1 2 9 0

UNIVERSIDADE DE COIMBRA

Francisco Fonseca Fernandes

DESENVOLVIMENTO DE UM SISTEMA DE CONTROLO ATIVO DE FORÇA PARA A INDÚSTRIA

ELETRÓNICA E SOFTWARE DE OPERAÇÃO

VOLUME 1

Dissertação no âmbito do Mestrado Integrado em Engenharia Mecânica, na especialidade de Produção e Projeto, orientada pelo Professor Doutor Joaquim Norberto Cardoso Pires da Silva e apresentada ao Departamento de Engenharia Mecânica, da Faculdade de Ciências e Tecnologias da Universidade de Coimbra

Dezembro de 2020

1 2 9 0
FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE D COIMBRA

Development of an active force control system for the industry – electronics and operating software

Submitted in Partial Fulfilment of the Requirements for the Degree of Master in Mechanical Engineering in the speciality of Production and Project

Desenvolvimento de um sistema de controlo ativo de força para a indústria – Eletrónica e software de operação

Author

Francisco Fonseca Fernandes

Advisor

Professor Doutor Joaquim Norberto Cardoso Pires da Silva

Jury

President	**Professor Doutor Ricardo Branco** **Professor Auxiliar da Universidade de Coimbra**
Vowels	**Professora Doutora Trayana Tankova** **Professora Auxiliar da Universidade de Coimbra** **Professor Doutor Carlos Viegas** **Professor Auxiliar Convidado da Universidade de Coimbra** **Professor Doutor Gil Lopes** **Professor Auxiliar da Universidade do Minho**
Advisor	**Professor Doutor Joaquim Norberto Pires** **Professor Associado c/ Agregação da Universidade de Coimbra**

Coimbra, December, 2020

“Don’t let what you cannot do interfere with what you can do.”

John Wooden

ACKNOWLEDGEMENTS

Before anything else, I have to mention the people who have supported me in every possible way in the making of this dissertation.

Among them, my parents and Carolina, who have always been the most supportive.

My friends, that were always there for me when I needed advice and to relax.

My project colleagues, João and Nuno, for being with me since the beginning of my college life.

My colleagues, João and Carlos, who have been of great help throughout the development of this work.

Last but not least, I would also like to thank my thesis advisor, Joaquim Norberto Pires, for enlightening me with his knowledge during the making of this thesis and for making this project possible.

Abstract

Robots are nothing more than a tool to mimic and augment human being intrinsic capabilities. As the industry developed, research started being conducted to not only concede movement to robots but also tactile sensations. Force control mechanisms are how those sensing capabilities are achieved. Allied with automation, this has been creating more efficient solutions in the industrial panorama.

Aware of this and of the clout from the new industrial revolution, industry 4.0, a group of senior college students from the Integrated Master's in Mechanical Engineering (MIEM) set to develop a solution of a remote active force control mechanism oriented to the current needs of the industry.

For this, it is necessary to put together a review of concepts and devices associated with automation and digitalization of processes as well as active force control. This thesis particularly focuses on the controlling mechanism, which concerns electronics and the operating software of a microcontroller with its connection to a user platform.

Therefore, how the electric circuit was built and what tensions it is supposed to carry is explored in detail. Then proceeds a description of how the *Arduino* was programmed to control the pneumatic components used and the logic behind that programming. It was also important to make this controlling mechanism user-friendly and for that to happen, was created a graphical user interface that provides real-time data from the system.

On what concerns these aspects, the work developed was a success since the connection between the devices described throughout this thesis was achieved and an electric circuit was successfully assembled to manage the electric signals in the system, settling a base for a new active force control system ready to be used and further developed.

Keywords Robots, Industry 4.0, Active force control, Pneumatics, Microcontroller, Connection.

Resumo

Os robots não são mais do que uma ferramenta para imitar e aumentar as capacidades intrínsecas ao ser humano. Com o desenvolvimento da indústria, a investigação começou a ser conduzida não só para dar movimento aos robots, mas também para lhes conferir sensações táteis. Os mecanismos de controlo de força são a maneira de alcançar estas capacidades de sentir. Isto, aliado à automação, tem vindo a criar soluções mais eficientes no panorama industrial.

Cientes disto e das influências da nova revolução industrial, indústria 4.0, um grupo de estudantes do último ano da faculdade do Mestrado Integrado em Engenharia Mecânica (MIEM) procurou desenvolver uma solução de controlo ativo de força remoto orientado para as necessidades atuais da indústria.

Para tal, é necessário fazer uma revisão de conceitos e tecnologias associados à automação e à digitalização de processos bem como ao controlo ativo de força. Esta tese foca-se particularmente no mecanismo de controlo, que inclui a parte de eletrónica e software de operação de um microcontrolador e a sua conexão a uma plataforma do utilizador.

É explorado em detalhe como é que o circuito elétrico foi construído e que tensões são supostas atravessar o mesmo. Segue-se a descrição da lógica com que o *Arduino* foi programado para controlar os componentes pneumáticos utilizados. Para que este mecanismo de controlo fosse tornado intuitivo, o que também era importante, foi criada uma interface gráfica do utilizador que fornece informação em tempo real do sistema.

No que diz respeito a estes aspetos, o trabalho desenvolvido foi um sucesso, uma vez que a conexão entre os dispositivos descritos ao longo desta tese foi alcançada e o circuito elétrico foi montado com êxito de forma a gerir os sinais elétricos no sistema, construindo uma base para um novo mecanismo de controlo ativo de força pronto a ser usado e a ser posteriormente desenvolvido.

Palavras-chave: Robots, Indústria 4.0, Controlo ativo de força, Pneumática, Microcontrolador, Conexão.

Contents

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS/ ABBREVIATIONS

Acronyms/Abbreviations

CPS – Cyber-Physical System

IoT – Internet of Things

IIoT – Industrial Internet of Things

ICT – Information and Communications Technologies

AI – Artificial Intelligence

IFR – International Federation of Robotics

DP – Digital Pneumatics

GUI – Graphical User Interface

Wi-Fi – Wireless-Fidelity

AFC – Active Force Control

PID - Proportional-Integral-Derivative

TCP/IP – Transmission Control Protocol / Internet Protocol

ACF – Active Contact Flange

I/O – Input/ Output

IDE – Integrated Development Environment

CAD – Computer-Aided Design

USB – Universal Serial Bus

PWM – Pulse-With Modulation

EEPROM – Electrically Erasable Programable Read-Only Memory

DAC – Digital to Analog Converter

OOP – Object-Oriented Programming

1. INTRODUCTION

The 4.0 Industry, known as the fourth industrial revolution of those represented in Figure 1.1, promises to change realities in the industry in a near future. According to (Xu et al., 2018), it was officially announced in 2013 as a German strategic initiative but it had already been introduced in the Hannover Fair back in 2011. It builds upon the improvements that came to the manufacturing sector with the 3rd Industrial Revolution, of which the introduction of computers and automation, and integrates them with network connectivity and smart systems that include devices such as sensors and actuators. This new revolution is mainly focused on the further automatization of processes and aims to provide levels of autonomy in factories that were not possible until recent times.

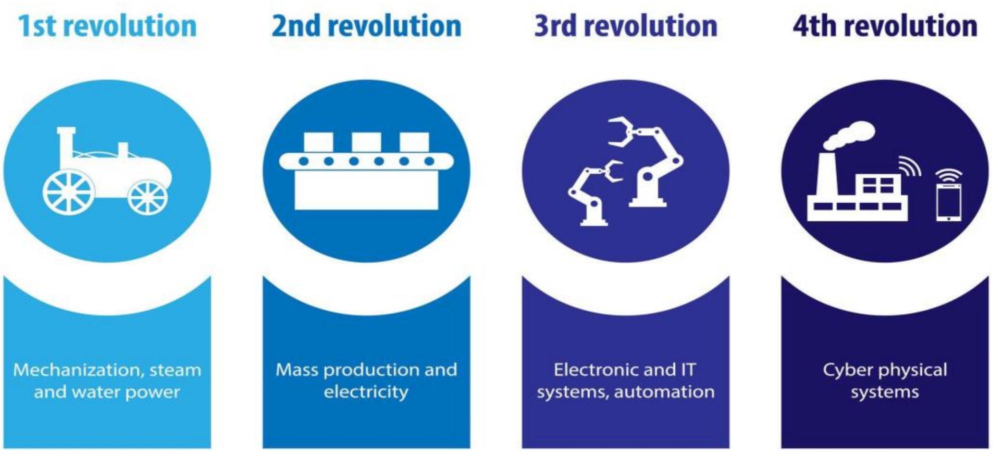

Figure 1.1. Industrial Revolutions in evolution. Source: https://www.trace-software.com/blog/what-is-industry-4-0/

On a more comprehensive level, (Hermann et al., 2015), based on an extensive literature review, concluded that industry 4.0 could be dismantled in four main concepts that are intertwined with each other. These, they claim to be *cyber-physical systems* (CPS), *the internet of things* (IoT), *the smart factory,* and *the internet of services* (IoS). These concepts derive into 6 design principles: decentralization, virtualization, real-time capability, interoperability, modularity, and service orientation that express the principles this new industrial revolution is based on.

Proceeding to the definition of CPS, like (Rojko, 2017) described, these are physical systems combined with *information and communications technology* (ICT) components. This allows for the physical system to have a constant information flow via the IoT that enables the above-mentioned design principles to be achieved. CPS have decentralized control and through communication between embedded systems in the same network allow for autonomous decisions to be made by algorithms, leaving only more complicated issues to be communicated to the process/task manager.

Moreover, as (Thames & Schaefer, 2016) refer to, one of the advantages of industry 4.0 is that it allows achieving higher levels when it comes to operational efficiencies. IoT, which in the industrial environment is designated *Industrial Internet of Things* (IIoT), achieves that with the "convergence of industrial systems with advanced computing, sensors, and ubiquitous communication systems", according to the authors. Along with this, the authors mention new rising paradigms such as *cloud-based manufacturing* and *social product development* that aim to boost those operational efficiencies referred and reduce the complexity of product development (R&D), respectively. The figure bellow illustrates most of these paradigms.

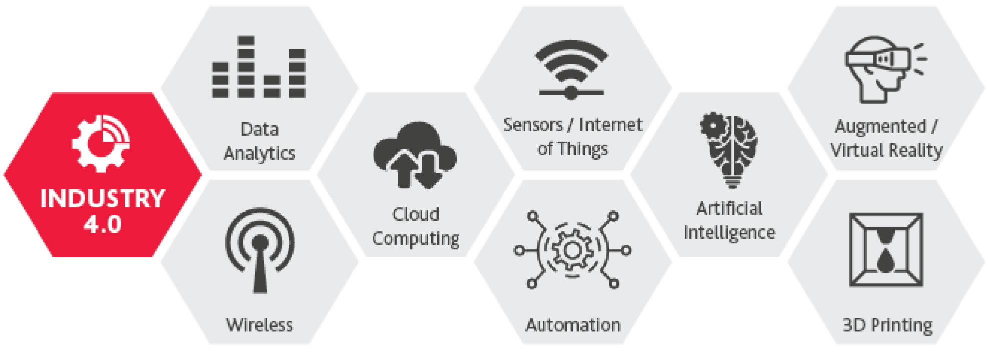

Figure 1.2. Industry 4.0. Source: https://www.designnews.com/automation-motion-control/only-5-mid-size-manufacturers-are-implementing-industry-40

1.1. Motivation

In a Covid-19 pandemic scenario, (Agrawal et al., 2020) research has shown that the outbreak has caused some companies to become resilient on investing in Industry 4.0 while others have accelerated their inclusion processes, in particular for cases where business continuity is at stake, such as digital remote work, automated planning, digital performance management, and automation. Automation and remote execution of processes not only bring more efficacy to these processes but also permits the reduction in human-to-human interaction, and contributes as much as possible to the continuity of the normal flow of the works in a scenario like the present one.

