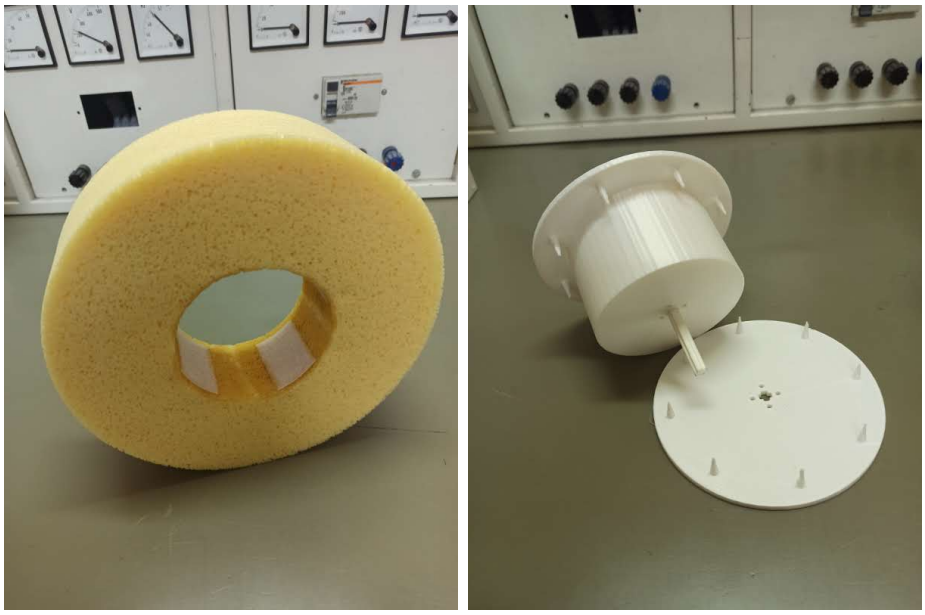


(a) (b)

Figure 3.11: DC Motor support

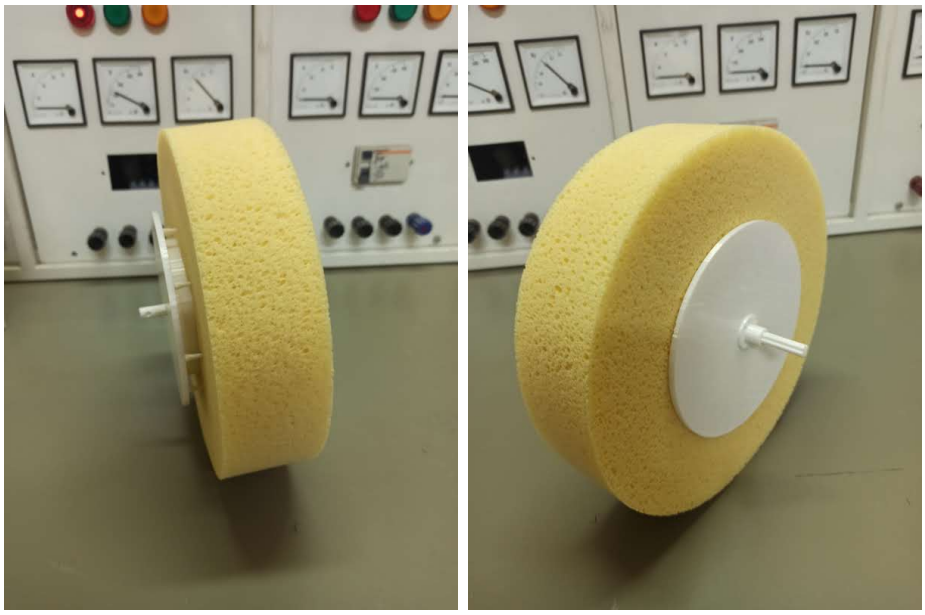
Figure 3.12: DC motor attached.

and the shaft is assembled with the center of the sponge,



(a) (b)

Figure 3.13: a) Sponge. b) Assembly of the center of the sponge.



(a) (b)

Figure 3.14: Sponge and its shaft assembled.

then it is coupled with a coupler to the shaft of the dc motor,

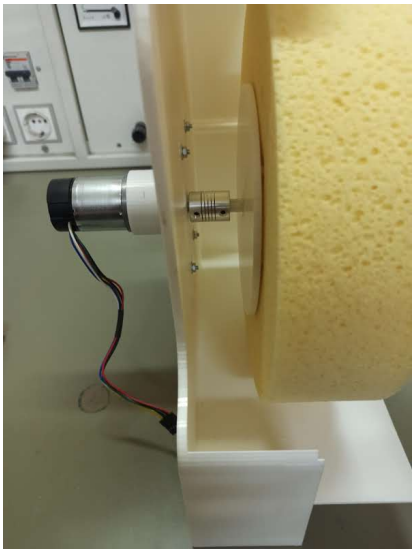


Figure 3.15: Shaft coupled with the DC motor

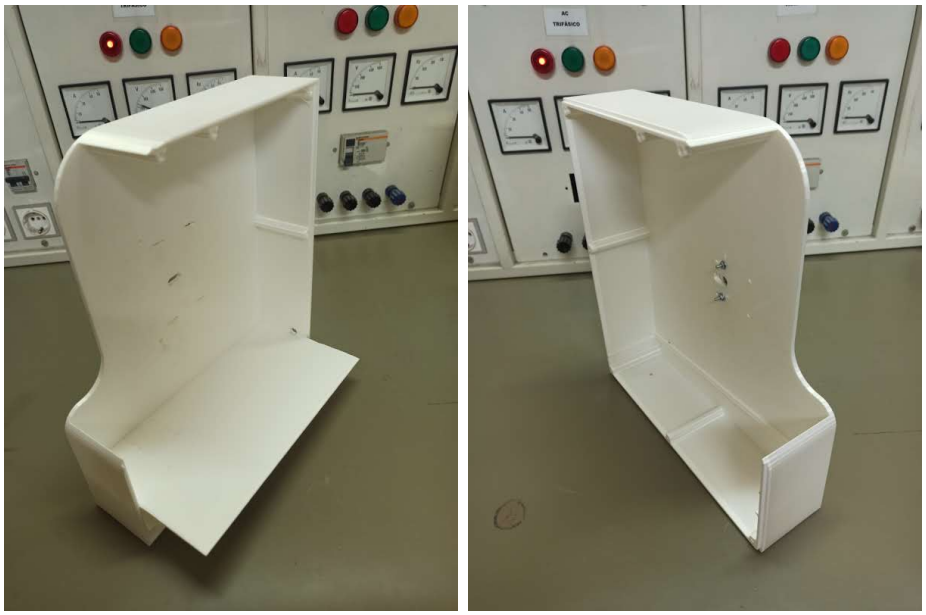
and on the other side of the box is add a bearing so that the ends of the shaft rest on it.

(a) (b)

Figure 3.16: Bearing

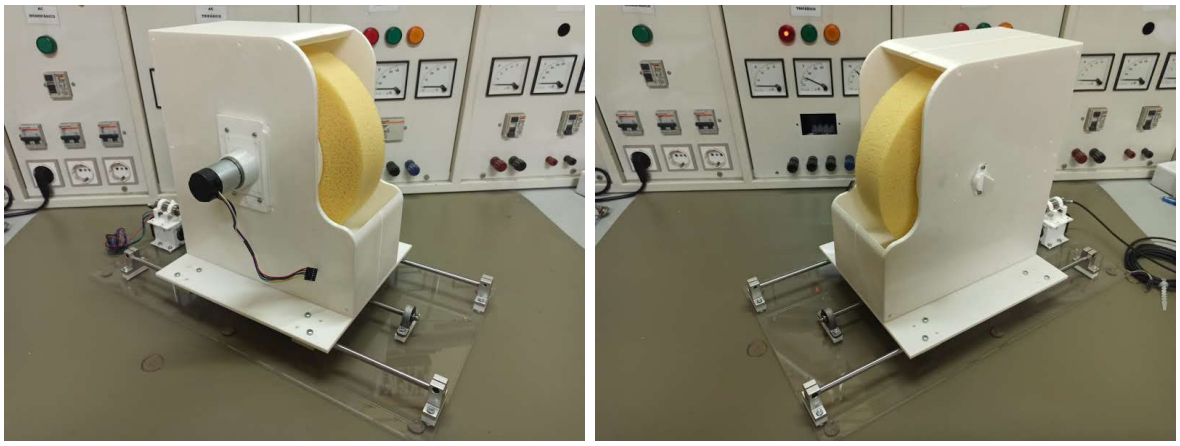
Once this is finished, the two sides of the box are joined together and fixed to the

base.



(a) (b)

Figure 3.17: a) Left side of the box that supports the sponge. b) Right side of the box.



(a) (b)

Figure 3.18: System assembled.

Once the entire system is assembled, the devices and their subsystems are interconnected.

3.2 Measurement and Actuation

This section will describe the interconnections and circuits used in the hardware integration. The basic architecture of the electronic systems used is shown in the Figure 3.19, it is composed of a microcontroller, the actuators and a series of sensors.

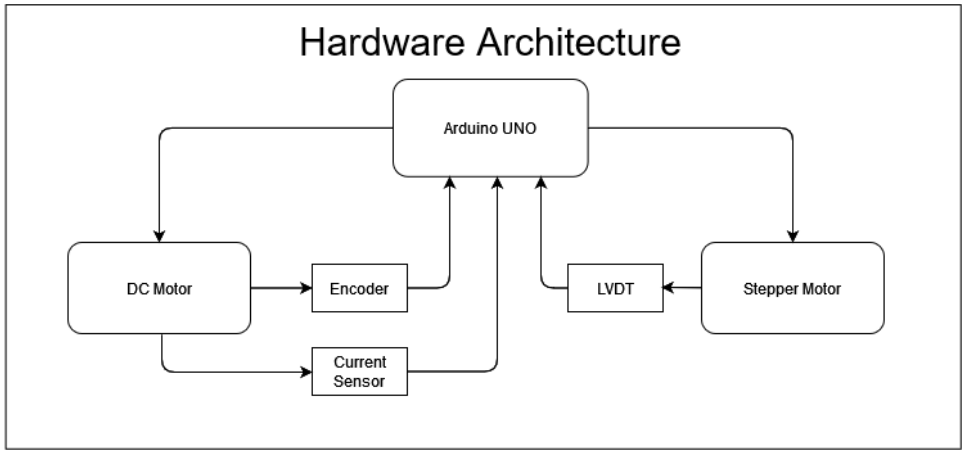


Figure 3.19: Basic electronic architecture overview.

For data acquisition and processing, an Arduino UNO Rev 3 microcontroller based on the AT-mega328P microchip was used [12], which has sufficient capacity for this stage of the project. The sampling frequency of the speed, current and position, together with the PI controller of the DC motor, is 100 Hz, a frequency necessary for the system response to be fast enough in relation to the process.

3.2.1 Speed Control

One of the parameters to control, is the angular speed at which the sponge rotates. It is sought to maintain, even when a load is applied, a constant speed in order to obtain an even finish. The figure 3.20 shows a block diagram of the process that is carried out for this control.

The actuator to be applied to rotate the sponge is a DC motor with encoder, model No.GB37Y3530-12V-251R. It is a 12 V motor with a 43.8:1 gearbox with an integrated

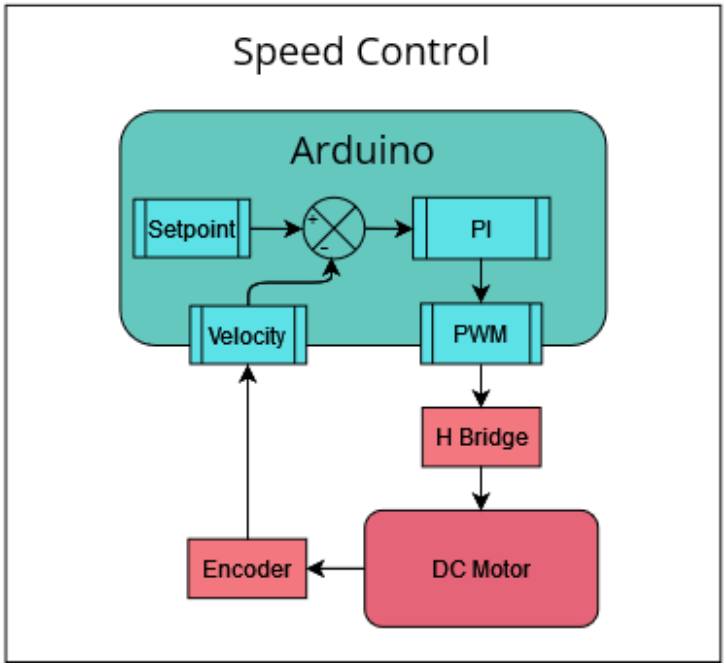


Figure 3.20: Speed control process.

quadrature encoder that delivers 16 pulses per revolution of the motor shaft, which corresponds to 700 pulses per gearbox output shaft [13]. The motor is powered by a 12 Vdc power supply and the arduino supplies the 5 V necessary for the quadrature encoder.

A quadrature encoder is an incremental encoder with 2 out-of-phase output channels. Each channel provides a specific number of equally spaced Pulse Per Revolution (PPR) and the direction of motion is detected by the phase relationship of one channel leading or trailing the other channel. These tracks or channels are coded ninety electrical degrees out of phase, as indicated in the Figure 3.21, and this is the key design element that will provide the quadrature encoder its functionality.

When more resolution is needed, it is possible for the counter to count the leading and trailing edges of the quadrature encoder's pulse train from one channel, which doubles (x2) the number of pulses per revolution. Counting both leading and trailing edges of both channels (A and B channels) of a quadrature encoder will quadruple (x4) the number of pulses per revolution (Figure 3.22) [14].

The encoder integrated in the motor has two output channels, this is useful in the case

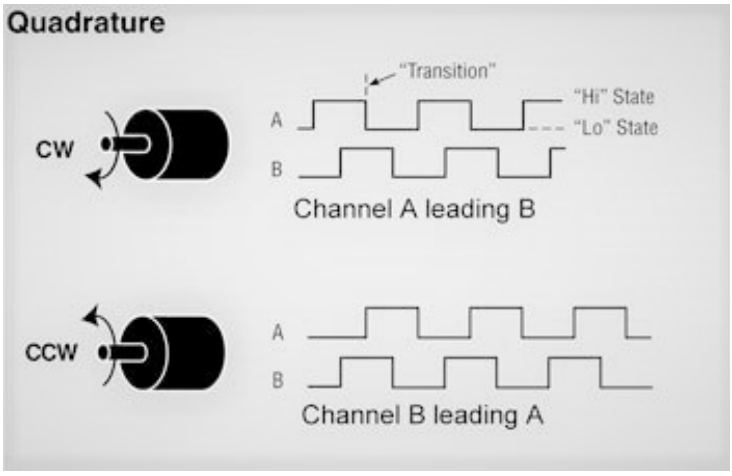


Figure 3.21: Determination of direction in a quadrature encoder [14].

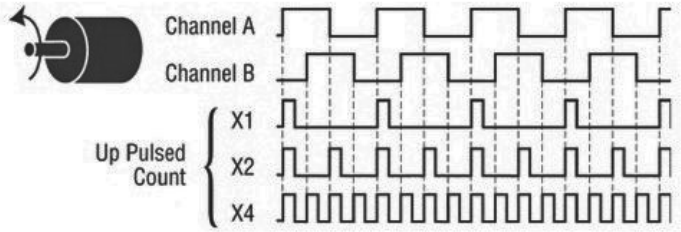


Figure 3.22: Achieving higher resolution with Quadrature Encoders [14].

of needing to measure the direction of rotation of the motor or in the need of precision increase, but since in this case the direction of rotation will always be the same, we only use channel A, this channel is connected to pin 5 of the digital I/O of the microcontroller. The number of pulses that the encoder sends to the Arduino are counted, at a sampling rate of 10 kHz, frequency that allows to not miss any transition. With this count, every 10 ms, the value of the angle that the sponge rotate is obtained, and by first-order integration we get the angular velocity. Once the current speed is determined, it is compared with a setpoint to pass it through a PI controller.

Once the controller performs the calculation, it generates a control variable that is sent through the Pulse Width Modulation (PWM) output 11 to the ENA pin of the H-bridge, which is one L298N Dual H-Bridge Motor Driver (Figure 3.23). It is a high voltage, high

current dual full-bridge driver designed to accept standard Transistor-Transistor Logic (TTL) levels and drive inductive loads such as relays, solenoids, DC and stepping motors [15].

The H-bridge is a simple concept that enables the polarity of the supply being applied to a brushed motor to be swapped. In combination with a PWM signal, the motor's rotational speed and direction can also be controlled [16].

The direction of rotation is determined by pins 6 and 7 that are connected to the IN1 and IN2 inputs of the H-bridge. These always remain the same since, as mentioned above, the direction of rotation does not change.

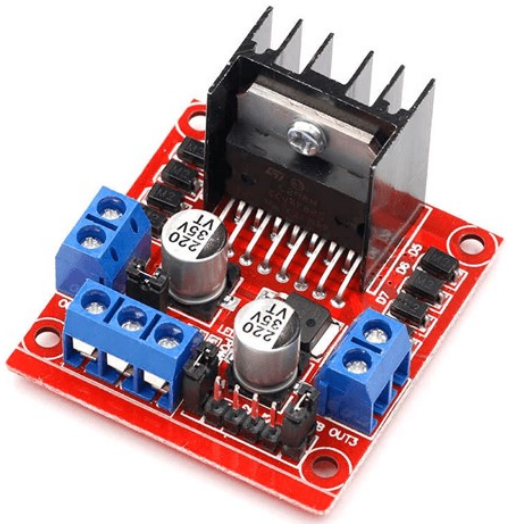


Figure 3.23: L298N Dual H-Bridge Motor Driver.

Finally, the output Out 1 is connected to IP + of the current sensor, in this way the current that circulates through the motor also passes through the sensor to be measured, and OUT 2 to Motor - of the DC motor, Motor + is connected with IP- to close the circuit.

3.2.2 Current Measurement

The current of the DC motor was measured by an ACS712 current sensor (Figure 3.24). The ACS712 uses a Hall effect sensor to output a voltage, providing information of the

current flowing through its copper conduction path. The current sensor provides electrical isolation between the device to be measured and the microcontroller, allowing the ACS217 to be used in applications where isolation is required without the need for optocouplers or other isolation techniques [17].

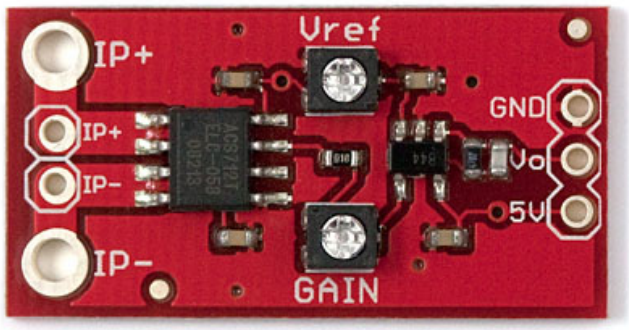


Figure 3.24: Current sensor breakout board.

To carry out a calibration, the value of the reference voltage was adjusted through a trimmer V_{ref} to leave it in the middle of its range, and in the same way the sensitivity value was adjusted with trimmer GAIN so that it had a maximum excursion. The sensor output signal is passed through a low pass filter with a cutoff frequency of 20 Hz, to remove the PWM frequency component, which is 120 Hz, then it is read by the analog input A0 of the Arduino. Using a conversion constant, the Analog to Digital Converter (ADC) value is passed to the equivalent value in voltage, once the voltage is obtained, the equation is applied with the values measured in the calibration to finally have the amount of current that is consumed by the DC motor.

3.2.3 Position Control

The position is another important parameter to take into account, this measurement is carried out thanks to the displacement sensor integrated with a LVDT from Solarton

Metrology, model VG/10/S [18]. This model has a voltage output linearly proportional to the compression suffered by the sensor spring, the proportion is 1 V for each mm of travel, with a range of 0 to 10 mm. It is coupled to the base of the machine and reads the position of the box in relation to the base, the sensor signal passes through a Signal Conditioning Circuit (SCC), which has an approximate gain of 0.5, bringing the 10 V from the LVDT output to a suitable voltage for the microcontroller to read it. Finally the value of the distance traveled in millimeters is calculated through the equation 3.2 obtained from the calibration of the sensor (appendix A).

$$V_{out} = 0.464dis \tag{3.1}$$

$$dis = 2.155V_{out} \tag{3.2}$$

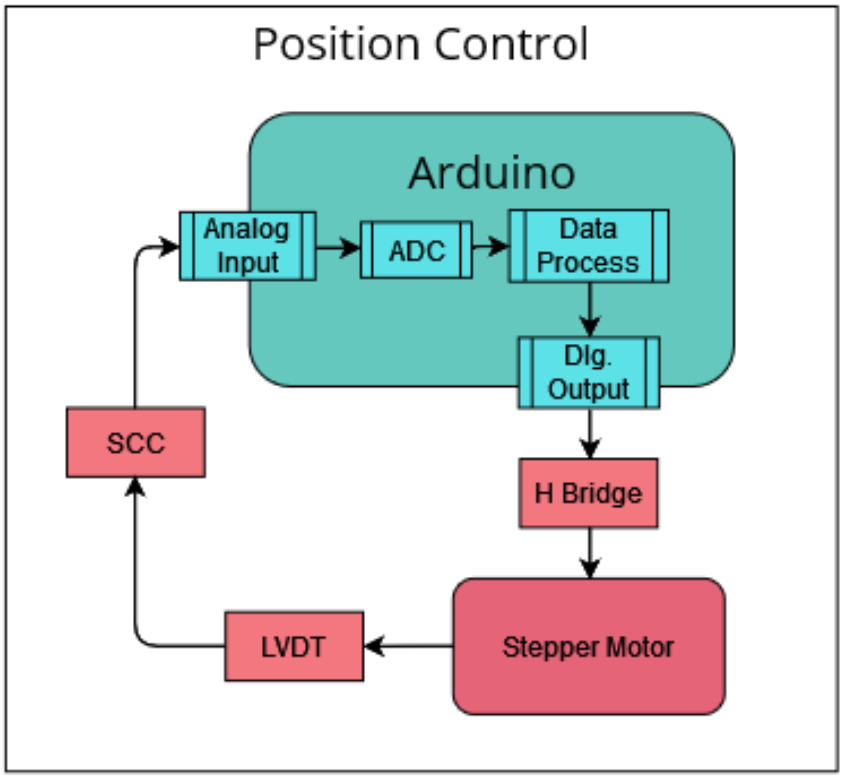


Figure 3.25: Position control process.

Once the measurement of the position is obtained, it is necessary to carry out the position control (Figure 3.25). This is achieved with the use of a stepper motor, along with a lead screw and two guides with their bearings. For the horizontal movement of the system, a bipolar stepper motor is used. This is a motor with step angle 1.8 deg and size 42x42x48mm. It has 4 wires, each phase draws 0.64 A, with holding torque 60 Ncm [11].

To use this motor, a specific driver for stepper motors is necessary, such as the Driver TB6560 Stepper Motor Controller (figure 3.26), it is an adjustable motor controller that can meet the needs of the application, is easy to use and uses high-speed coupling to ensure that it does not influence losses in the control of the steps that the motor provides. It works by providing a digital pulse of step and direction. It comes with a heat sink which allows the module to be kept at a suitable temperature and its operation is not affected.

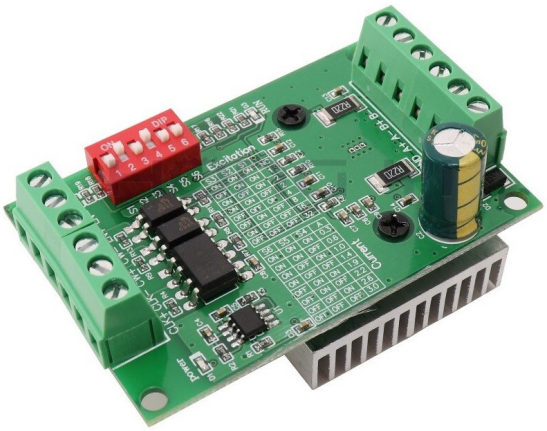


Figure 3.26: Driver TB6560 Stepper Motor Controller.

The stepper motor coils are connected to the driver. A+, A-, B+ and B- of the motor with their homonyms in the driver. From the microcontroller, outputs 2 and 3 are connected to CW+ and CLK+ of the driver, respectively. These signals give the frequency of the steps and the direction of rotation.

Two modes were implemented in the position control. At first the position is kept fixed at 4 mm while the ceramic piece approaches the sponge, until the speed of the sponge reaches the stationary state, once the speed is stable, it is changed to the second

mode where it begins to adjust the position to keep the current constant. For this, it is programmed so that it maintains the current of the DC motor within a pre-established range, therefore if the current increases beyond the maximum limit, the box moves back to reduce the pressure that is exerted, in the opposite case where the current is less than the minimum limit, the stepper motor pushes the box to keep the current within range. The linear speed (v) is given by the equation 3.3, where f_s is the frequency at which the steps occur, and R_s is the resolution of each step. These displacements also have a limit, the box cannot move more than 10 mm backwards to maintain the integrity of the LVDT sensor, nor less than 1 mm, since it goes out of the measurement range of the sensor.

$$v = f_s R_s [mm/s] \tag{3.3}$$

Summarizing, the diagram in the Figure 3.27 shows the interconnections of the systems. The DC motor together with its encoder is connected to the H bridge and the current sensor, which has a low pass filter. The stepper motor with its driver. And finally the LVDT connects to an SCC. All of these systems communicate with the Arduino which is connected to a computer via Universal Serial Bus (USB).

3.3 Controller design

In this section the design of the controllers used will be explained, such as the PI controller, the low pass filter and the SCC.

3.3.1 PI controller

The Proportional–Integral–Derivative (PID) controller is the most common form of feedback. It was an essential element of early governors and it became the standard tool when process control emerged in the 1940s. In process control today, more than 95% of the control loops are of PID type, most loops are actually PI control. PID controllers are today found in all areas where control is used.