IIoT plays a big role in the field of automation through artificial intelligence (AI). Without injury risk or the need to rest, it is advantageous for manufacturers that some repetitive tasks developed by humans to be, instead, done by systems connected to the IIoT (CPS), (Ciraldo, 2020). The author also explains that despite the obvious social and ethical implications this will have, humans, will still have fundamental roles to play, e.g., remotely control equipment functions, analyse data given by IIoT communications and making data-based decisions, or monitor the process in real-time. The author also notes that theses perks are adding a highly increasing worth to the industries.

As mentioned above, one of the main objectives of Industry 4.0 and IoT is the optimization of operational efficiencies and, thus, the reduction of defects through the automation of processes. The work developed in this thesis is intended to be a practical form where these concepts can be recognised.

For this automation to happen, industrial robots are one of the fundamental tools used. One example is the ABB robot depicted in Figure 1.3. Based on (Bahrin et al., 2016), industrial robots have been being developed at a paced speed since their introduction to the industry in the 3rd Industrial Revolution and are achieving higher standards of precision, customization, autonomy and flexibility. Therefore, robots are becoming more useful each day to improve the quality of execution of ordinary tasks that, when done by humans, are prone to errors or can expose them to dangerous situations.

Figure 1.3. Industrial Robot. Source: https://new.abb.com/products/robotics/industrial-robots/irb-1410

The (International Federation of Robotics, 2020) states that global robot installations have decreased by 12% in 2019 due to trade tensions and a "global economic downturn" after 6 years of big growth. The values of 2020 industrial robot sales will be determined by the global economic crisis related to the current pandemic situation and are, therefore, not expected to be better as well. However, the IFR claims that although a contraction is expected in the short term, the cause of that contraction will also be a booster for digitalization, in the near future, and will keep the future perspectives for growth in the worldwide robotics industry high.

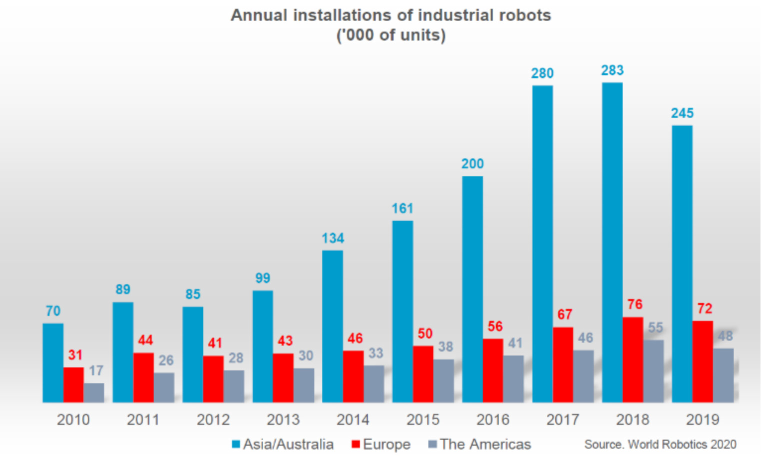

Figure 1.4. Annual installations of industrial robots (International Federation of Robotics, 2020).

One such task of industrial robots is maintaining a constant trajectory during the application of force on a defined surface. This can prove to be difficult to perform by a human worker due to all types of factors like surface irregularities, environmental conditions and the own physical limitations of the human being. In the project that concerns this thesis, an industrial robot will ensure that all the desired trajectories, simulated beforehand in a virtual environment, are covered exactly how they are supposed to.

Moving to another subject - in the industry, when the goal is to achieve a constant force application, the logical solution is to resort to a fluid powered system. Inside this field, there are two options regarding the type of fluid utilized: pneumatics (gas) and hydraulics (liquid). Hydraulics are destined to applications involving heavier loads, so pneumatics is the one to consider for this system. One of the key advantages of pneumatics is that it is much less expensive on the short-term. Maintenance or leaks, although still important, are also not as big of a problem in comparison since normally the fluid being used is air.

Remote controlling systems are preferable in situations where pneumatic components are not of easy access. Not only is this because of the better functionality in specific cases, but also because it allows the user, as (Reljic et al., 2019) explain, to change parameters and configure them or the control system according to the current requirements. This remote configuration and data collection lead to more efficient, flexible and faster processes that end up providing better quality products with decreased costs.

Figure 1.5 illustrates the concept of digital pneumatics (DP) that, according to (Reljic et al., 2019), originated from the ability of pneumatic components to achieve a connection to computers and electronic components. As for the system represented itself, it will be discussed in the next chapter.

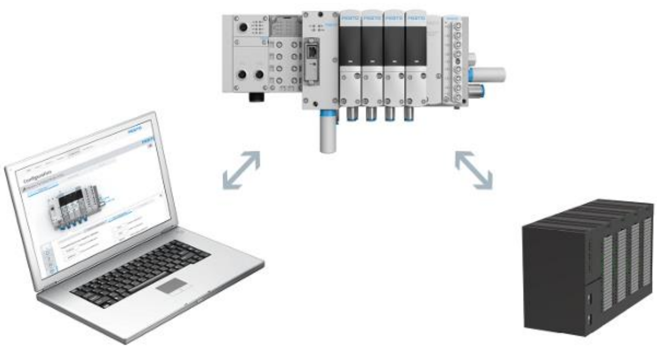

Figure 1.5. Digital pneumatics by Festo. Source: https://www.festo.com/vtem/nl/cms/10169.htm

On the work being developed in this, (Carvalho, 2020), and (Gomes, 2020) dissertations, the merging between pneumatics, microcontrollers, 3D manufacturing and industrial robots, will result in a complete system remote-controlled of active force control.

1.2. Objectives

When this project started, a common goal was set to develop an end effector system that would then be attached to a robot and that could provide constant values of strength output even when exposed to a reaction force. This would need to rely on feedback given by a pneumatic structure.

The main objective of this thesis is to develop the control mechanism of this referred structure through the connection of several devices. To build this whole process of control there are a few objectives needed to be set, and those are:

- Build an electric circuit to manage the tensions coming in and out of the microcontroller into the electrovalve and the pressure regulator;
- Program the microcontroller to give commands and receiving feedback from the pneumatic components;
- Build a graphical user interface (GUI) that can provide easy communication through Wi-Fi between the user and the microcontroller.

Along with these, the implied objective is to develop credible work while accomplishing these objectives to establish the bases of a new system. It is advisory to read my colleague's (Carvalho, 2020) and (Gomes, 2020) thesis for achieving full comprehension of the different parts of the system.

1.3. Chapter organization

This dissertation is organized in 7 chapters aimed for the comprehension of the reader about the work developed and the biases that lead to it. For better achieving this purpose, in chapter 5 a chronological order is followed as much as possible except for parts of the work where that was not possible because of the late arrival of some important materials. For time reasoning, the rest of the system was then prepared earlier to the possible extent as well as to receive the lacking hardware.

The reasoning in each chapter is:

- Chapter 1 – Presents the moment lived in the industry and its concepts and addresses the motivation to develop this dissertation's system;
- Chapter 2 – Expresses the state of the art following the concept of force control and systems related to automation;
- Chapter 3 – Presents the components that would be used in this dissertation;
- Chapter 4 – The software used to achieve control over the components in chapter 3;
- Chapter 5 – Resumes the experimental work developed in this dissertation, almost chronologically;
- Chapter 6 – Presents the tests and the results achieved;
- Chapter 7 – Discusses the conclusion and future work prospects.

2. STATE OF THE ART

This chapter will consider bibliographic research on the concepts and technologies transversal to the ones being present in this dissertation. Special emphasis shall be given to subjects directly related to this thesis, like force control, but this chapter will also include some of the cross-technologies, used as well in my colleague's (Carvalho, 2020) and (Gomes, 2020) dissertations, to get "the full spectrum" of the work developed. The goal is to give the reader a better insight into the subjects and technologies referred and, therefore, making it easier to comprehend in-depth the context of this dissertation.

2.1. Force Control

Force control associated with robotics has been approached in many ways over the last decades as a form of regulating the contact force at the end-effector of a robot.

This thesis, as mentioned in the Objectives, focuses on the constant monitoring of the force when applied to a certain surface, therefore, only makes sense consider the perspective of active force control. Nevertheless, to exploit where the concept "active" comes from follows a brief distinction between active and passive force control.

(Garcia et al., 2007) stated that active force control can be divided into two categories: one that includes schemes which rely on indirect force control such as, stiffness control or impedance control; and another that relies on schemes of direct force control that include explicit force control and hybrid force/position control. The first ones achieve, as stated, force control via motion control and the second ones through explicit closure of a force-feedback loop. Passive force control, as (Garcia et al., 2007) continue, can constitute an interesting alternative as a system called *remote center compliance* comes to prove. However, despite the added simplicity when compared to AFC strategies, the authors refer that the fact that it is the robot's trajectory being changed by interaction forces can make it unable to avoid high contact forces, adding to the decreased flexibility of this system.

The sub-subchapters that follow embed a state of the art on subjects/technologies related to active force control.

2.1.1. Active force control with a macro-mini robotic system

Recently, (Li et al., 2020) proposed a macro-mini robotic polishing system to accomplish active force control for polishing. The macro part refers to a KUKA LBR iiwa 7 R800 robot which provides position control during the process and the mini refers to an end-effector that includes a series of devices and is responsible for constant force control.

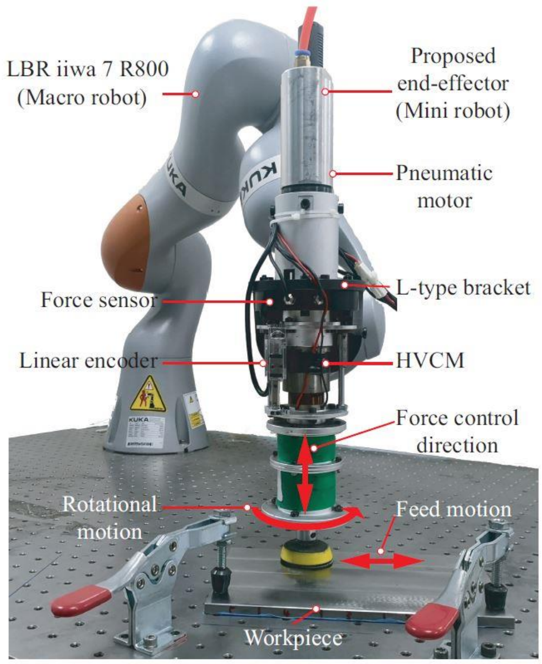

Figure 2.1. Macro-mini robotic system (Li et al., 2020).