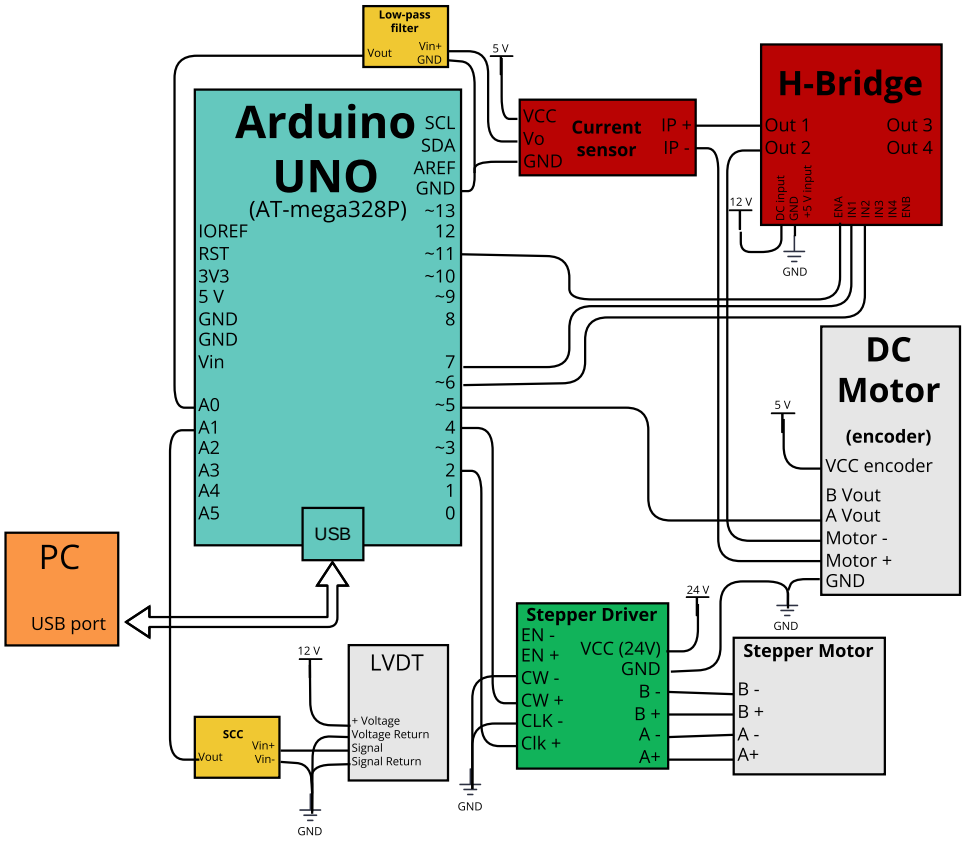


Figure 3.27: Interconnection of electromechanical systems.

The PID algorithm is described by:

$$u(t) = K_P \left(e(t) + \frac{1}{T_i} \int_0^t e(\tau)dt + T_d \frac{de(t)}{dt} \right) \tag{3.4}$$

where y is the measured process variable, r the reference variable, u is the control signal and e is the control error ($e = r - y$). The reference variable is often called the set point. The control signal is thus a sum of three terms: the P-term (which is proportional to the error), the I-term (which is proportional to the integral of the error), and the D-term (which is proportional to the derivative of the error). The controller parameters are proportional gain K_P, integral time T_i, and derivative time T_d. The integral, proportional and derivative part can be interpreted as control actions based on the past, the present and the future [19].

As mentioned in the previous paragraph, PI controllers are the most used when it comes to controlling a plant with a first-order transfer function. That is why for this work a PI controller is implemented, leaving the equation 3.4 as follows:

$$u(t) = K_P e(t) + \frac{K_P}{T_i} \int_0^t e(\tau)dt \tag{3.5}$$

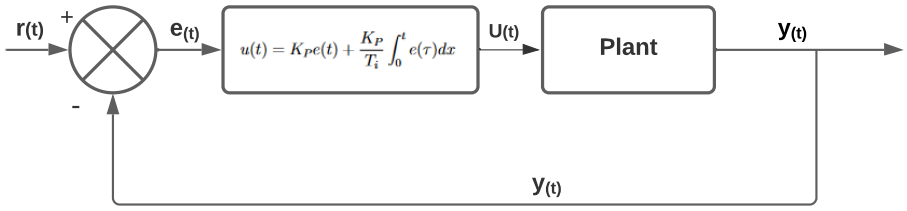


Figure 3.28: Scheme of a PI controller.

To implement the PI controller in the microcontroller it is necessary to discretize the previous equation, for that there are several methods, one of them through the numerical approximation. The most commonly used numerical approximations to discretize the differential equation of an analog system are:

- Euler method
- Backward Euler method
- Trapezoidal rule

For the integral action we will use the trapezoidal method, according to this rule, the integral 3.6 is approximated by the equation 3.7

$$u(t) = \frac{K_P}{T_i} \int_0^t e(t)dt \tag{3.6}$$

$$u_{(nT)} \cong K_i \frac{T}{2} \sum_{i=0}^{n} \left[e_{(i-1)T} + e_{iT} \right] \tag{3.7}$$

this expression can be reduced

$$u_{(nT)} = K_P e_{nT} + u_{(n-1)T} + K_i \frac{T}{2} \left[e_{nT} + e_{(n-1)T} \right] \tag{3.8}$$

This rule (equation 3.8) uses for the calculation of the integral at the current instant u_{nT}, the result of the integral in the previous step $u_{(n-1)T}$ and the current values e_{nT} and previous values $e_{(n-1)T}$ signal.

The process of defining the optimal gains for P, I, and D to obtain an optimal response from a control system is called tuning. There are different adjustment methods, of which the "guess and verify" was the one implemented, for this a series of tests were carried out, varying the parameters of the controller and analyzing its response to the step until get an answer that suits the system requirements. In Table 3.1 the values obtained are detailed.

PI control	
Parameter	Value
K_c	15.4
T_i	0.260
T_d	0

Table 3.1: Controller parameters

3.3.2 Low Pass Filter

As already mentioned in the previous section, the chosen current sensor was ACS712. On one side, the current that circulates through the DC motor enters. And on the other side, a voltage signal proportional to the current comes out through Vo, and this is passed through a RC low-pass filter (Figure 3.29) with a cutoff frequency of 20 Hz given by the equation 3.9. Whose values of R = 8.2 kΩ and that of C = 1 μF.

$$f_c = \frac{1}{2\pi RC} [Hz] \tag{3.9}$$

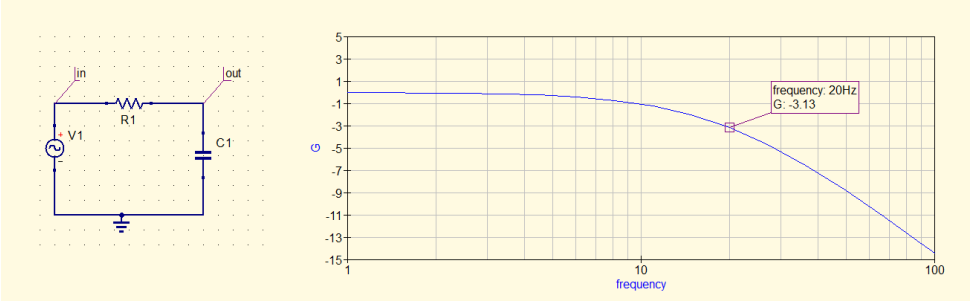


Figure 3.29: RC Low Pass Filter.

The output of the low pass filter is connected to the analog input A0 of the arduino.

3.3.3 Signal Conditioning Circuit

The LVDT sensor measures the displacement of the box with respect to the platform and sends the signal to the SCC, which is a voltage follower followed by an RC filter with gain less than 1.

Initially, the use of an instrumentation amplifier was considered as it meets several desired characteristics for a conditioning circuit, such as high input impedance and high common mode rejection. For this case, an INA114 package was considered. But it was necessary to discard this idea since this integrated circuit can only have a gain greater than unity, being that in this case an approximate gain of 0.5 is required. For this reason it was decided to continue with a circuit based on discrete components.

First a voltage divider circuit was considered, together with a voltage follower based on an OpAmp, as seen in the figure 3.30.

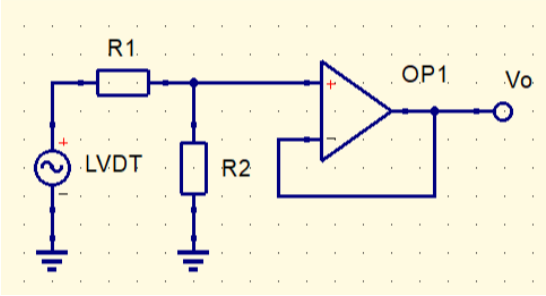


Figure 3.30: First version of the SCC.

But the idea was discarded since in this way the LVDT is over-saturated by requesting an excess of current. For this reason it was decided to pass the voltage follower circuit to the entrance (Figure 3.31), in this way the high input impedance of the operational prevents too much current from being requested from the sensor. Also, an antialiasing filter was implemented, which cleans the signal, before sampling, from frequencies above $Fs/10$, with F_s being the sampling frequency.

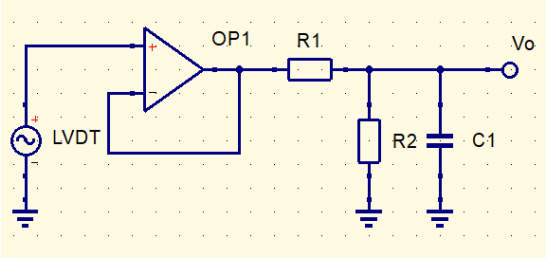


Figure 3.31: New version of the SCC.

Different OpAmps were analyzed taking into account their characteristics, among which were TLC272, the TL07x family, and OP270. The TLC272 was chosen as it has a similar power range to the power supply and can be powered from a single source.

Once the measurements for the calibration were carried out, the maximum output of the sensor obtained was 10.6 V, and since the input of the ADC admits a maximum voltage of 5 V, it is necessary that the gain of the circuit to be:

$$G_m = \frac{5V}{10.6V} = 0.46 \tag{3.10}$$

To obtain the gain of the circuit first it is necessary to obtain the equivalence of the impedance in Laplace domain.

Where

$$Z_1 = R_1 \tag{3.11}$$

and

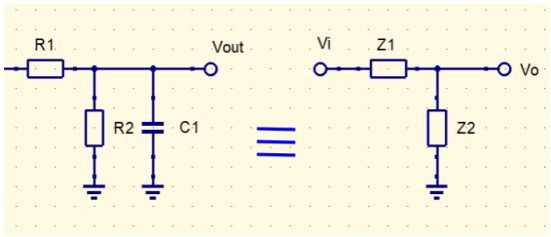


Figure 3.32: Equivalence of the impedance.

$$Z_2 = R_2 \parallel \frac{1}{Cs} = \frac{R_2 * \frac{1}{Cs}}{R_2 + \frac{1}{Cs}} = \frac{R_2}{R_2 * Cs} \tag{3.12}$$

therefore

$$G_m = \frac{V_o}{V_i} = \frac{Z_2}{Z_1 + Z_2} = \frac{\frac{R_2}{R_2 * Cs}}{Z_1 + \frac{R_2}{R_2 * Cs}} \tag{3.13}$$

$$G_m = \frac{\frac{R_2}{R_1 + R_2}}{(R_1 \parallel R_2) * Cs + 1} \tag{3.14}$$

Replacing $s = j\omega = j\pi 2f$, with an angular frequency tending to 0

$\omega \to 0$

$$G_m = \frac{R_2}{R_1 + R_2} \tag{3.15}$$

$$G_m = \frac{\frac{R_2}{R_1 + R_2}}{(R_1 \parallel R_2) * C * j\omega} \tag{3.16}$$

to calculate the values of R1 and R2 for a gain in direct voltage of $G_m = 0.46$, it defines $R_1 = 10k\omega$ and obtain

$$R_2 = \frac{G_m}{1 - G_m} * R_1 = 8.51k\Omega \simeq 8.66k\Omega \tag{3.17}$$

At the cutoff frequency, the resonance of the RC filter occurs, where the values of the resistance and the capacitive reactance at that frequency are equalized.

$$R = X_c \Rightarrow R_1 \parallel R_2 = \frac{1}{2\pi f C} \tag{3.18}$$

Since the sampling frequency is going to be $fs = 100Hz$ the cutoff frequency need to be $f_c = f_s/10 = 10Hz$. The value of C is obtained by means of the equation

$$C = \frac{1}{2\pi f R_1 \parallel R_2} = 3.51\mu F \simeq 3.3\mu F \tag{3.19}$$

With this value, all the values of the components to assemble the circuit are ready.

With this model (Figure 3.33) made in Qucs Studio 4.4.2, a simulation was carried out to verify that the calculated values are correct. The frequency response to the step was performed, obtaining the bode graph of the gain as a function of the frequency.

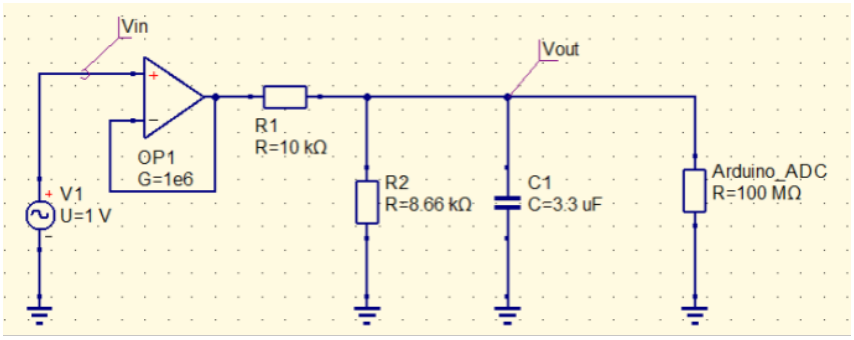


Figure 3.33: QUCS circuit of the SCC for simulation

In the graph (Figure 3.34) it can be seen that when the frequency is very low the gain is -6.67 dB which is equal to a gain of 0.46 times, it can also be seen that at a drop of -3 dB we obtain the cutoff frequency $f_c = 10.4\ Hz$, the difference is due to the fact that a commercial capacitor value was chosen, but for practical purposes it does not affect the operation of the filter.

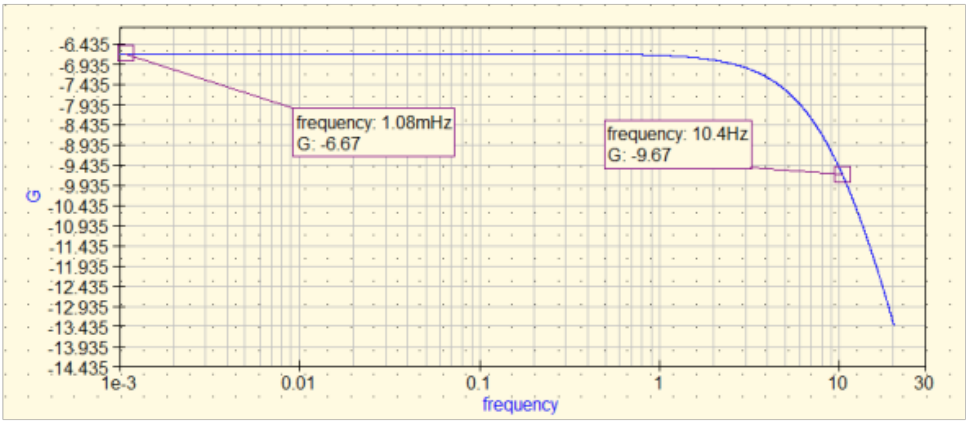


Figure 3.34: Bode diagram of the SCC.

3.4 Firmware

In this section the logic used in the programming of the microcontroller will be described. Starting with the interruption section, where samples are taken at a certain frequency determined by software interruptions. After a certain number of interrupts, the flags are activated so that the main program stages can be executed. There are two main loops, one that is performed at a frequency of 100 Hz that is responsible for calculating and controlling the speed of rotation of the sponge, along with the measurement of the current and the position of the box. The second loop runs at an approximate frequency of 660 Hz and is responsible for generating the signal to control the stepper motor.

The diagram in the figure 3.35 represents the program implemented in Arduino. On the left side are the declarations of the variables to be used together with the configuration of the microcontroller registers. On the right side you see the section that is executed on each interrupt.

Starting with the program, the first thing is to initialize the variables necessary for the correct operation, followed by the declaration of the inputs and outputs of the Arduino, both analog and digital.

Then the registers are configured, starting by changing the frequency of the PWM output. For this, the prescaler of Timer 2 is modified, since it is a Timer that is not used for anything else. It is configured so that the PWM switching frequency is 120 Hz, this

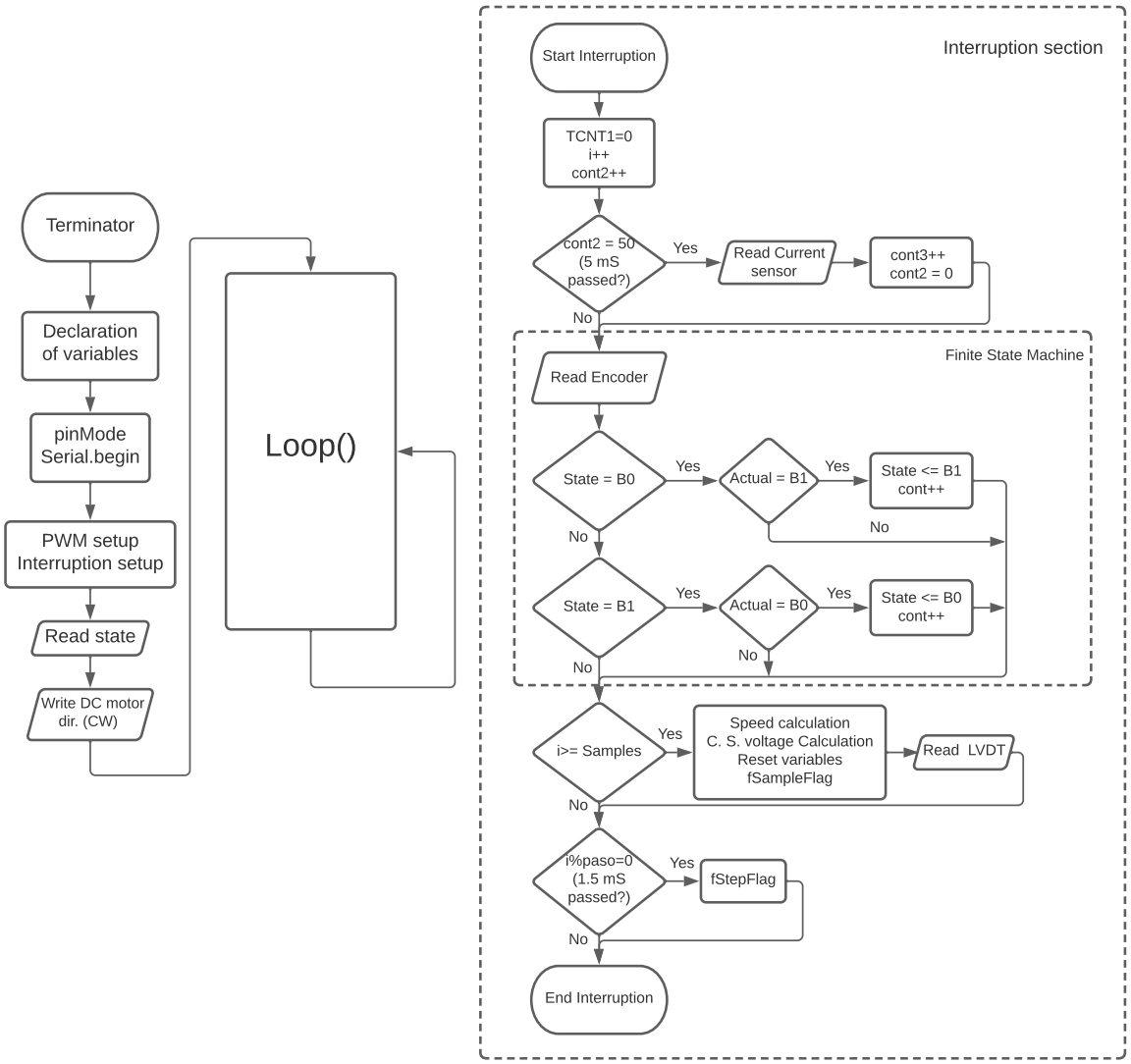


Figure 3.35: Variable declaration and register configuration (Left). Interruption section (Right).

is done since having a low switching frequency, the torque generated by the DC motor is higher. For this, the instruction is used,

```
TCCR2B = TCCR2B & B11111000 | B00000110;
```

With the code programmed in Arduino, it is intended to have an interruption frequency of 10 kHz, that is, an interruption every 100 us. To have a precise frequency, the Arduino Timers were used in conjunction with interruptions.

The arduinos have different amount of Timer depending on the model, in the case of arduino UNO, has three Timers, Timer0, Timer1 and Timer2 which can be programmed through their registers. These work as a counter with a frequency of 16 GHz, this frequency can be scaled using prescalers that are included in the microcontroller.

Interrupts can occur for different events, in this case the type of interruption is by compare match, in which it is written a value in the comparison variable, and when the counter is equal to this value, the interruption occurs.

The first thing is to determine which Timer to use, for this the value of the variable to be compared with the timer counter is calculated. This is calculated as follows:

$$T_s = \frac{f_{clk}}{prescaler} \tag{3.20}$$

$$P_t = \frac{1}{T_s} \tag{3.21}$$

$$C = \frac{P_{int}}{P_t} \tag{3.22}$$

It gives a value of C = 1600, which rules out the 8-bit timers, Timer0 and Timer2, leaving only Timer1.

Once the prescaler and the value of the variable to be compared have been determined, the time registers are configured. To control the timers there are two main registers, the TCCRA and the TCCRB, each with the number of the timer. So for timer 1 it is TCCR1A and TCCR1B.

Now, it changes the TCCRB register. Here, what matters are the first 3 bits which are used to define the prescalar value. Using the CS10 CS11 and CS12 bits, we can disable the prescalar or set it to 1, divided by 8, 64, 256, 1024 or even use an external clock source.

To activate the compare match interruption it is set the TIMSK register. Setting the bits 1 and 2, can be enabled the time compare interrupt on the value defined on registers

OCRA and OCRB. So these OCR registers will tell when to make the interruption. For example, if its set OCR1A to be 20000, when timer 1 reaches 20000 it will create an interruption.

The instructions are as follows

```
cli();                  // Stop interrupts

TCCR1A = 0;             // Reset entire TCCR1A to 0
TCCR1B = 0;             // Reset entire TCCR1B to 0

TCCR1B |= B00000001;  // Set CS12 to 1 so we get prescaler 1

/* We enable compare match mode on register A*/
TIMSK1 |= B00000010;  // Set OCIE1A to 1 so we enable compare match A

OCR1A = 1485;           // Calculations (for fs = 10kHz):
                        // System clock 16 Mhz and Prescalar 1;
                        // Timer 1 speed = 16Mhz/1 = 16 Mhz
                        // Pulse time = 1/16 Mhz =  62.5 ns
                        // Count up to = 100us/62.5ns = 1600
                        // C = 1485 for 10k
sei();                  // Enable back the interrupts
```

On the right side of the Figure 3.35, the section of what happens in the interruption is shown.

It begin by resetting the variables and increasing the necessary counters. In the first decision block, it is verified if 200 ms have passed since the last current sample, if it is true, a new sample is taken.

Then it enter in a finite state machine, where the pulses sent by the encoder are counted, each time a change of state is read, a counter is increased by one.

Every 100 ms the LVDT signal is sampled and the transformations of the variables are implemented, and the "100 ms flag" is activated so that the loop in the main program is executed.

Finally, every 1.5 ms the second flag for the other loop is activated.

In the main loop (Figure 3.36) there are two secondary loops, each triggered by a different flag. The first loop is in charge of controlling the speed of the DC motor and to measure the current consumed, and it is executed with the flag "*fSampleFlag*". For which a PI controller is used, it has the ability to use the two control terms of proportional and integral influence on the controller output to apply accurate and optimal control. The read value of the speed (Process Variable (PV)) is compared with the one established as the desired value (Set-Point (SP)) to obtain the error. The controller attempts to minimize the error over time by adjustment of a control variable.

Term P is proportional to the current value of the $SP - PV = e$. For example, if the error is large and positive, the control output will be proportionately large and positive, taking into account the proportional gain factor K_P.

Term I accounts for past values of the error and integrates them over time to produce the I term. For example, if there is a residual error after the application of proportional control, the integral term seeks to eliminate the residual error by adding a control effect due to the historic cumulative value of the error. When the error is eliminated, the integral term will cease to grow. This will result in the proportional effect diminishing as the error decreases, but this is compensated for by the growing integral effect.

There are some non-linear aspects that must be considered. The actuators have a proportional band in their operation, where if it goes out of range the control becomes an open loop control, this is maintained until the control variable returns to the linear range. To prevent this from happening, a saturation block is used, where if the control variable goes out of range, it is limited to the maximum and minimum values. The output can be expressed as:

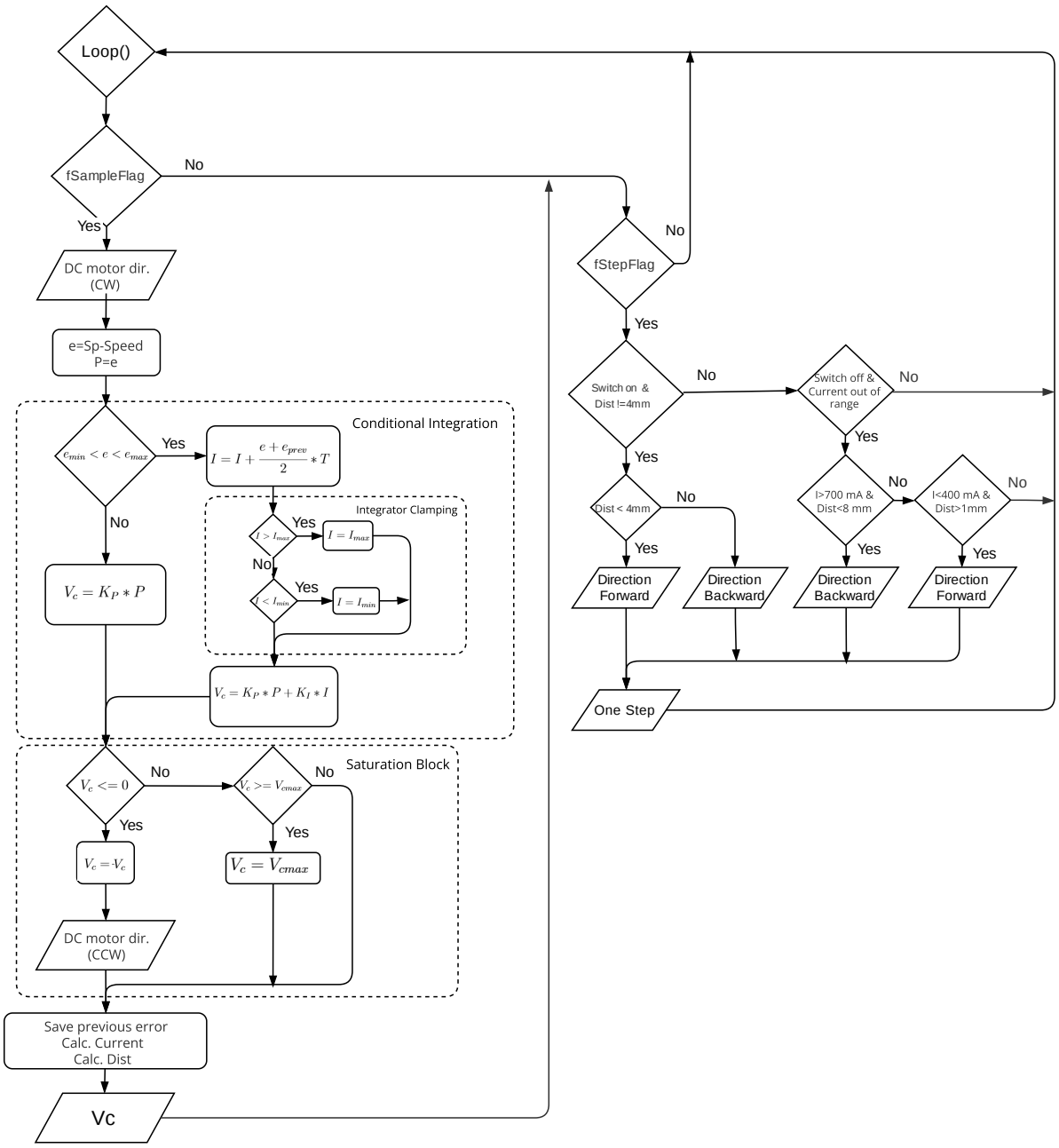


Figure 3.36: Diagram of the Loop() function.

$$V_C = \begin{cases} V_{Cmin}, \text{ if } V_c < V_{Cmin} \\ V_C, \text{ if } V_{Cmin} \leq V_C \leq V_{Cmax} \\ V_{Cmax}, \text{ if } V_C > V_{Cmax} \end{cases}$$

But the above approach would lead to problem if the input e remains non-zero for long and during that period the integrator output keeps on accumulating. This may either introduce delay in response when the input error changes or even lead to roll over of output. To avoid this it is necessary to check the integration process during such situations which is in general known as anti-windup. In this case, two methods are used to reduce these errors [20]. The first is to condition the integration stage so that it only integrates the error for low error values, this helps the process inertia to be lower and the reaction times faster, this is achieved thanks to the conditional integration block. Another method is to limit the integrating action, when the parameter I reaches a very high level.