The above figure illustrates the full system where, as the authors explain, the force output of 27.9 N was ensured by a force driving motor (HVCM) and the contact force in the normal direction was monitored by a force sensor with a sampling rate of 2.5 kHz that transmitted information back to the control station. The polishing device consisted of a pneumatic motor of 640 W and up to 2100 rpm of rated rotational speed.

To eliminate the force tracking error (F_e) existent between the set value of the desired contact force (F_d) and the actual contact force (F_c, equal to the motor force plus the force of the polishing head), the writers describe the construction of a closed-loop force control mechanism based on the proportional-integral-derivative (PID) control algorithm. This PID method will be better defined in chapter 2.1.2.

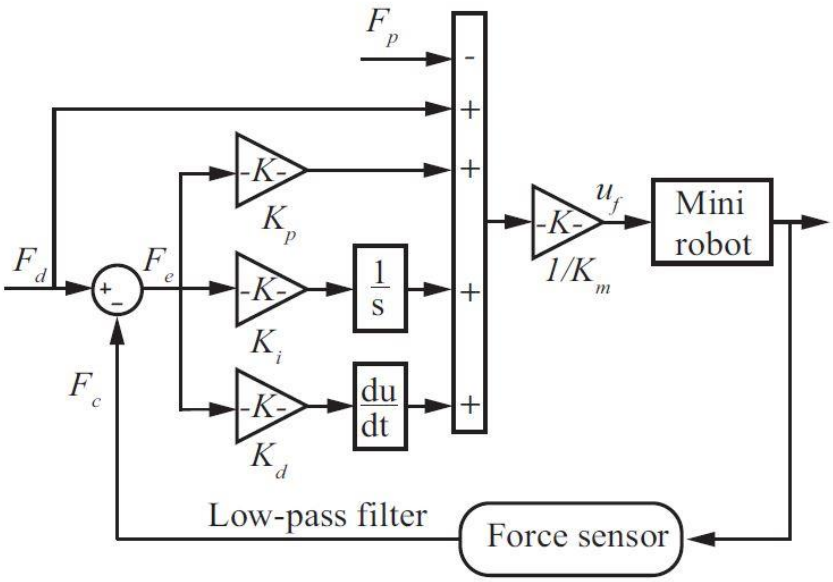

Figure 2.2. End-effector PID loop (Li et al., 2020).

To evaluate the dynamic response performance (DRP), the first tests were conducted by the authors considering the obtained frequency bandwidths of the end-effector, and comparing these with the theoretical ones, as the evaluating method. In this case, the conclusion was that a higher bandwidth can provide better control. (Li et al., 2020) proceeded with further experimental tests including some, where the *KUKA iiwa*, which is known for its very good performance in force control, was used for this capability as a comparison model.

(Li et al., 2020) concluded that a force control strategy based on the conventional PID method exclusively without at least considering a fuzzy adaptive control is not reliable enough in changing conditions. Not being able to adjust the control parameters in real-time for a situation where the experimental conditions are not as controlled as in their experimental tests, like a real polishing process, also constitutes a limitation.

2.1.2. Algorithms in AFC

Two strategic algorithms were just referred to in the context of improvement of the active force control strategy, therefore, proceeds a brief contextualization of both.

Fuzzy logic is one of the strategies that can be integrated into PID loops when looking for improving the accuracy of the output data by eliminating internal or external disturbances that affect the input values. It can be described as a mathematical mean of reasoning inherently vague information, in other words, consists of an evaluation process normally based on *if-then* rules that interpolate using rational values within the 0 and 1 (Boolean logic) range (Mailah & Rahim, 2000; Meon et al., 2012).

The PID control algorithm is essentially a combination of three types of controllers: proportional, integral and derivative, that can be mathematically represented by the following equation:

$$m(t) = k_p * e(t) + k_i * \int_0^t e(t)\,\partial t + k_d * \frac{\partial e(t)}{\partial t}. \quad (2.1)$$

Consists in the modulation of the control signal, m(t), illustrated in the Figure 2.3 below through variation of interdependent parameters associated to each controller (k_p, k_i, k_d, respectively) and constantly updates in a loop like the one illustrated in Figure 2.2. to achieve the desired set point as effectively as possible.

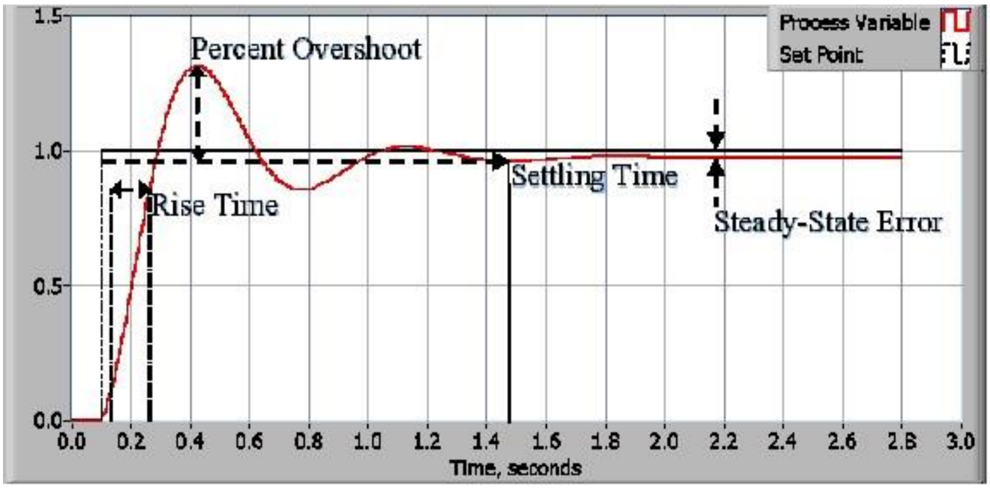

Figure 2.3. PID typical response curve. Source: https://www.ni.com/pt-pt/innovations/white-papers/06/pid-theory-explained.html

The variation of each parameter has a different influence on the curve's overshoot, rise time, settling time and steady-state error. As mentioned in (Pires, 2019), since these parameters are interdependent from each other, there cannot be a precise definition of how each one alone affects the curve characteristics, but generically:

- K_p – Decreases rise time for being a gain proportional to the error e(t);
- K_i – Eliminates steady-state error;
- K_d – Improves system stability and decreases the overshoot and the settling time.

As for the applicability of these algorithms in the work being developed in this thesis, none of them will be taken into consideration on an initial stage. However, they were presented as a state of the art because of their potential for an optimization phase, especially PID control, since it could be suited for the current work. The set value from the pressure regulator, e.g., has chances of not corresponding exactly to the force value it is intended to correspond. In this case, with the use of a force sensor, similar to what was done in 2.1.1, PID control could be used to improve the set value accuracy and else.

2.1.3. Digital Pneumatics

In the Motivation, pneumatics was introduced as a powerful tool for force control because of its suitability to keep constant forces. In chapter 2.1.1 was presented a case where the inability to change control parameters in real-time was a constraint for the optimal performance of the force control system. The merging of pneumatics with remote control seems is, in this light, preferable to having to rely on other devices like the one described previously or on built-in robot force control.

Festo is a multinational company that exports automation technology and technical solutions in that field worldwide. In recent years they have developed an innovative technology to upgrade pneumatic circuits, the Motion Terminal VTEM.

This device is a CPS that puts together valves with embedded pressure, position and temperature sensors with controller for closed-loop control, ethernet interface, electrical inputs (digital/analogue), integrated supply temperature and pressure sensors for data analysis, and more in one single station (Quintero, 2019).

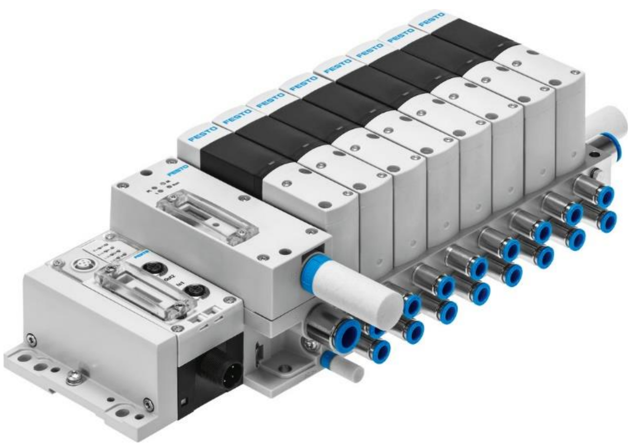

Figure 2.4. Motion Terminal VTEM. Source: https://www.festo.com/hk/en/p/motion-terminal-id_VTEM/

According to (Festo, n.d.), the station offers decentralized intelligence and app-based function implementation that allows adaptations to be done decentrally within the system, reducing the complexity of programming and controlling tasks. They claim that motion terminal VTEM has a bridge circuit with integrated sensors made up by 2/2-way valves with piezo pilot control and diaphragm poppet valves that can assume multiple configurations and allow to pressurise and exhaust independently from each other, permitting the user to do proportional pressure regulation and complex control solutions.

Multi-modal communication is also available and this (Festo, n.d.) explains to be achieved through various user interfaces. These, they say, are of simple and fast adjustment that can be done via different options like Ethernet connection or the WebConfig interface. Furthermore, the compactness of this device and its motion apps result in a much better system, energy-efficiency wise, when compared to standard pneumatic systems, fully integrating this device into Industry 4.0.

2.1.4. Active Contact Flange (ACF) by FerRobotics Inc.

FerRobotics's patented active contact flange is an active force control technology intended to automate tasks that require flexibility and sensing capabilities (FerRobotics Inc., n.d.). This automation targets jobs that required specialized handcraft like polishing, granting the highest quality standards by excluding human physical limitations.

As the standard communication interface, it uses Ethernet TCP/IP and there are multiple models available to suit the application requirements.

Figure 2.5. ACFs. Source: https://www.ferrobotics.com/en/technologie-produkte/produkte/acf/

The device itself has one degree of freedom and combines a pneumatic actuator with sensors and closed loops, illustrated in Figure 2.6. An ACF/110/01 was used in (Mohsin et al., 2019) work to achieve full control over a polishing force. It is described by them as a hybrid position/force control system programmed with force outputs via its Ethernet-based interface while constantly monitoring the actual applied force and position displacement.

Ultimately, it is referred by (Mohsin et al., 2019) that the pneumatic actuator in the ACF quickly compensates any abrupt changes in the contact force and includes gravitation compensation by providing constant force with changing orientations. This is especially important when the device is being used as the end-effector in a robot.

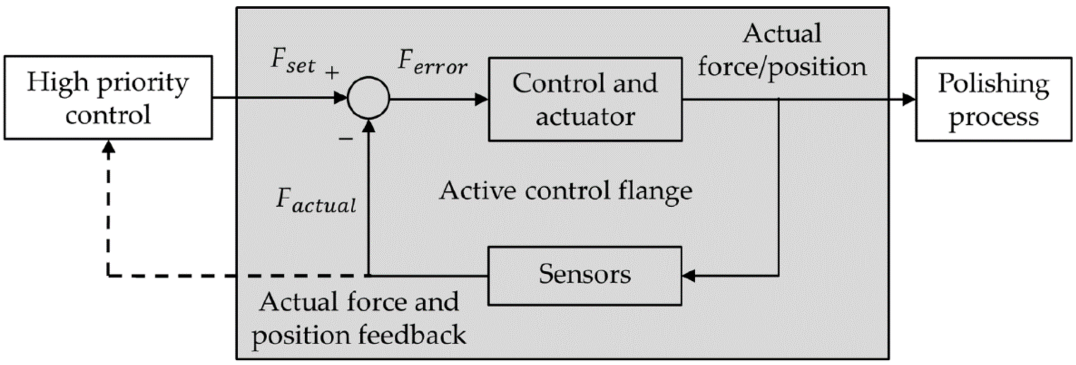

Figure 2.6. Closed-loop of the active contact flange (Mohsin et al., 2019).

2.2. Industrial robots

The first industrial robot, known as the *Unimate #001*, was developed in 1959 by George Devol and Joseph Engelberger. Soon after, in 1961, George Devol would be awarded the US patent No. 2,988,237 for this innovation. Further on, industrial robots would start to spread across the industry.

In the beginning, according to (Garcia et al., 2007), because of the technical necessities and the market influence of the industry, the specifications of the industrial robots met the needs of the automotive industry. Kinematic calibration, motion planning for an accurate movement of the robot and also control laws that would ensure the proper execution of the robot's task were, in the authors understanding, the main research and development topics in the field of robotics on those first 30 years. Further on, in the dawn of the 90s, the authors mention that flexibility started to be imposed as one of the primary characteristics due to an expanding variety on the application areas for industrial robots.

The present industrial robot seeks to adapt to the emerging industry 4.0 and the exponentially rising demand from the most various types of industries. While all the other mentioned required characteristics like flexibility continue to be developed, others like their connection with humans, autonomy and remote-control capabilities have been being developed in hindsight of IIoT advancements.

When selecting an industrial robot, primarily one must consider the application it is meant for, the maximum payload needed, and the reach required. Looking at a practical case: in this dissertation, the solution available was the *ABB IRB 140* which is a very functional option to conduct testing in a limited space like a laboratory, due to its compact size. An identical model to the one used is depicted in the Figure 2.7 below.

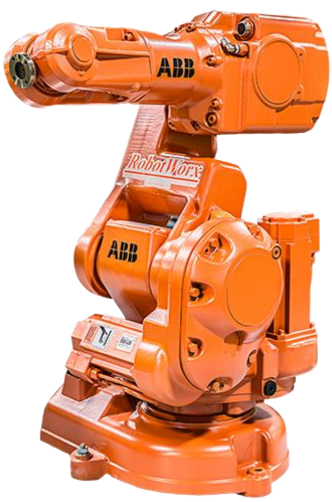

Figure 2.7. Industrial Robot ABB IRB 140. Source: https://www.robots.com/robots/abb-irb-140

Moreover, on the (ABB, n.d.) website it is possible to obtain some other interesting specs on the IRB 140, such as it offers a reach of 810 mm along with its 6 axes and can handle payloads up to 6 kg, the cabling is integrated so that there will not be loose cables where they are not supposed to, and it has a collision detection function for safety and reliability purposes. Also, with a weight of only 98 kg and for the fact that it can be wall-mounted, inverted or floor mount makes it a flexible and multipurpose industrial robot.

Additionally, like most *ABB* industrial robots, this one is equipped with the IRC5 controller and robot control software and allows for network communication which is done using *ABB* specific robot language – *RAPID*.

2.3. Microcontrollers and their role in robotics

A controlling device can prove to be useful in robotics, especially when there is the necessity to automate a process and making a robot capable of taking decisions.

When choosing the appropriate controller, there are many options to choose from and the choice, as always, should be fit to the project needs. If the project involves lots of information processing then probably the best option to consider is a microprocessor like the *Raspberry Pi*, which has more memory, a better processor, but it is not so good for I/O data exchange and consumes significantly more power. On the other hand, if the goal is to connect the controller directly to external hardware and use I/O to send and receive information to and from, e.g., some type of sensor, then a microcontroller like an *Arduino* is the best choice to make, as it will happen to be the case for this project.

Figure 2.8. *Raspberry Pi* (left) and *Arduino* (right). Adapted from source: https://create.arduino.cc/projecthub/rjrajbir/raspberry-pi-vs-arduino-05f69b

Arduino was created in 2005 at the Interaction Design Institute Ivrea in Italy as an open-source hardware and software educational platform. It was designed to be inexpensive, include an *Integrated Development Environment* (IDE), to be programmable via USB and promote knowledge trade between its community (Gibb, 2010).

Some *Arduino* boards, like the MKR family, focus mostly on IoT. Associated with the low power consumption of an *Arduino*, this can be very useful for remote control applications and eliminates some of the wiring complexity electric circuits sometimes have. The MKR *Arduino* family offers options to project developers that range from Wi-Fi, allowing for the connection of devices embedded in the same network, to GSM (*global system for mobile communications*) for worldwide communications (Pecina, 2018).

3. COMPONENTS

This project contains 2 different types of components. The ones that will be recognised, in this thesis, as the first type of components are the ones that are related with the functioning of the system and were bought from third party companies. As for the second type, it includes support components that make is possible for the components to all work together as one bigger system and are developed and described in my colleague João's thesis (Carvalho, 2020). These were designed in CAD software and further printed on a 3D printer.

This thesis will only cover the first type of components which are directly related to the controlling mechanism of the system.

3.1. Electrical Components

This subchapter presents the electrical hardware used. Initially, most of this hardware was installed in two breadboards for prototyping experiments.

3.1.1. Tension Regulation

To ensure that the required tensions were available in the circuit, there was the need to create 2 groups for tension regulation. The left one ensures the 24V from the source are reduced to 12V and the right one reduced to 5V. These groups were made with the following components:

- 1x 0.33 µF capacitor
- 1x L7812 tension regulator
- 1x 0.1 µF capacitor

- 1x 0.33 µF capacitor
- 1x L7805 tension regulator
- 1x 0.1 µF capacitor

An operational amplifier OP07CP was also installed to increase the output tension from the *Arduino* to the pressure regulator.

Initially, a 5 V relay was also installed to activate and deactivate the electrovalve.

3.1.2. Arduino MKR1000

The *Arduino* board chosen for this system was the *Arduino MKR1000*. It is programmed in the *Arduino* IDE like every other *Arduino* board but operates on a tension of 3.3 V unlike other models as the *Arduino Uno* which operate on 5 V.

The reason to select this board over another like the Uno was its ability for network connection. This way, once the board is fixated into the device structure and the program has been uploaded through the micro-USB cable connected to the computer, commands can be sent remotely via TCP/IP connection. The programs stay stored in flash memory, meaning that they do not need to be re-uploaded each time the board is powered.

Apart from this key feature, the board illustrated in Figure 3.1 has multiple digital pins for digital input/output connections, including PWM pins, also 6 analogue pins are available for analogic inputs, one of which (A0) can also function as an analogic output via a 10-bit digital to analogue converter (DAC). This last feature allows for this microcontroller to send continuous signals in a 0 – 1023 (2^{10}) range in applications where square waved signals (digital) are not adequate.

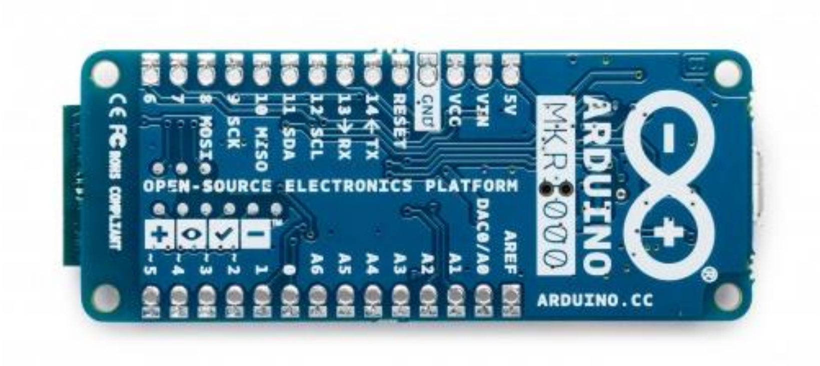

Figure 3.1. *Arduino* MKR1000. Source: https://store.arduino.cc/arduino-mkr1000-wifi

Finally, a last feature on this board, that was particularly important for this dissertation are the Vin and 5 V pins it has. The first can be used as a 5 V power input for the board, replacing the micro-USB port as the source of power, and the last can be used as a 5 V power output if the micro-USB cable is plugged in. Both these functionalities had their importance in different phases of the work and will be discussed in the chapter dedicated to the procedure.

3.1.3. Arduino MKR Relay Proto Shield

Shields are boards that are mainly used to extend the features offered and *Arduino* board. There can be many different types of shields attending to what upgrades the *Arduino* used requires.

This one is depicted in Figure 3.2 and features a couple of integrated relays, one of which will be used to replace the 5 V relay used in prototyping. These relays are activated instead by 3.3 V set on and off by pin 1 and 2 which correspond to relay 1 and 2, respectively. The shield also has 3 positions screw terminal block for each relay (NO, COM, NC) and another 8 for easy connections with pins of the MKR board like SCL and SDA for the I2C bus and ground. NO stands for "normally open", NC for "normally closed" and COM is the port where the signal conditioned by the relay, arrives the same.

At last, the shield has a proto area to assembly other components like the ones described in 3.1.1.

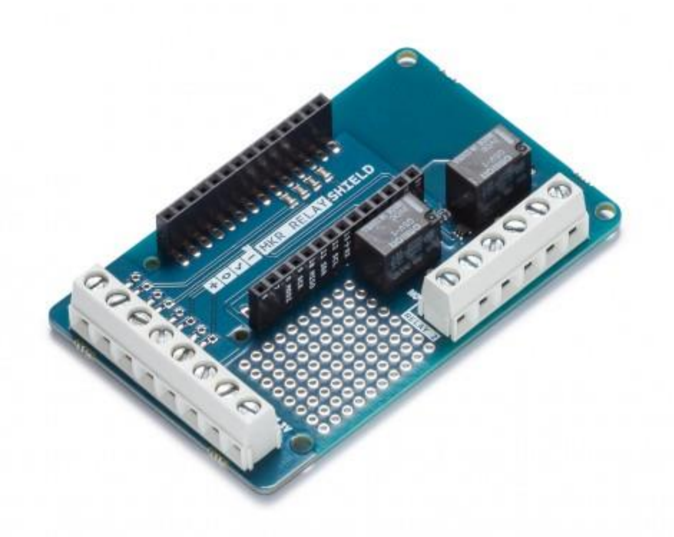

Figure 3.2. *Arduino* MKR Relay Proto Shield. Source: https://store.arduino.cc/arduino-mkr-relay-proto-shield

3.1.4. Sparkfun I²C DAC Breakout - MCP4725

This breakout board allows for converting a digital signal into analogue with a 12-bit resolution DAC. It also has EEPROM for settings storage and 6 pins of which a ground and an OUT pin for connecting to an oscilloscope or other devices, SCL, SDA, another ground and a VCC pin, through which the board is powered.