Once the control variable is obtained, it is sent through the PWM output of the Arduino, and the current and position calculations are made. These values are used in the second loop.

The second loop is activated by the "*fStepFlag*" flag, which is executed every 1.5 ms. This loop analyzes the need to make an adjustment to the position of the box. Since there are two modes of position control, it is first checked whether or not the mode selection switch is active. In the event that it is active, it is seen if the position is outside the preset range, in the event that this is true, it is verified if the position is less than that of the origin. In the true case the direction of movement is set forward, otherwise the direction will be backward.

In the event that the switch is off, it enters the mode where the position is controlled according to the applied force. First, it is verified if the current is outside the desired range, in the event that it is in range, the system skips this loop. On the other hand, if the current is out of range, it is necessary to apply a correction. If the position is less than 8 mm and the current is greater than 700 mA, the direction of movement is set backwards. On the other hand, if the position is greater than 1 and the current is less than 400 mA, the direction will be forward. Finally, a signal is sent to perform a step and the main loop is executed again.

Chapter 4

Obtained Tests

In this chapter, the analysis of a series of tests and controls on the developed systems will be carried out, starting with an analysis of the tuning parameters of the PI controller, followed by a series of tests of the speed and position control systems.

4.1 PI tuning

With the parameters Kc and Ti obtained (Table 3.1), the gain parameters of the PI controller can be determined, knowing that,

$$K_P = K_c = 15.4 \tag{4.1}$$

$$K_I = \frac{K_c}{T_i} = 59.23 \tag{4.2}$$

Once these parameters were obtained, tests were carried out with these values, and with others to compare the results with the step response. In the figure 4.1, tests were carried out with

$$\{K_P, K_I\} = \{114, 400\},$$
$$\{K_P, K_I\} = \{7, 24\},$$
$$\{K_P, K_I\} = \{15, 59\},$$
$$\{K_P, K_I\} = \{15, 100\},$$

for a set-point of 13 rad/s and then increases to 18 rad/s

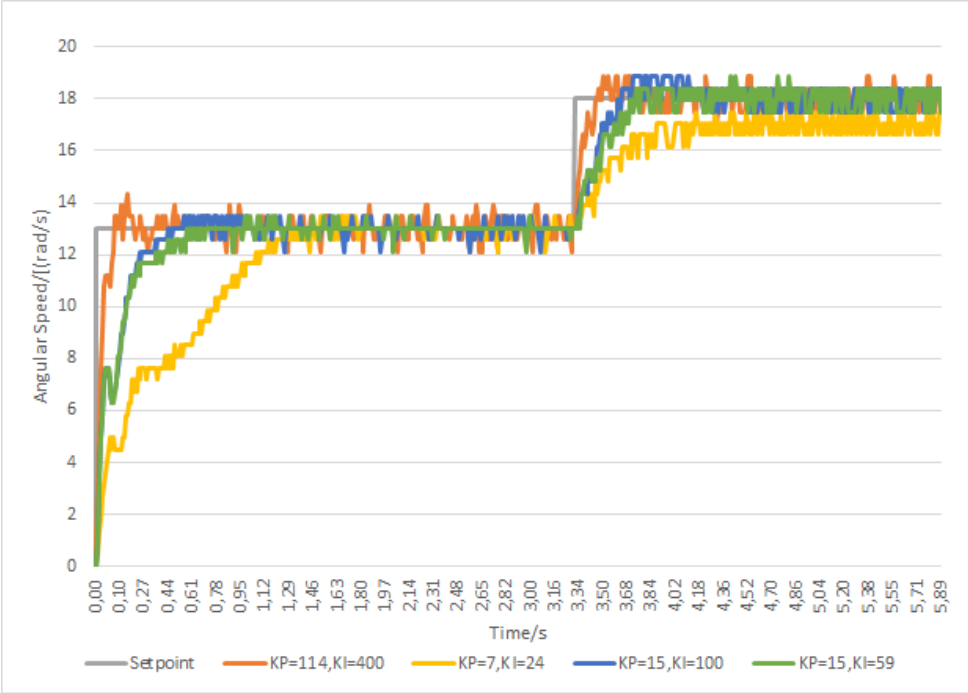


Figure 4.1: Test with different values of K_P and K_I.

The figure shows an enlargement of the figure 4.1 where the difference between the times of each test can be better appreciated.

For the case of $\{K_P, K_I\} = \{114,400\}$, the rise time is much shorter than the others, but a very high over-peak can be seen, which causes an under-damped transient response, increasing the settling time. In the case of $\{K_P, K_I\} = \{7,24\}$ the rise time is very high, leading the system to have a slow response to disturbances.

Now taking the parameters to values close to those obtained with the IMC method,

you can see an improvement in the response, for the case of $\{K_P, K_I\} = \{15{,}100\}$ the rise time is less than the previous case but a small overshoot can be seen. Finally, for $\{K_P, K_I\} = \{15.59\}$, there is a lower rise time than in case 2, but without having an over-peak as in the first and third cases, thus achieving a lower settling time. than in the other cases.

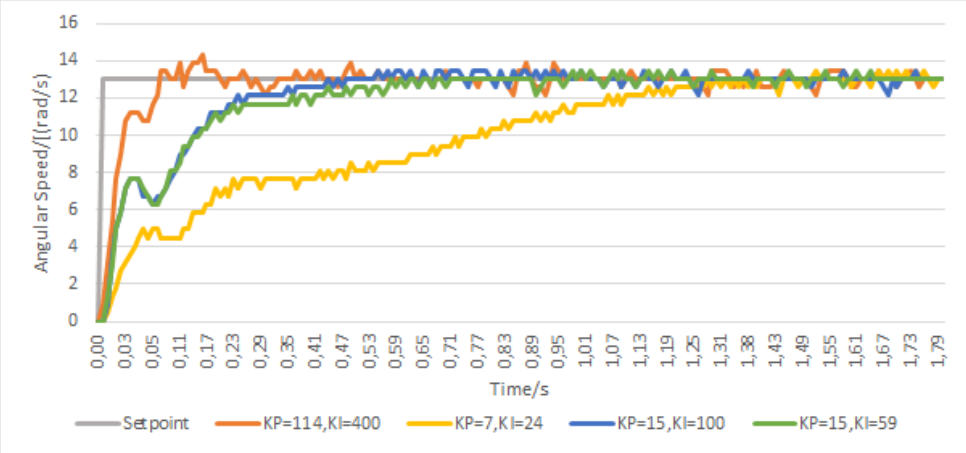


Figure 4.2: Enlargement of Fig. 4.1

4.2 Prototype control

Once the tuning is done, the first stage that was tested was the feedback of the DC motor. It can be seen in Figure 4.3 the step response of the closed-loop system for a setpoint of 15 rad/s, together with the control signal. From this graph it can obtained the value of the time constant $\tau = 0.18\ s$ and the rise time $t_r = 0.46\ s$.

In the Figure 4.4 are the results of a test where a load of a variable value is applied to the sponge and it responds to maintain its constant speed, from here you can also know the time it takes for the system to return to its steady state after a disturbance, the approximate time is about 0.78 s.

Once the proper functioning of the speed control was verified, a test was made on the position control (Figure 4.5). The machine is turned on and it is waited for it to reach the speed of the regime (a), after this a slight pressure was applied to the sponge, where it can

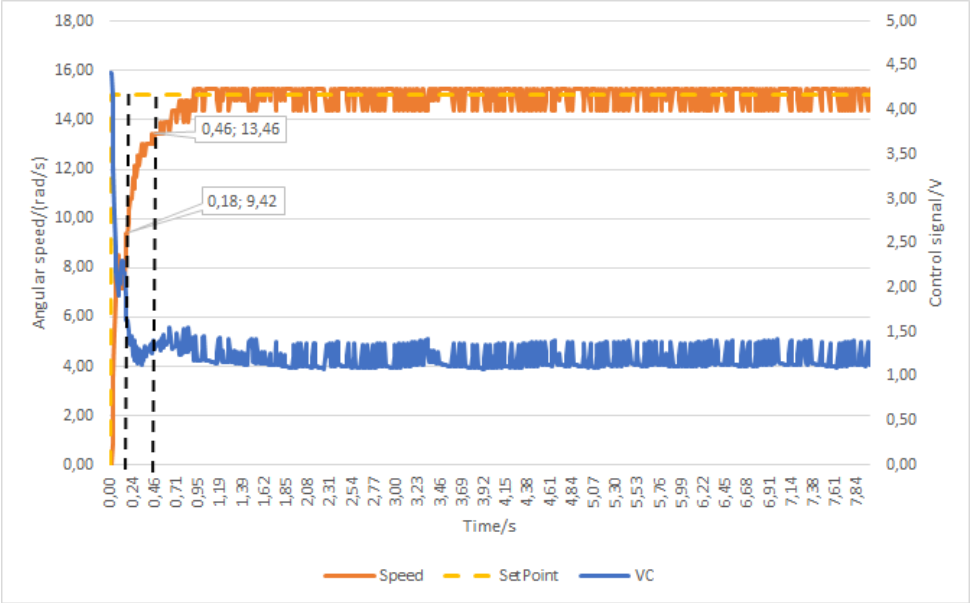


Figure 4.3: Step response of the closed-loop system

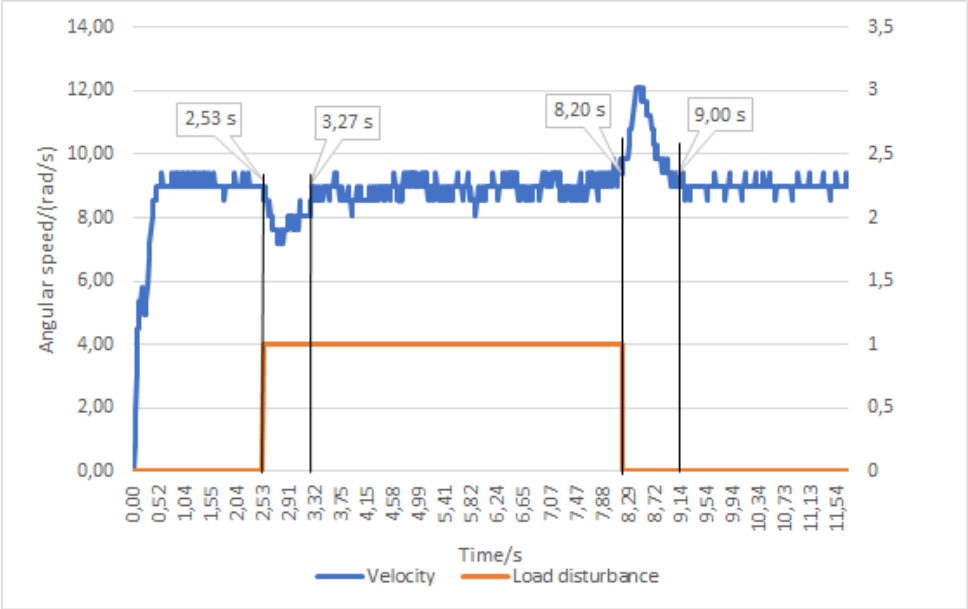


Figure 4.4: System response to load disturbances

be seen that the speed remains constant but the box does not move (b), then the force tracking mode is activated, the box adjusts (in this case it advances since the current was lower than the required level), to keep the current constant and adjusts to force variations (c).

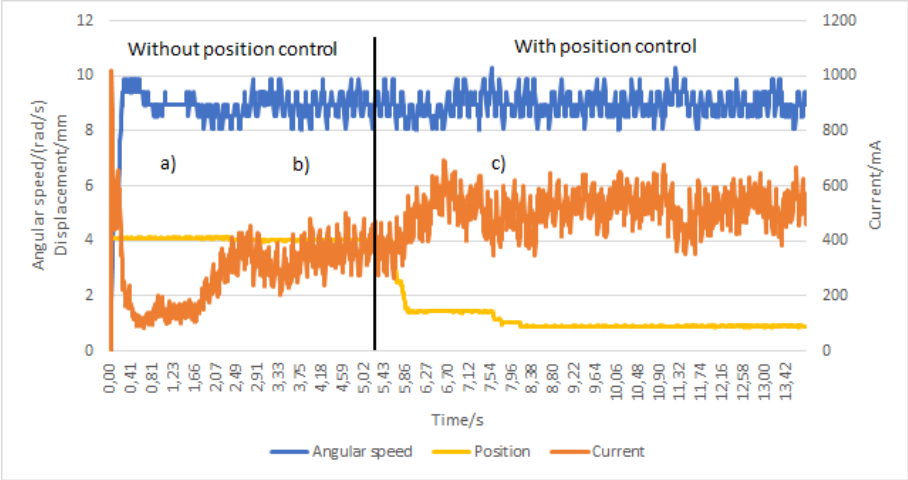


Figure 4.5: Position control test.