The OUT pin is where the analogic signal leaves the board, therefore, the one to connect to the operational amplifier which would then carry the amplified signal to the pressure regulator being present next.

The board and all the pins described are visible in Figure 3.3.

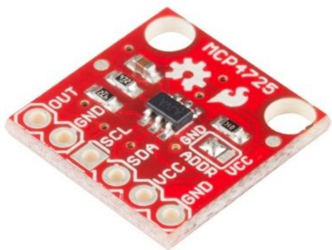

Figure 3.3. Sparkfun MCP4725. Source: https://www.sparkfun.com/products/12918

3.2. Pneumatic Components

This subchapter is dedicated to the pneumatic components used in the project. The presentation order follows the order in which they are used from since an electrical signal is sent from the *Arduino* until it translates into force applied at end-effector. All the pneumatic components, as it will be noticeable, are from SMC Corporation, a company specialized in pneumatic control.

3.2.1. Pressure Regulator

The SMC ITV0050-3BL, presented in Figure 3.4, was the electro-pneumatic regulator used. Once the compressed air reaches it, it allows controlling pressures with set values between 0.001 MPa and 0.9 MPa in linear relation to an electric signal ranging from 0 V to 10 V. It has four electrical connections destined to ground, power supply, input signal and output signal.

Figure 3.4. Pressure Regulator. Source: https://octopart.com/itv0050-3bl-smc-9858218

3.2.2. Electrovalve

The model of the SMC solenoid valve used is SY5120-5DZ-01F-Q and is depicted in Figure 3.5. It is a 5/2 ports solenoid valve and has a maximum operating pressure of 0.7 MPa. It is used to control the flow in and out of the pneumatic cylinder and requires a tension of 24 V to switch position. Once the tension is supplied, an indicating red led turns on and the valve advances position. If the tension stops being supplied, the led turns off and the valve returns to its standard position.

Figure 3.5. Electrovalve. Source: https://uk.rs-online.com/web/p/pneumatic-solenoid-pilot-operated-control-valves/6862668/

3.2.3. Pneumatic Cylinder

With a maximum operating pressure of 1 MPa, the SMC MGP L40 TF-25AZ was the double-action pneumatic cylinder, with a bore size of 40 mm, used. Its course has a length of 25 mm with cushioning. A pneumatic cylinder from the same series is presented in Figure 3.6.

Figure 3.6. Pneumatic cylinder. Source: https://www.smc.eu/en-eu/products/mgp~88411~nav?productId=126586&partNumber=MGPM25TF-25AZ

4. SOFTWARE

In this chapter, it will be presented shortly the software used in the work this thesis refers to emphatically. How this software was used throughout the project shall be discussed in the next chapter.

4.1. Arduino IDE

The *Arduino* Integrated Development Environment (IDE) is a cross-platform application, which means it can be run in various operating systems like Windows, Linux or macOS and which is written and works with the fundamentals of the C/C++ programming languages. It allows the user to upload the sketch to the board, has a wide range of libraries, each with its own functions, available to fit the project needs and it is also open source. It relies on two primary functions called void setup and void loop, of which the first one runs once when the program is uploaded to the board and the second keeps running on a loop. Furthermore, it also includes a serial monitor and a serial plotter to keep control of the different signals the board is sending or receiving.

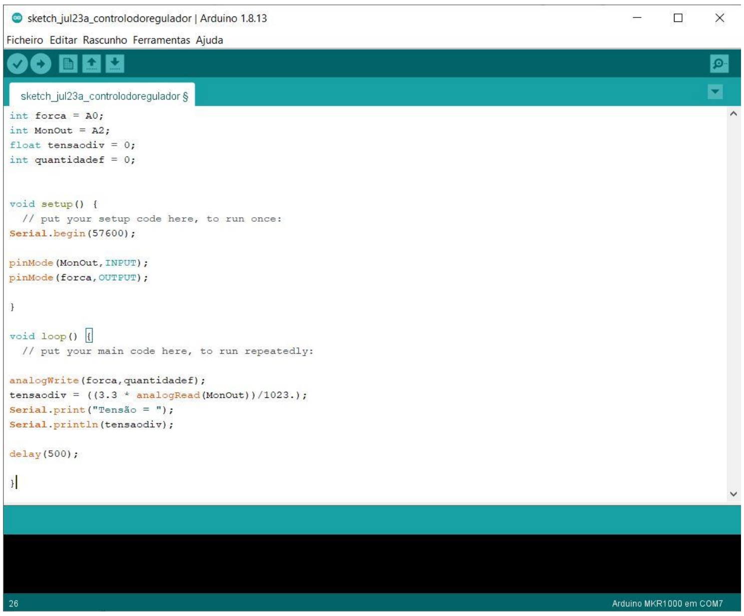

Figure 4.1. *Arduino* IDE with the two primary functions referred and a short example of code.

4.2. Microsoft Visual Studio 2019

To make the bridge between the user and the control of the equipment emerged the need to create a GUI. To create this interface, another integrated development environment was used. *Microsoft Visual Studio* is a software developed by Microsoft and it supports a variety of programming languages, including C# that was used to programme the different elements of the interface. C#, on the other hand, is an object-oriented programming (OOP) language which in practical terms means that it provides a clearer structure for the programs with fewer code lines than non-OOP languages as well as shorter development periods.

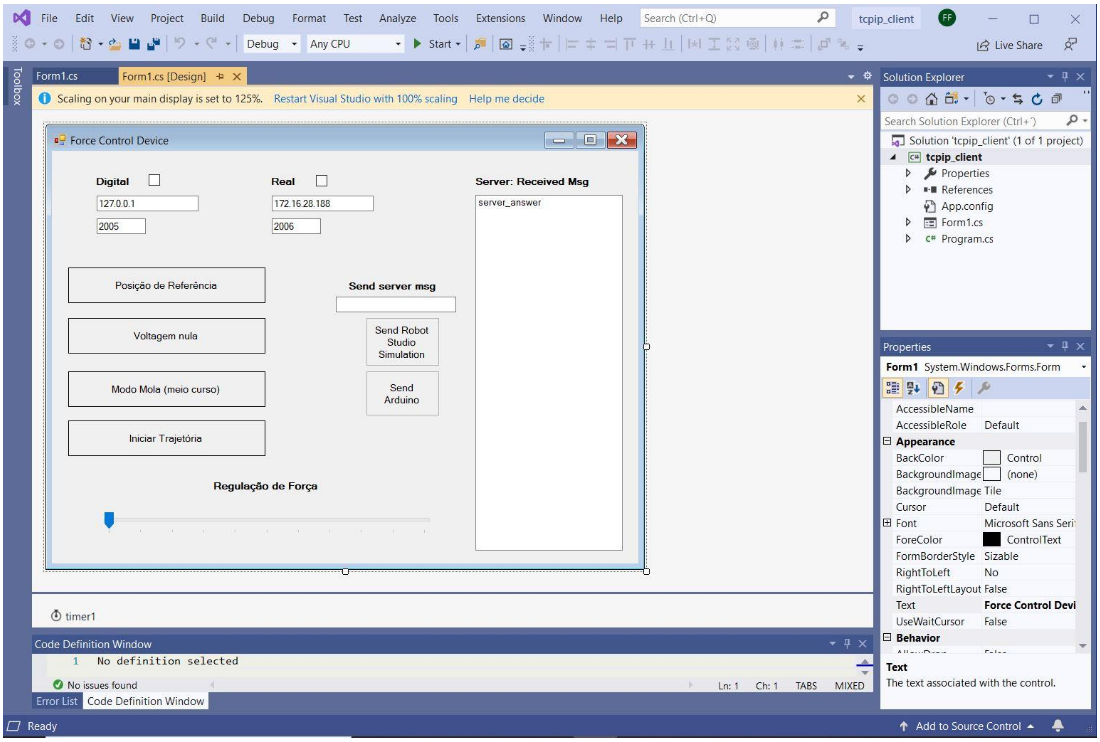

Figure 4.2. Example of GUI developed in Microsoft Visual Studio 2019.

5. CONTROL AND COMMUNICATION

In this chapter, the aim is to describe how the different tensions travel along the circuit into the control board (*Arduino MKR1000*) and how they work and are communicated as signals for the different parts of the circuit. This thesis is aimed for the electronics and control software part of the project, which is done through the hardware presented in chapter 3 and the software presented in chapter 4, therefore, this chapter focuses mostly on how those aspects were developed experimentally.

In the first subchapter, all the small steps were considered from the beginning phase of the project, since a missing detail could represent a constraint in the understanding of the work developed to the full extent.

5.1. Tension Management Circuitry

To begin, a source was set to give the system a tension of 24 V as well as a common ground. This source of tension would furtherly be replaced by the robot's once the end-effector was attached it. For the first tests, a prototype circuit was built using two breadboards. The electric signals were constantly monitored with the use of an oscilloscope and a multimeter as seen in Figure 5.1.

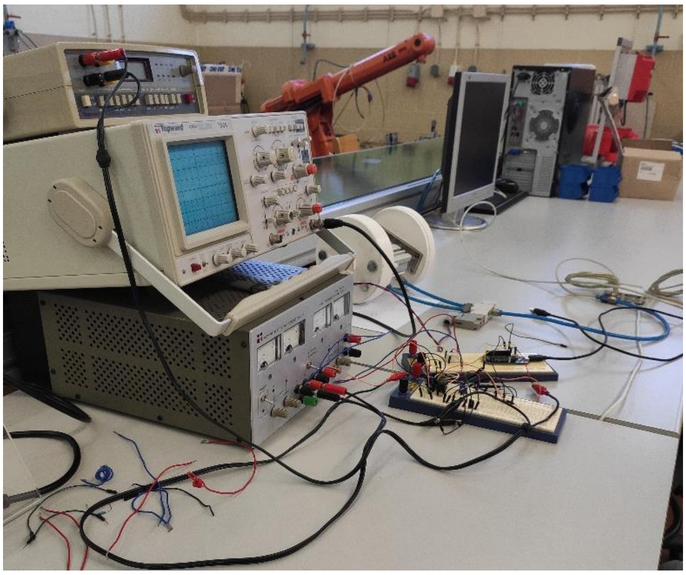

Figure 5.1. Initial circuit setup.

Furthermore, in Figure 5.1 is represented the initial prototype circuit with the hardware mentioned in chapter 3, except for the DAC breakout board and the MKR shield that had not yet arrived at the time. In Figure 5.2 there is a scheme representing the circuitry on the left breadboard. This left and right breadboard notions can be better acquired in Figure 5.5, page 30, that focuses on both breadboards from the real initial circuit setup.