Finally, a test was carried out with the entire system (Figure 4.6), at first the system is turned on and reaches a permanent regime, a force is applied to the sponge, here it is observed how the speed regulates itself and how the current begins to increase, only when the current exceeds the established level is when the position of the box is regulated, it goes back so that the current remains at an acceptable level. In a second moment, the piece is withdrawn a little, causing the speed to increase and the current to decrease, but at that same instant the box advances and the speed of the sponge adjusts to the regimen speed, returning to the required levels. The process is repeated but at a higher speed, being able to observe a greater degree of variation and adjustment.

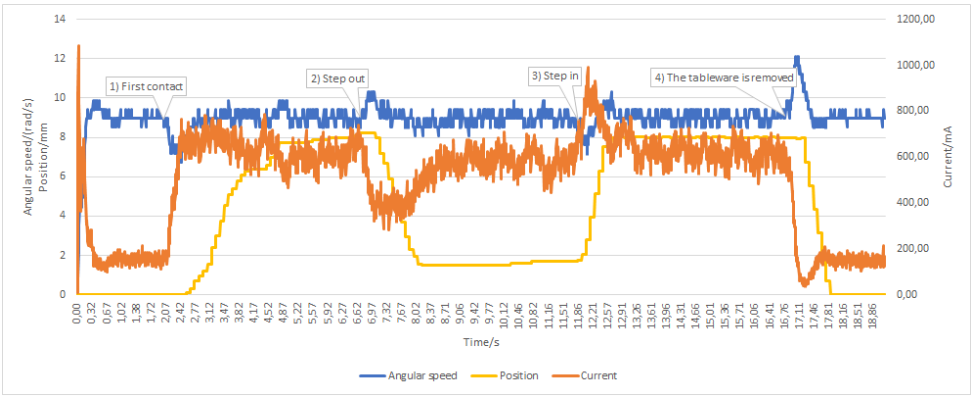


Figure 4.6: Velocity, position and current as a function of time.

After performing these tests they can be analyzed and obtain a series of conclusions.

Chapter 5

Conclusions and future work

5.1 Conclusions

The finishing of the ceramic pieces, after being removed from the molds, is one of the quality processes that the tableware of the GRESTEL industry undergoes. The manufacture of irregular pieces makes the control of the force applied for this finishing extremely important. With this first prototype it was possible to analyze the possibility of using different control and measurement methods to carry out the task. It was observed that the results met the criteria proposed for the control of speed and position. However, it is not possible to be certain of the values necessary for an optimal finish, since this is a first prototype and tests have not yet been carried out in the field and with real conditions.

With this control, together with the measurement of the current and the location of the box with respect to the base, it allows us to adapt to the variations in force that are involved in the task of finishing ceramic pieces.

5.2 Future work

With the use of these degrees of freedom, they allow to have an independent system capable of performing the finishing of both simple and geometric pieces as well as more complex and irregular pieces. However, it is also possible to modify it so that it has

integration with systems such as: A collaborative robot, capable of handling the part, cameras and laser sensors to obtain three-dimensional models to calculate the grinding path, or the use of other sensors such as a force sensor coupled to the robotic arm.

Also another consideration for future prototypes could be working with materials and devices more suitable for the industrial environment, since the materials for this prototype were mainly for laboratory use.

Bibliography

[1] P. da Cerâmica, "Portal da cerâmica, accessed: 2022-01-26," *http://www.ceramica.pt/setor.php.*, 2022.

[2] G. Veiga, P. Malaca, and R. Cancela, "Interactive industrial robot programming for the ceramic industry," *International Journal of Advanced Robotic Systems*, vol. 10, no. 10, p. 354, 2013.

[3] D. Onita, N. Vartan, M. Kadar, and A. Birlutiu, "Quality control in porcelain industry based on computer vision techniques," *2018 International Young Engineers Forum (YEF-ECE)*, no. 5, p. 79, 2018.

[4] A. Frenzl, "A virtualview into the sintering process by thermokinetic simulation," *Digital transformation in the ceramic industry.*, p. 10, 2018.

[5] T. Watanabe, K. Yamazaki, and Y. Yokokohji, "Survey of robotic manipulation studies intending practical applications in real environments: Object recognition, soft robot hand, and challenge program and benchmarking," *Advanced Robotics.*, vol. 31, no. 19-20, p. 1114, 2017.

[6] SLON, *Auto finishing machine for ceramic cups and bowls*, `https://www.ceramicmachine.com/product-details/auto-finishing-machine-for-ceramic-cups-and-bowls/`, Mar. 2022.

[7] CERINNOV, *Robotised finishing unit*, `https://www.cerinnov.com/wp-content/uploads/2020/04/finition.pdf`, Mar. 2022.

[8] B. Kuhlenkötter and X. Zhang, "A robot system for high quality belt grinding and polishing processes," in Jul. 2005, ISBN: 3-86611-038-3. DOI: 10.5772/4680.

[9] W. Wang, F. Liu, and Z. Liu, "Prediction of depth of cut for robotic belt grinding," in Jul. 2017. DOI: 10.1007/s00170-016-9729-3.

[10] M. Alvarez, J. P. Coelho, and J. Gonçalves, "Prototyping and control of an automatic ceramic tableware finishing device," Feb. 2022.

[11] PtRobotics, *Bipolar Nema 17 stepper motor*, https://www.ptrobotics.com/motor-stepper/5784-nema-17-bipolar-18deg-60ncm-85ozin-064a-10v-42x42x60mm-4-wires.html, 2022.

[12] Arduino, *Arduino uno rev 3*, http://store.arduino.cc/products/arduino-uno-rev3, Feb. 2022.

[13] DFrobots, *Metal dc geared motor w/encoder*, https://www.dfrobot.com/product-634.html, 2022.

[14] Dynapar, *Quadrature encoder overview*, https://www.dynapar.com/technology/encoder_basics/quadrature_encoder/, 2022.

[15] ST, *L298 dual full-bridge driver*, https://www.sparkfun.com/datasheets/Robotics/L298_H_Bridge.pdf, 2022.

[16] TOSHIBA, "The hbridge motor driver with a bridge to a wonderful heritage," *https://toshiba.semicon-storage.com/*, 2022.

[17] I. Allegro MicroSystems, *Fully Integrated, Hall Effect-Based Linear Current Sensor with 2.1 kvrms Voltage Isolation and a Low-Resistance Current Conductor, Rev. 7.*

[18] S. Meterology, *Displacement Sensors Probe with direct analogue output Position feedback and linear displacement measurement G type.*

[19] K. J. Åström, *Control System Design*. 2001.

[20] A. Ghoshal and V. John, "Anti-windup schemes for proportional integral and proportional resonant controller," Jun. 2010.

Appendix A

Calibration Certificate

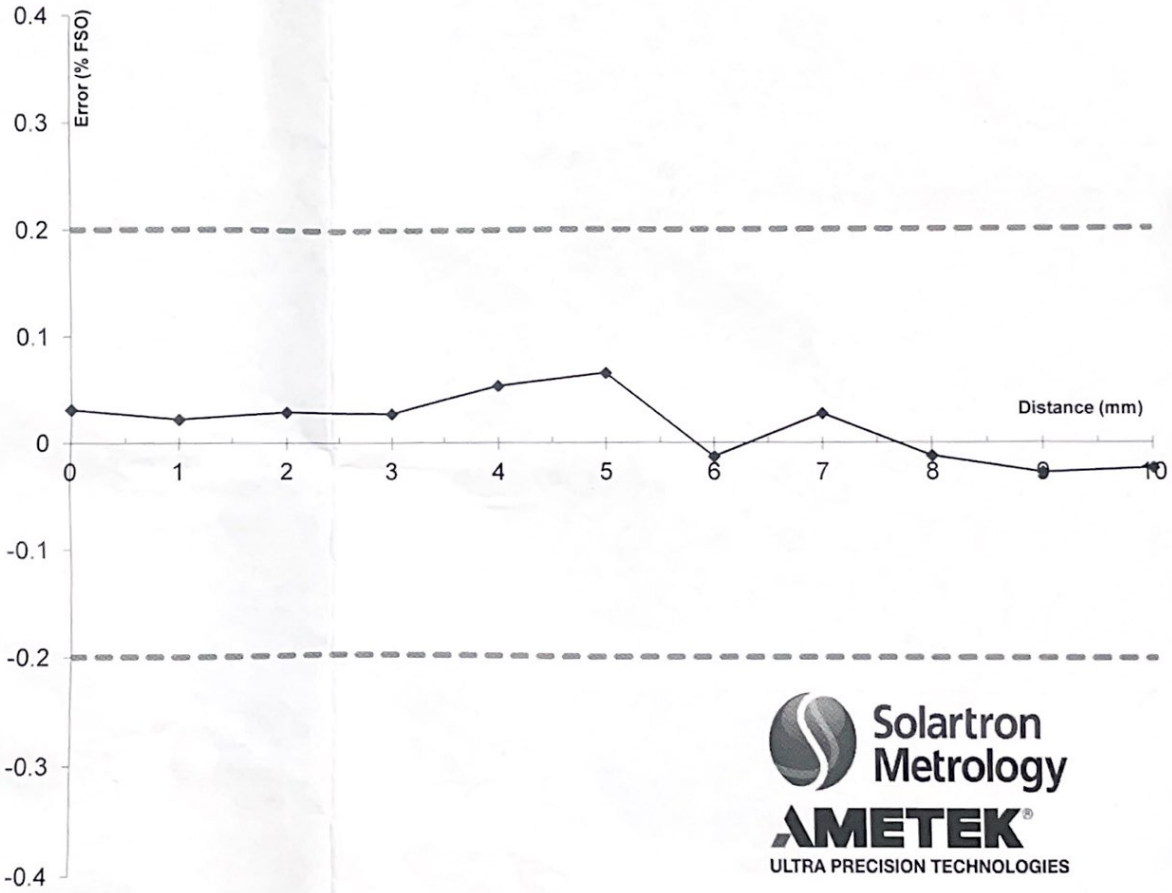
Calibration Certificate

Probe SNo: M931286AK18-03
Probe Part No: 931286
Probe Type No: VG/10/S
Date: 11/05/17,14:18
Operator: S.ROWLAND

Fixture: CAL01-DT3_V4.4
Position Reference: 3687
DMM Ser No: 2732
Test Equipment: 2368

Max Error: 0.065 % Full Scale

Position (mm)	Output (V)	Error (% FSO)
0.000	0.0031	0.031
1.000	1.0022	0.022
2.000	2.0029	0.029
3.000	3.0027	0.027
4.000	4.0053	0.053
5.000	5.0065	0.065
6.000	5.9987	-0.013
7.000	7.0027	0.027
8.000	7.9987	-0.013
9.000	8.9972	-0.028
10.000	9.9976	-0.024