Moreover, on the circuitry on left breadboard the holes in the leftmost column of the breadboard are connected to ground and the holes in the rightmost column are supplied with 24 V. This way, there is no need to connect to the source each time a tension of 24 V or a connection to the ground is required, which would be unfeasible.

In the breadboard to the right, the leftmost column is also connected to the common ground and the *Arduino MKR1000* and 5 V relay.

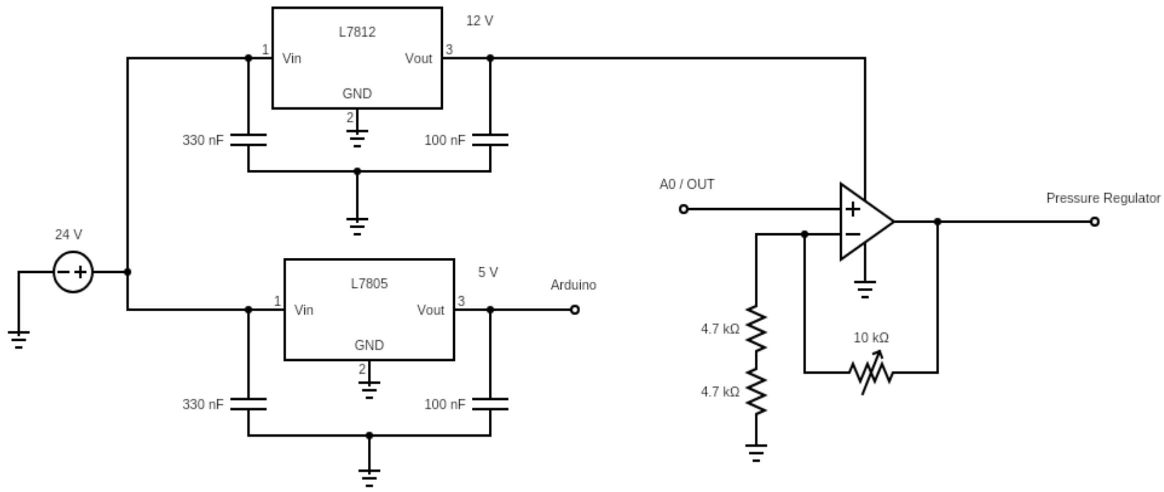

Figure 5.2. Electric circuit scheme.

Back to the left breadboard, the 24 V transforms into 5 V in the right part of the breadboard and into 12 V on the left, always with the multimeter being used to confirm the desired tensions in each part of the circuit are being achieved.

The 5 V path is destined to supply power to the *Arduino MKR1000* through the Vin pin. From the description of the *Arduino* in chapter 3.1.2 it is already known that it can also output a 5 V for powering another element from the 5 V pin, and initially that pin was occupied with a connection to a 5 V relay. This was useful because the operating tension of the MKR1000 is 3.3 V, so no other pin in this *Arduino* could output the 5 V required for the relay, represented in Figure 5.4, used in the prototype circuit. For this supply, however, the *Arduino* had to be powered by USB, which it was in this initial phase, but it is not the end goal since one of the main objectives in this thesis is to have remote control of the system.

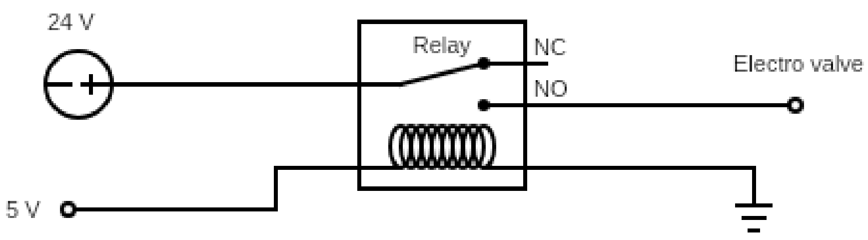

Figure 5.3. 5 V relay scheme.

To explain some more about this relay before progressing further, it was only used for this initial test phase to control the electrovalve because it would then be replaced by the built-in relays in the MKR shield. The need for it, nevertheless, concerns the electrovalve, which, as stated in the previous chapter 3.2.2, requires 24 V to be activated and, being a 2 position valve, works as a 0/1 (*Boolean*) system. This means, or it is 1 because it has a power supply and the valve advances to the second position; or it is 0 and there is no power being supplied, causing the valve to return to its standard position. The following scheme was created based on the scheme printed in the electrovalve to represent these 2 positions.

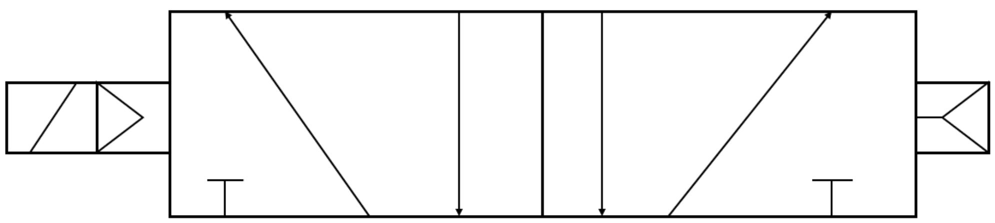

Figure 5.4. Electrovalve positions scheme.

On the other hand, back to Figure 5.2, the 12 V path is utilized to power the operational amplifier. This amplifier is part of an amplification process in which the purpose, as the name says, is to “amplify a signal” arriving between 0 V and 3.3 V and increase it to values between 0 V and 10 V corresponding to the pressure regulator’s input values range. This signal was intended to come from the *Arduino*’s DAC pin in an initial phase, but after testing, it would end up being replaced by the DAC breakout board analogic signal.

This last board was introduced to the project because the DAC pin (A0) of the *Arduino* was not permitting the usage of the full frame of tension values intended, especially the lower one’s bellow approximately 0.7 V, once those were amplified. This would cause the system to not be prepared to operate in the low force ranges required. The solution was then to find a board, Sparkfun I^2C DAC Breakout – MCP4725, that would function as DAC itself and theoretically make these lower force values achievable.

To measure the output from the pressure regulator between the tensions of 1 V and 5 V, coming into the analogue input of the *Arduino*, the incoming tension had to be adjusted to do not exceed the 3.3 V of *Arduino*’s operating tension.

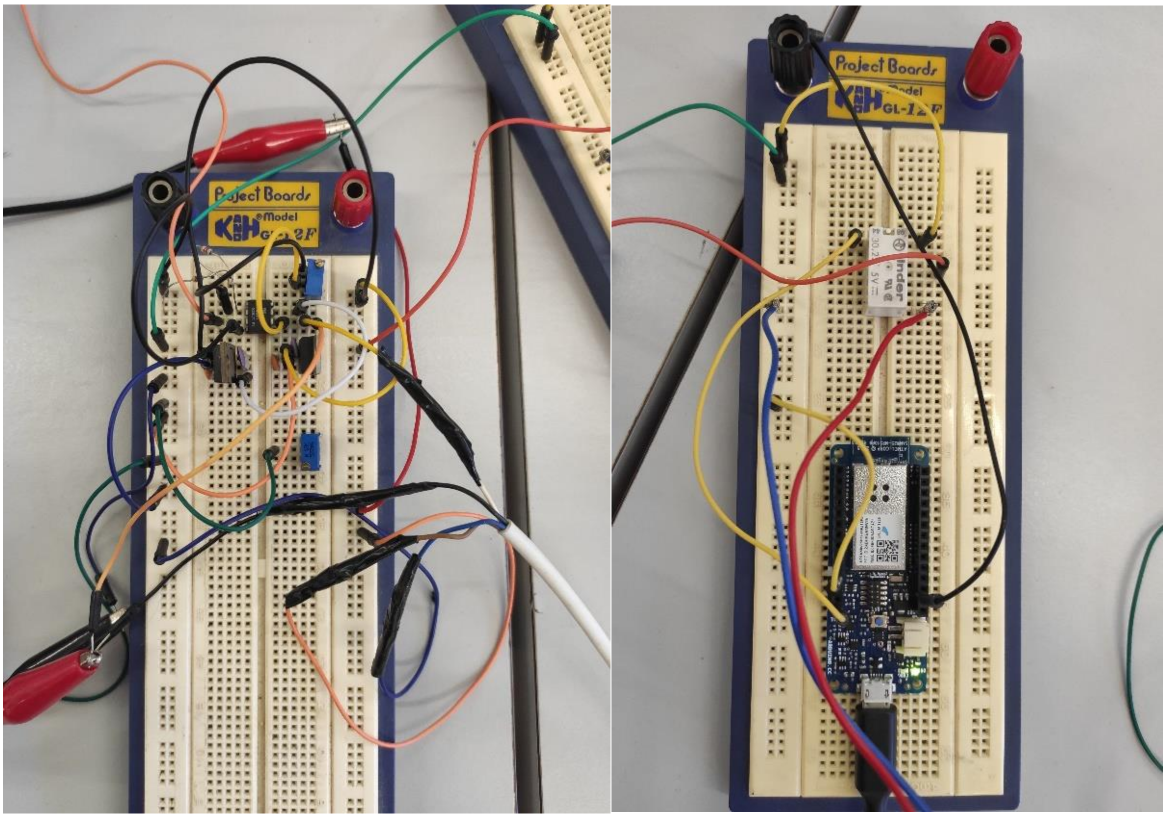

Figure 5.5. Breadboards connections.

This entire circuit, presented in Figure 5.5, would be much simplified with the introduction of the shield to the system, that for its functionalities presented in 3.1.3 would compact the circuit and make it possible to prepare tests for the system as a whole.

5.2. Arduino Control and Communication Protocols

Through the *Arduino* IDE introduced in the last chapter, it was needed to program the board to receive and send the signals described. The *Arduino* acts like a dull figure who is given a series of instructions and executes them. One of the advantages of it being an open-source software is that it already offers a vast quantity of libraries made by developers.

This project required communication via Wi-Fi and for this, the libraries <SPI.h> and <WiFi101.h> were needed to be included to establish that communication. The first allows the communication to the board's Wi-Fi system, that is already built into the same, via Serial Peripheral Interface (SPI) bus; as for the second allows the board to connect to the network.

Following up, to use the DAC breakout board mentioned, theoretically some more libraries would have to be included in the code once this board was integrated into the circuit: <Wire.h> for the use of the I^2C lines and <Adafruit_MCP4725.h>, the library correspondent to the DAC board.

5.2.1. Writing and Reading Pressures

The *Arduino MKR1000*, as mentioned, can send and receive analogic signals, within a byte-range to devices like the pressure regulator, that are expressed into tensions. The value of pressure, in the pressure regulator, is known to be linear to the value of tension inputted, therefore, it is possible to assume a linear distribution as seen in Figure 5.6.

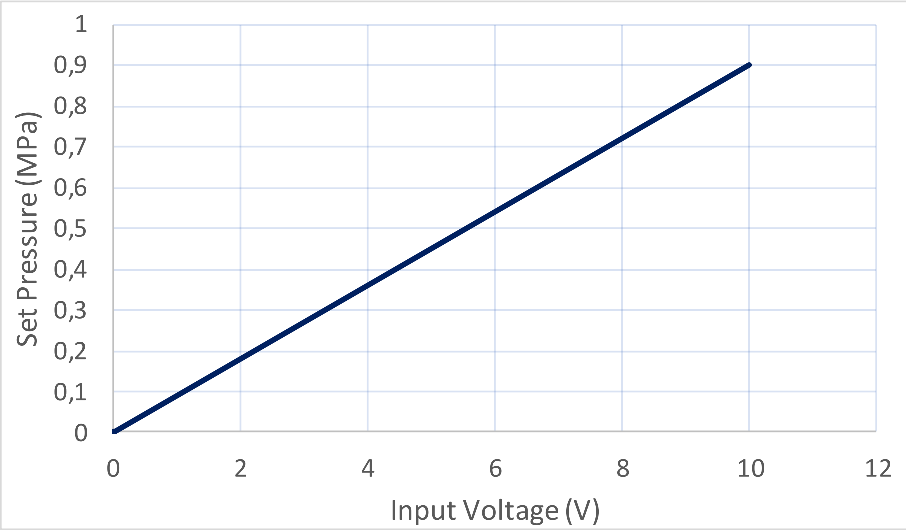

Figure 5.6. Graphic of the linear progression of set pressure.

Knowing this, the goal was then to discover what values to output from the *Arduino* according to this and what influence would they have into the real system.

Following this, in APPENDIX B it is presented a table with 7 parameters resumed in the next 5 points:

- the force setting option in the GUI (to be explained further);
- the number of bits the *Arduino* associated with the Breakout board sends as an analogic signal;
- the tension it corresponds to in the *Arduino* and the pressure regulator input;
- the pressure defined as set pressure;
- and finally, the theoretical forces in two different units.

This table was done in *Excel* to automatize the calculi of the values. Those, except for the theoretical force values, were obtained by linear interpolation:

$$y = y_1 - (y_1 - y_0) \times \frac{(x_1 - x)}{(x_1 - x_0)}, \quad (5.1)$$

and for the theoretical force values occurring in the cylinder, knowing that A_{bore} represents the area of the piston with a bore size of 40 mm and P the set pressure:

$$F = P \times A_{bore}. \quad (5.2)$$

To note that predicting the maximum value for the theoretical cylinder force output has its specific utility. During application, this value is not intended to exceed by much 3.5 kg.f, otherwise the robot would stop acting, as a safety procedure, because the maximum force it can withstand would be surpassed since the device itself will already weight around 1.5 kg and the maximum payload of the robot is 6 kg, as referred in chapter 2.2.

5.2.2. Server

On the *Arduino* IDE was created a server to establish communication with the local network. The IDE has serial monitor functionality that allows the programmer to monitor the messages sent and received from the *Arduino*. Through this, it was possible to generate a routine and see if it was all working properly.

What the *Arduino* does when the program is run and the serial monitor opened, is fetch the IP of the network. For this to happen, the name and password of the network it is intended to connect to must be expressed on the IDE. Once the connection is established, the IP from the network is written in the serial monitor.

This is the first step to create interaction with the user, however, now the program requires instructions to do something. It would be possible to get the system working just by interacting with the values in the *Arduino* code. Nevertheless, this way of interaction would require more knowledge about programming than it is desirable in a real-world application since one the primary objectives of having a controlling software is not that the user is proficient in programming it, but instead knows how to use it without much effort.

For this to be achieved, emerged the need to create a graphical user interface (GUI), which will be presented in the next subchapter.

5.3. GUI for communication with *Arduino*

The communication between the *Arduino* as well as the robot was done through a graphical user interface (GUI) created with *Microsoft Visual Studio* in C#. This interface communicates to the systems in the same network through network sockets with TCP/IP protocols.

TCP/IP is constituted by 2 main protocols that define the connection: TCP, which defines how the application transmits a network communication and how that transmission is broken down; and IP that establishes the address to which the data is sent and received from. Associated with the IP there is also a Port that addresses to the correct location within the device.

In this system, there are 3 different IPs to be accounted for: the IP from the *Arduino*, which is the one used directly in this dissertation, the generic IP used for the *Robot Studio* simulation, and the IP from the *ABB* robot. These last two are part of a communication protocol developed in my colleague Nuno's dissertation (Gomes, 2020) and, therefore, all the communication with the robot/*robot studio* will be developed in full on his dissertation.

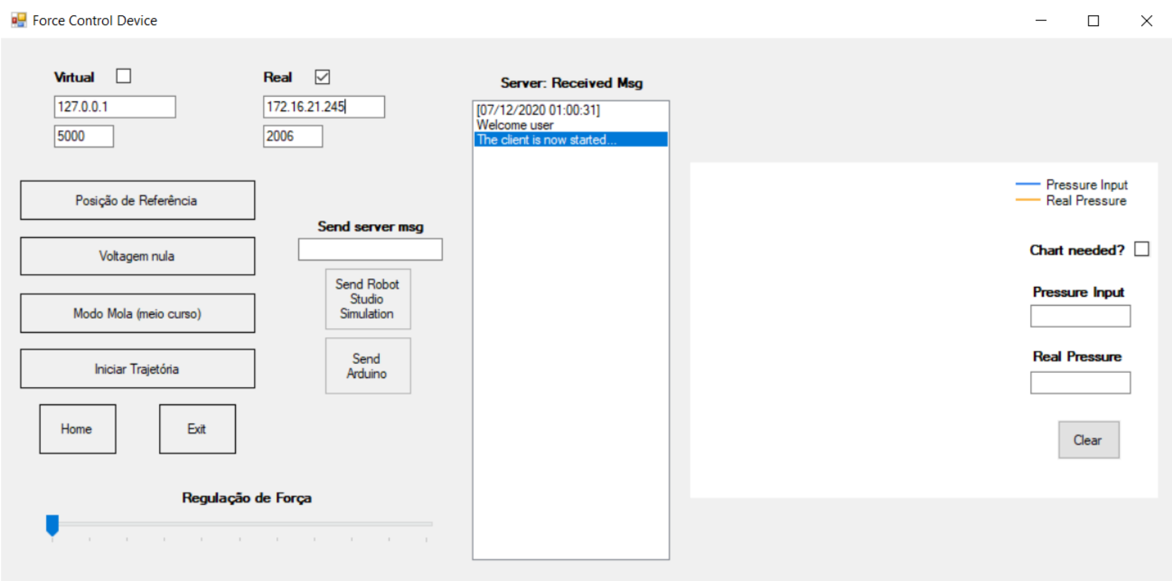

Figure 5.7. GUI – Real checked.

Continuing in the elements of the interface, two realities were incorporated – virtual and real. In the case of virtual reality, communication was oriented to the virtual environment of *Robot Studio*, whereas in real, it would be destined to the controlling mechanism, in which the *Arduino* is incorporated. Then, many buttons worked accordingly to the capabilities of each reality. For this dissertation, the only relevant reality was chosen by checking the real checkbox.

Once this was done, all the interactive elements of the GUI were enabled, and the client becomes ready to perform.

Under the “Send server msg” label there is a text box where both the systems fetched the message/command and sent it to the respective socket. Those messages could be written manually inside the box, but that does not offer the desired easy functionality of a GUI because the user would have to know the exact name of the command to send it. However, an editable text box like this, allows the programmer to test new commands without having to create new buttons every time. Anyways, the buttons and trackbar presented in Figure 5.7 were all programmed to send the command associated with each one.

After the command was printed on the text box, depending on wherever the GUI is supposed to send the value to, the button “Send Robot Studio Simulation” or “Send Arduino” are automatically clicked and the communication protocol between the client and the device begins.

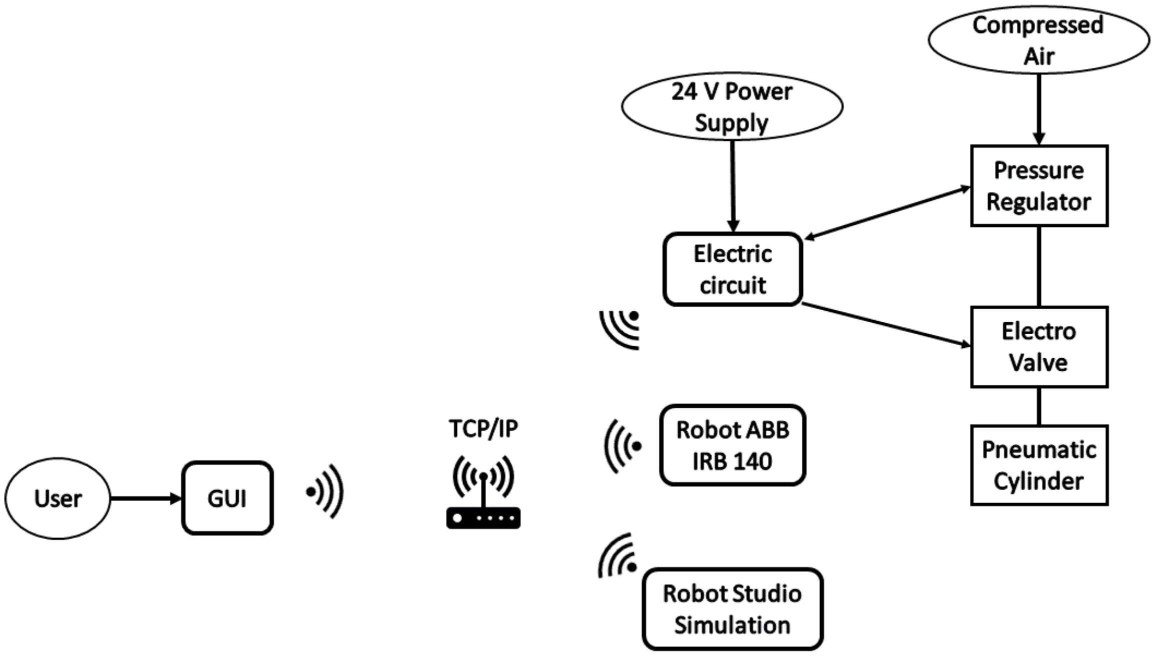

Figure 5.8. Communication scheme of the full system.

Figure 5.8 depicted above goal is to completely illustrate the communication process conceived in this and Nuno’s, (Gomes, 2020), dissertation. The “electric circuit” englobes the *Arduino* as the primary element, but also the other electric components already presented.

Proceeding on the communication protocol associated with this dissertation, the client basically fetches the IP and the Port under the “Real” checkbox and tries to establish communication with the *Arduino*. If that is accomplished, the command is sent and then a message correspondent to the command given is written in “Server: Received Msg” text box along with the time when it was received.

The elements in the interface that promote this communication are the button correspondent to null tension and also the trackbar. It is, however, convenient to first understand how the messages were recognized by the *Arduino*. The messages each element prints on the “Send server msg” text box are sequences of strings separated by spaces, “ ”, and are then interpreted by the *Arduino* as parameters. The first, parameter[0], always corresponds to a command, and the following to parameters of that command, e.g.:

- Command“ ”Parameter[1]“ ”Parameter[…].