Solartron Metrology
AMETEK®
ULTRA PRECISION TECHNOLOGIES

Appendix B

Code

System program implemented in the Arduino UNO microcontroller

```
1
2   // Declaration of i/o variables //
3
4   //Variables for speed control
5   const int Dir1 = 7;    // IN1 conectada al pin 7
6   const int Dir2 = 6;    // IN2 conectada al pin 6
7   const int ENA = 11;     // ENA conectada al pin 3 de Arduino
8   const int ChA = 5;     // Digital input 5
9
10  //Variables for stepper
11  const int Dir = 4;    // CW+ conectada al pin 4
12  const int Step = 2;    // CLK+ conectada al pin 2
13  const int Run = 12;
14
15  //Analog variables
16  const int lvdt = A1;
17  const int currentSensor = A0;
18
19
20  // Variables of the FSM
21  int valor = 0;
```

```
22  float cont = 0, cont2 =0, cont3 = 0, cont4 = 0;
23  byte state = 0, actual = 0;
24  float Speed = 0;
25  int i = 0, in = 0;
26  bool fSampleFlag = false, fStepFlag = false;
27  float sensorValue = 0;
28
29  //Variable for current control
30  float Vref = 2570; // Output voltage with no current: ~ 2525mV or
      2.5V
31  float sensitivity = 0.901;
32  float voltage = 0, current = 0;
33
34  // Variables for the PID of the speed
35  float Vc = 0;
36  float Sp = 15;    //Speed of the sponge - Max 22.4 rad/seg
37  int sample = 100;
38  float fs = 10000; //10khz
39  float T = sample/fs; // period of 10mS
40  float e = 0, e_prev = 0;
41  float P = 0, I = 0, D = 0;
42  float Kp =15, Ki =59; //Kd = 1; Derivative constant
43
44  //Variable for stepper
45  bool J = LOW;
46  int paso = 15;
47
48  //variable for the LVDT
49  float dist = 0;
50  float dist_prom = 0;
51  float distmm = 0;
52
53 void setup() {
54  //I/O declaration//
55  //DC motor
```

```
56  pinMode (ENA, OUTPUT);
57  pinMode (Dir1, OUTPUT);
58  pinMode (Dir2, OUTPUT);
59  pinMode (ChA, INPUT_PULLUP);
60
61
62  //Currente sensor
63  pinMode (currentSensor,INPUT);
64
65  // Stepper
66  pinMode (Dir, OUTPUT); //dir
67  pinMode (Step, OUTPUT); //steps
68  pinMode (Run, INPUT_PULLUP);
69
70  Serial.begin(115200);
71
72
73  TCCR2B = TCCR2B & B11111000 | B00000110; //Change the frecuency of
      the PWM to 122 Hz
74  cli();                          //Stop interrupts
75
76  TCCR1A = 0;                     // Reset entire TCCR1A to 0
77  TCCR1B = 0;                     // Reset entire TCCR1B to 0
78
79  TCCR1B |= B00000001;            //Set CS12 to 1 so we get prescalar 1
80
81  /* We enable compare match mode on register A*/
82  TIMSK1 |= B00000010;            //Set OCIE1A to 1 so we enable compare
      match A
83
84
85                                  //Calculations (for fs = 10kHz):
86  OCR1A = 1485;                     // System clock 16 Mhz and Prescalar
      1;
87                                  // Timer 1 speed = 16Mhz/1 = 16 Mhz
```

```
88                                  // Pulse time = 1/16 Mhz =  62.5 ns
89                                  // Count up to = 100us / 62.5ns = 1600
     //1485 para 10k
90
91    sei();                        //Enable back the interrupts
92
93
94    state = digitalRead(ChA);     //Read encoder and save in state for the
        first time
95
96    //Set the DC motor to rotate in one way
97    digitalWrite (Dir1, LOW);
98    digitalWrite (Dir2, HIGH);
99
100 }
101
102 void loop()
103     {
104       //Velocity control
105       if(fSampleFlag)
106       {
107            digitalWrite (Dir1, LOW);
108            digitalWrite (Dir2, HIGH);
109
110            e = Sp - Speed;
111            P = e;
112
113            if (e>-15&&e<5)
114              {
115              I = I + (((e + e_prev)/2)*T);
116              if(I>3)          //Max value of the I parameter.
117                I = 3;
118              else if(I<-3) //Min value of the I parameter.
119                I =-3;
120
```

```
121             // D = (e - e_prev)/T;
122
123             Vc = Kp*P + Ki*I;    // + Kd*D;
124             }
125           else
126             Vc = Kp*P;
127
128           if (Vc>=255)  //Max value of the control variable.(Anti-
      WindUp)
129             Vc = 255;
130           else if (Vc<=0){
131             Vc = -Vc;
132             digitalWrite (Dir1, HIGH);
133             digitalWrite (Dir2, LOW);
134             }
135           e_prev = e;
136           current = (voltage - Vref) / sensitivity;
137           analogWrite (ENA,Vc);
138
139
140           //Mesurement of the displacement
141           //reinitialize variables
142           distmm = 0;
143           distmm = 2.155*dist;          //dist_prom *((5*10)
      /(4.64*1024));  //Calibration equation
144           dist = 0;
145
146           fSampleFlag = false;
147       }
148
149       //Position control
150       if(fStepFlag)
151       {
152       if((distmm > 4.15 || distmm < 3.85) && digitalRead(Run))
153         {
```

```
154         if(distmm<4)
155           digitalWrite (Dir, LOW);
156         else //if (distmm >4)
157           digitalWrite (Dir, HIGH);
158 
159         if (abs(4-distmm)<.5)
160           paso = 50;
161         else
162           paso = 15;
163 
164          J = !J;
165          digitalWrite(Step, J);
166          fStepFlag = false;
167        }
168        else if((current<400 || current>700) && !digitalRead(Run))
169        {
170          paso = 15;
171          if ((current>700 && distmm<8)  ) //Move to the Right
172                digitalWrite (Dir, LOW);
173 
174          else if(current<400 && distmm>1)  //Move to the Left
175                digitalWrite (Dir, HIGH);
176 
177          if ((distmm <8 && distmm >1) || (current>700 && distmm<1) ||
    (current<400 && distmm >8))
178          {
179            J = !J;
180            digitalWrite(Step, J);
181          }
182          fStepFlag = false;
183        }
184 
185 
186       }
187 
```

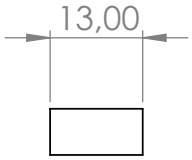
```
188     }
189 ISR(TIMER1_COMPA_vect)
190  {
191     TCNT1  = 0;              //First, set the timer back to 0 so it
        resets
192     i++;                     //Count until reach to samples
193     cont2++;
194
195
196     actual = digitalRead(ChA); //Read encoder and save in actual state
197
198     if(cont2==50) //Read the currente sensor with a fs=200Hz => 5mS
199      {
200       sensorValue += analogRead(currentSensor);
201       cont3++;
202       cont2 = 0;
203      }
204
205     switch (state)            //FSM with 4 states. It changes state,
        when the variable val changes its value to the next state.
206      {                        //Incrementing the variable cont by one.
207        case B0:
208             if(actual == B1)
209             {
210               state = B1;
211               cont++;
212             }
213             break;
214        case B1:
215             if(actual == B0)
216             {
217               state = B0;
218               cont++;
219             }
220             break;
```

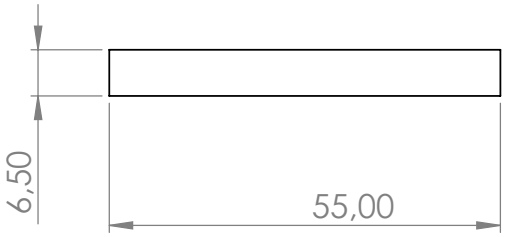
```
221
222         default:
223               break;
224       }
225
226
227
228     if (i>=sample) // 100Hz
229       {
230         Speed = (cont*0.4488);  //calculation of the angular speed
      0.4488 rad/seg = ((2*pi)/1400 pulsos )*100 pulsos/seg
231
232         sensorValue = sensorValue / (cont3);
233         cont3 = 0;
234         voltage = 4.88 * sensorValue;  // The on-board ADC is 10-bits
      -> 2^10 = 1024 -> 5V / 1024 ~= 4.88mV
235
236         dist = (float)analogRead(lvdt)*0.004882816;  //Read the
      converter and multiplied by a constant to transform it to voltage [
      V]
237
238         sensorValue = 0;
239         cont = 0;
240         i = 0;
241         fSampleFlag = true;
242       }
243
244     if(i % paso == 0)// velocidad de los pasos
245       {
246         fStepFlag = true;
247       }
248
249   }
```

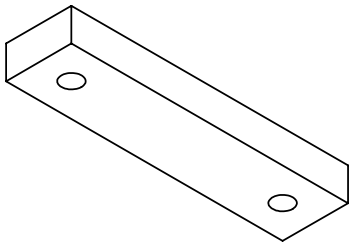
Listing B.1: Arduino Code

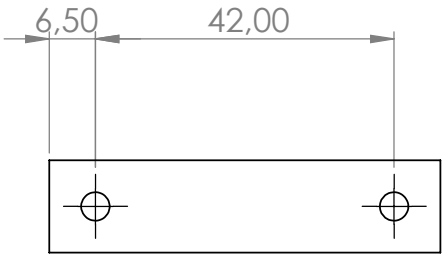
Appendix C

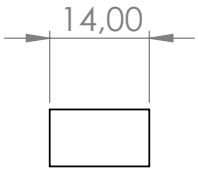
Technical Drawings

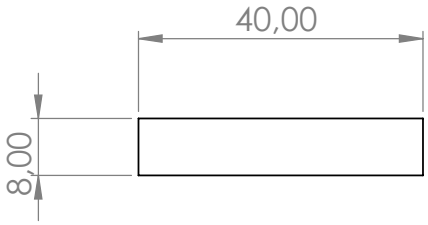


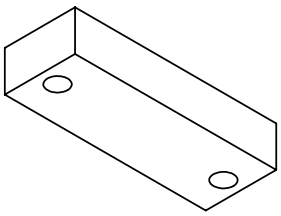


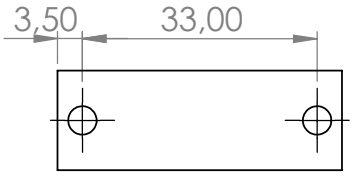












UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN MILLIMETERS SURFACE FINISH: TOLERANCES: LINEAR: ANGULAR:	FINISH:	DEBURR AND BREAK SHARP EDGES	DO NOT SCALE DRAWING / REVISION

	NAME	SIGNATURE	DATE
DRAWN			
CHK'D			
APPV'D			
MFG			
Q.A			

TITLE:

Height adjusters

MATERIAL:

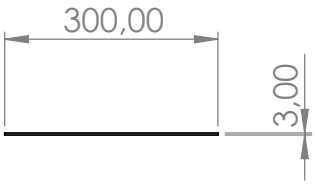
DWG NO. Base_Rodamento
Base_agarre

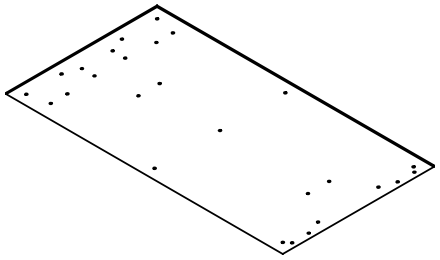
A4

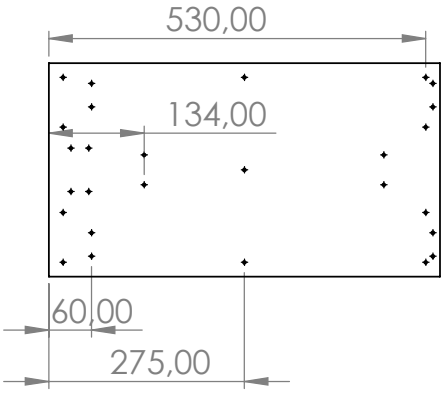
WEIGHT:

SCALE:1:1

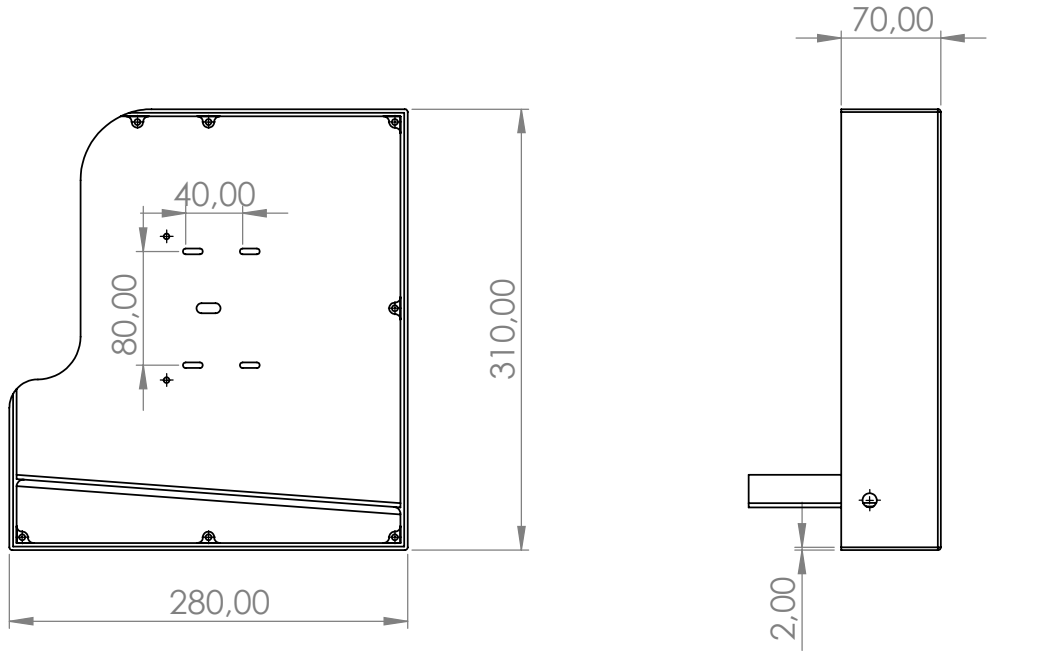
SHEET 1 OF 1

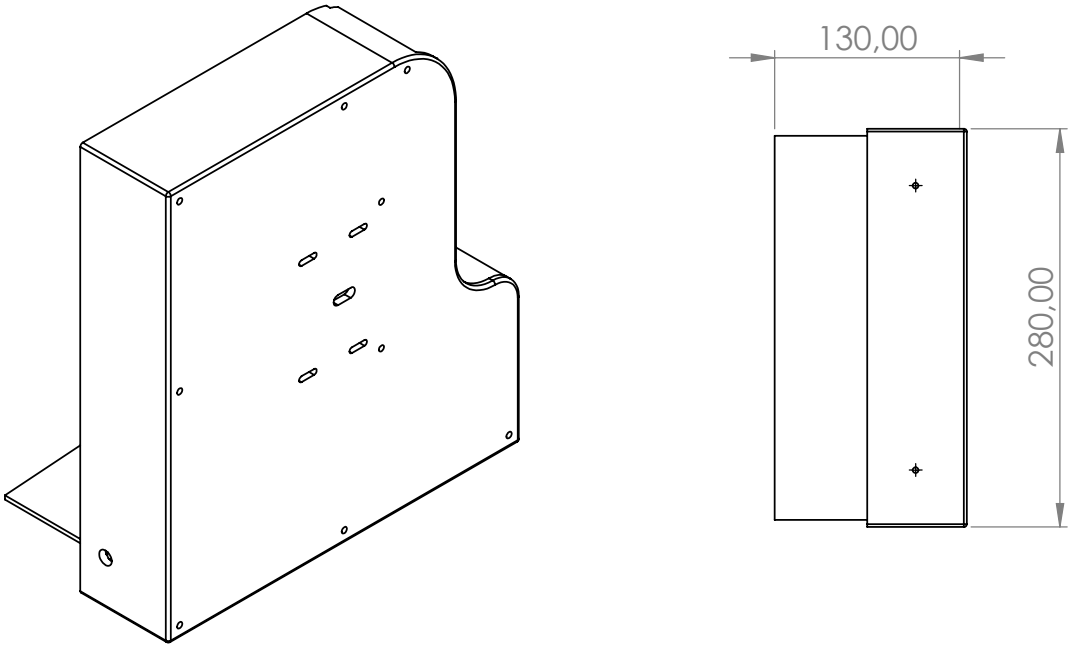






UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN MILLIMETERS SURFACE FINISH: TOLERANCES: LINEAR: ANGULAR:		FINISH:		DEBURR AND BREAK SHARP EDGES		DO NOT SCALE DRAWING	REVISION
	NAME	SIGNATURE	DATE			TITLE:	
DRAWN						Platform	
CHK'D							
APPV'D							
MFG							
Q.A				MATERIAL:		DWG NO.	A4
						Base_T	
				WEIGHT:		SCALE:1:10	SHEET 1 OF 1





UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN MILLIMETERS SURFACE FINISH: TOLERANCES: LINEAR: ANGULAR:	FINISH:	DEBURR AND BREAK SHARP EDGES	DO NOT SCALE DRAWING / REVISION

	NAME	SIGNATURE	DATE
DRAWN			
CHK'D			
APPV'D			
MFG			
Q.A			

TITLE: Box_1

MATERIAL:

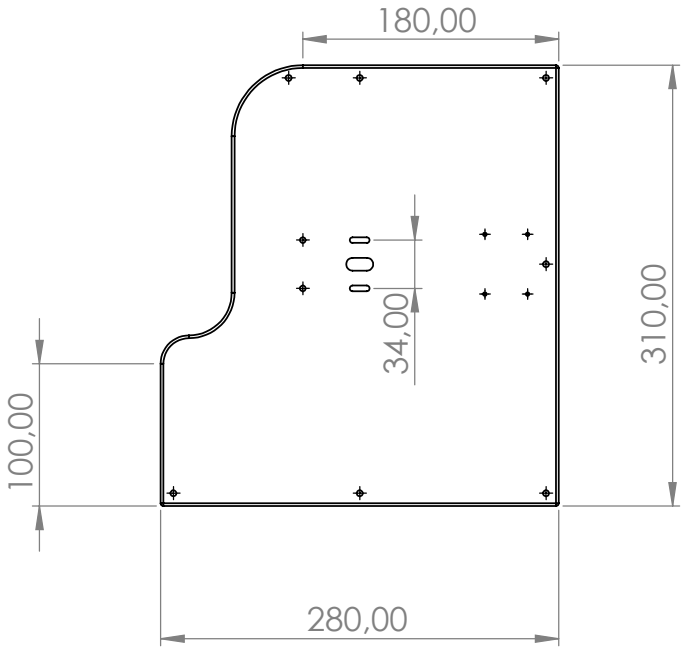
DWG NO. Caja_1_v2

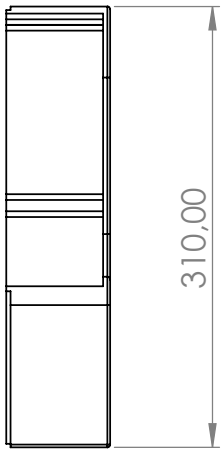
A4

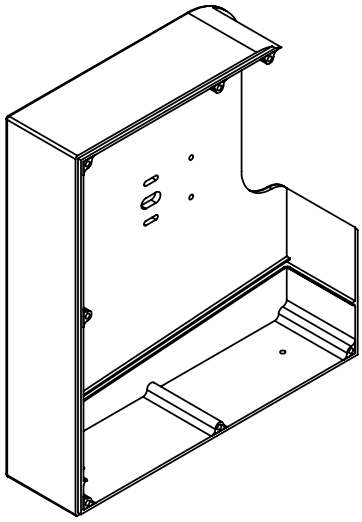
WEIGHT:

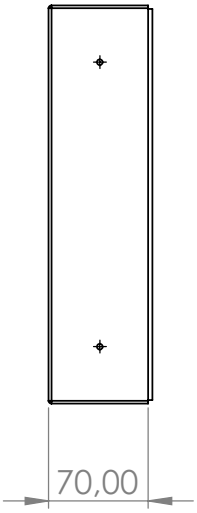
SCALE:1:5

SHEET 1 OF 1









UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN MILLIMETERS SURFACE FINISH: TOLERANCES: LINEAR: ANGULAR:	FINISH:	DEBURR AND BREAK SHARP EDGES

	DO NOT SCALE DRAWING	REVISION

	NAME	SIGNATURE	DATE
DRAWN			
CHK'D			
APPV'D			
MFG			
Q.A			

TITLE:

Box_2

MATERIAL:

DWG NO.

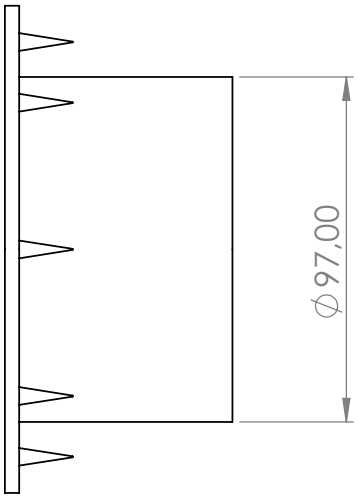
Caja_2_v2

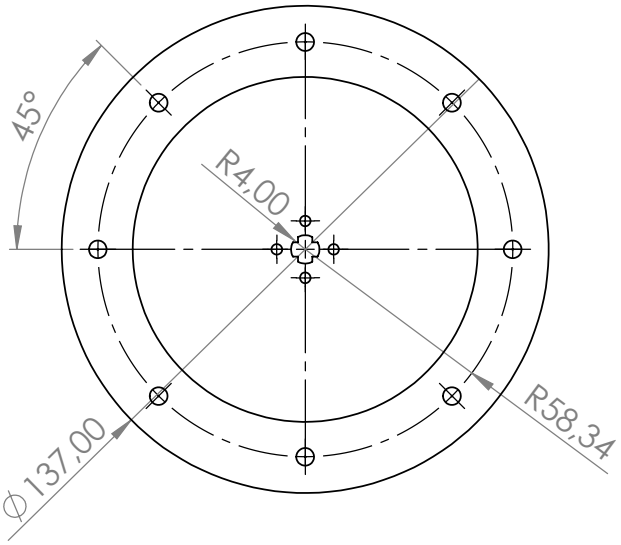
A4

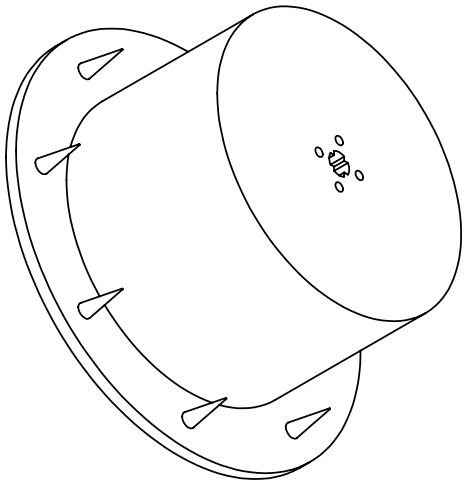
WEIGHT:

SCALE:1:5

SHEET 1 OF 1







UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN MILLIMETERS SURFACE FINISH: TOLERANCES: LINEAR: ANGULAR:	FINISH:	DEBURR AND BREAK SHARP EDGES	DO NOT SCALE DRAWING / REVISION

	NAME	SIGNATURE	DATE
DRAWN			
CHK'D			
APPV'D			
MFG			
Q.A			

TITLE:

Center of the sponge

MATERIAL:

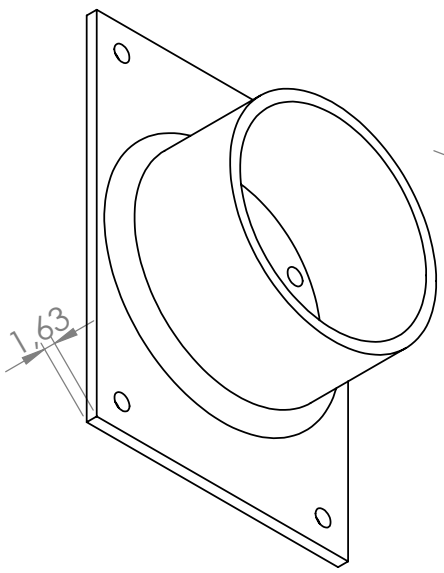
DWG NO. centro_esponja

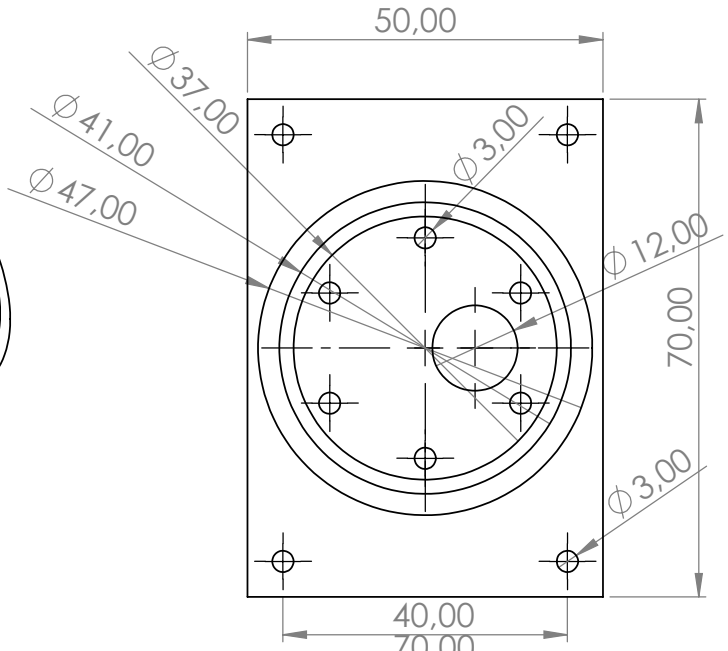
A4

WEIGHT:

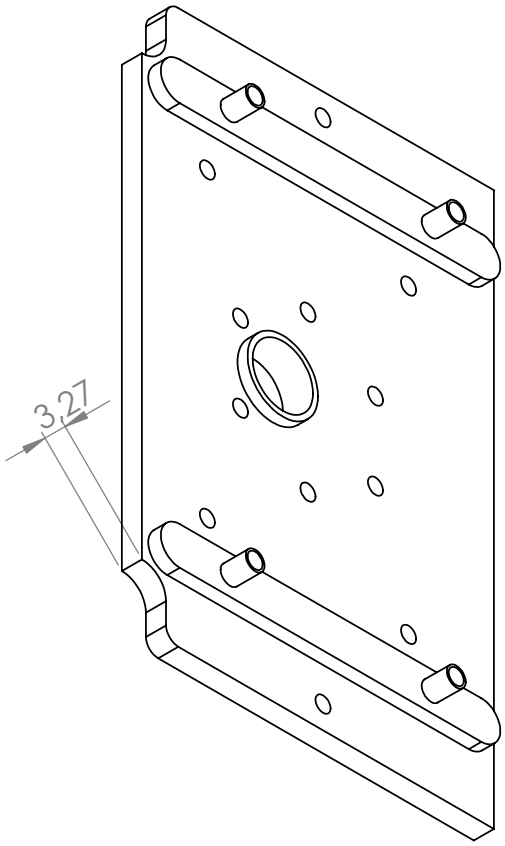
SCALE:1:2

SHEET 1 OF 1

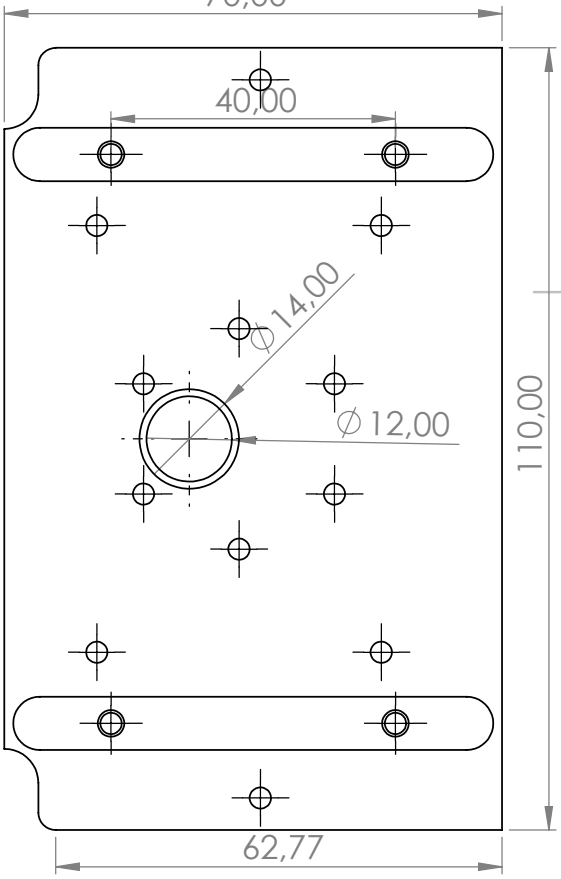




1,63



3,27



UNLESS OTHERWISE SPECIFIED: DIMENSIONS ARE IN MILLIMETERS SURFACE FINISH: TOLERANCES: LINEAR: ANGULAR:	FINISH:	DEBURR AND BREAK SHARP EDGES	DO NOT SCALE DRAWING / REVISION

	NAME	SIGNATURE	DATE
DRAWN			
CHK'D			
APPV'D			
MFG			
Q.A			

TITLE:

DC motor suport

MATERIAL:

DWG NO. DC_MOTOR_SUPORT

A4

WEIGHT:

SCALE:1:1

SHEET 1 OF 1