Despite the *Arduino* code being prepared to accept 6 different parameters apart from the command, in this dissertation it was only needed to go as far as one parameter.

The button “Voltagem nula” sends a command that is recognized by the *Arduino* as “voltagem_nula” which output a digital HIGH signal that turns the relay to the NO position, turning on the valve, and at the same time sets the tension in the pressure regulator to its minimum.

On the trackbar labelled “Regulação de Força”, when the slider is used, the command “pressao” is sent and as the slider is moved parameter[1] changes between the integer values of [0;10] being then converted to a string and the message is sent in the format:

- “pressao”“ ”“parameter[1].

How that theoretically translates into values throughout the system is, once again, made explicit in Table 8.1 on APPENDIX B.

To illustrate the ongoing communication between the GUI and the *Arduino* an example with purely fictional values is represented in Figure 5.9 on the next page.

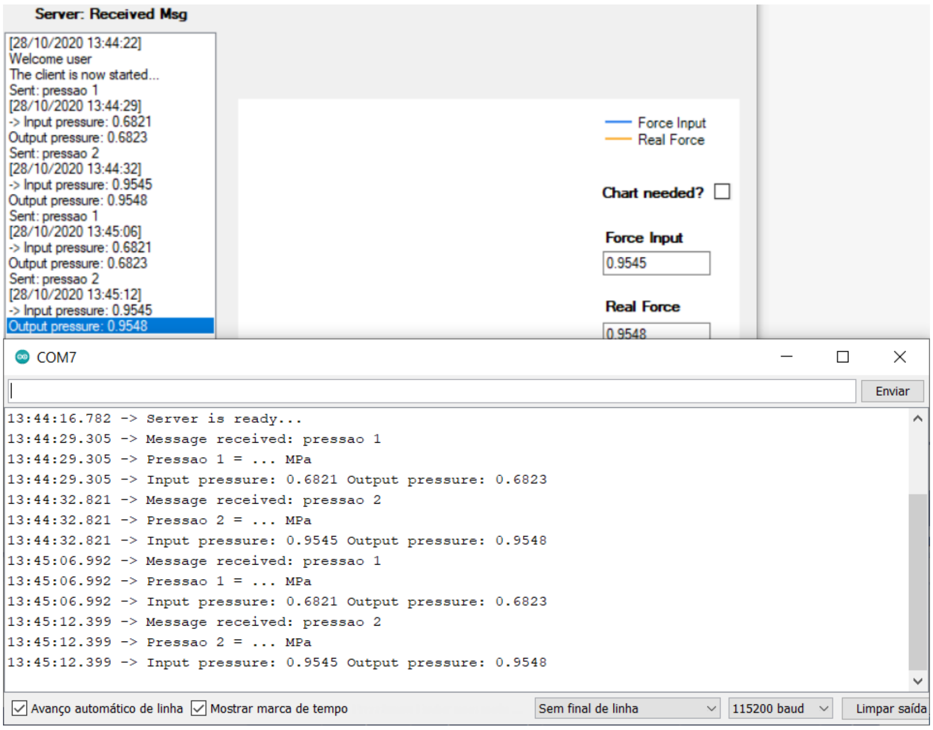

Figure 5.9. Communication between GUI and *Arduino*'s serial monitor.

Finally, when the pressure is applied and the device faces contact force alterations, the pressure of the system suffers a disturbance before it returns to the set value of pressure again. To observe this disturbance, a dynamic chart was created in the client using the input and output signals from the pressure regulator.

The option to have this chart displayed or not is given to the user through the checkbox after "Chart needed". Once selected and a value of set pressure is changed, a timer with a predefined interval will begin looping the output and input values and register them in the respective text boxes and further on the chart. The user can then stop this process anytime by simply unchecking the checkbox.

The full logigram of the communication protocol between the client and the system (end-effector + robot) is presented in APPENDIX A.

The connection quality is an important factor in this process of communication. Considerable differences in performance have been observed from one network to another probably because of existent noise or filtration in the network. This can corrupt the communication process, especially when the chart is activated because it demands constant information trade between the GUI and the *Arduino* server in very short time periods.

5.4. Pressure chart

As just referred in chapter 5.3, the GUI has a chart available to compare the input values with the output values of pressure occurring throughout time. The values were expressed on the chart with, recurring to a timer that looped every 100 milliseconds and extracted these values from the *Arduino*.

At the time in which this thesis was written however, some equipment was still missing and because of that, it was not possible to test this chart in the real system. Even so, by replacing the I/O values with some basic functions in the *Arduino* program to simulate those values in a fictional way, it was still possible to check that the chart was working properly in theory. Figure 5.10 exhibits an example using those fictional values with the only purpose being to show that the chart works.

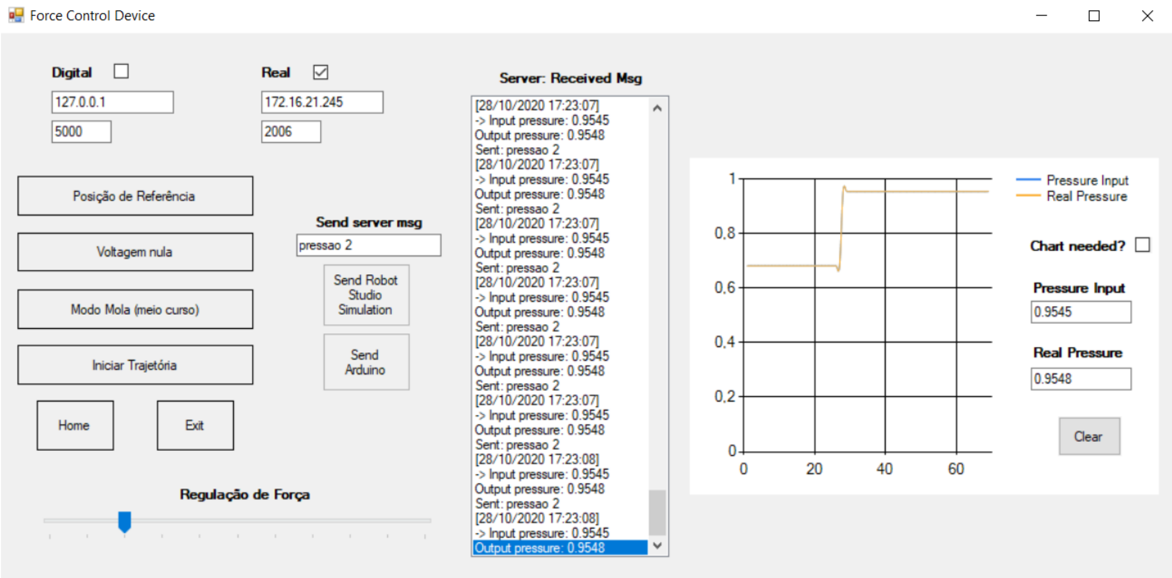

Figure 5.10. Pressure chart.

6. TESTS AND RESULTS

In light of what was developed in the previous chapter, there were some results to consider that proceeded from the testing of the system throughout its development. Some of those results were simply obtaining the correct regulated tensions, described in chapter 5.1, in the different parts of the circuit, since the assembling of electronics was made with constant testing of tensions with the measurement of the same by the proper equipment. One of these tests resulted in the need to acquire more equipment than originally planned to, specifically the Sparkfun breakout board described in chapter 3.1.4.

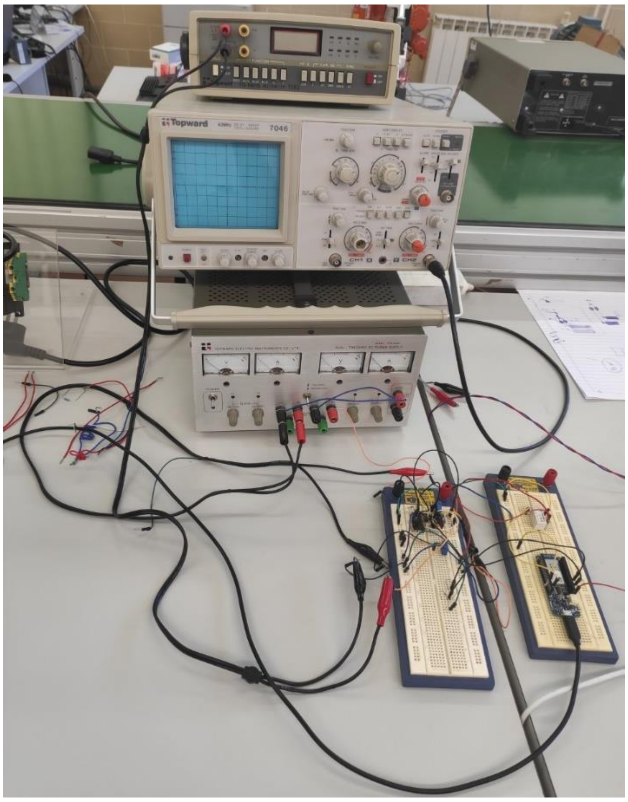

Figure 6.1. Electronics testing setup.

The different aspects of communication between the pneumatic components and the *Arduino* as well as between the user interface and the *Arduino* were also tested as much as possible in an experimental environment, like the description of the developed experimental work in the preceding chapter makes noticeable. In this case, the results have shown that the system was able to manage the pressure of the compressed air, and its flow through the system.

Due to the, already referred, continuous delays in obtaining some of the necessary equipment, some of the desired results, as the correction of the minimum output value (input to the pressure regulator) to 0 V, were not possible to attain. These delays also caused constraints in testing the full system with all the support components linking the system to the robot.

7. CONCLUSIONS AND FUTURE WORK

This final chapter presents the conclusions to make from the work developed and the future work prospects.

7.1. Conclusions

From the work done in this project, it is reliable to conclude that at least the bases of a new system were developed on the different aspects that concern this system.

Particularly, in this dissertation, the electric circuit was assembled according to the necessities of the project and was efficient, enabling the *Arduino* to send and receive signals from the system's pneumatic components.

It is also safe to conclude that the *Arduino* was the right choice for this system since the program in it ran without issues nor lags while serving its purpose.

The connection between the GUI and the *Arduino* was also a success since the user could not only send and receive information from the *Arduino* but also express that information graphically in an intuitive manner.

All of the above are enough to conclude that the 3 objectives set in chapter 1.2 were achieved. As for the functionality of the system itself, it was not possible to test the real complete system as much as desired because of the previously expressed delays in obtaining items needed to progress further in the work. Therefore, some limitations like making it able to do more complex applications with, e.g., contact forces in multiple directions, opened the way for future work.

7.2. Future Work

On the future work segment, there is a continuity of tests to be done to make the system more flexible and also less prone to errors. On this last one, a solution would possibly be to implement a closed-loop control algorithm like, or partially like, PID control as referred in chapter 2.1.2. Partially because not all the constants in PID control, also previously referred, if any, would be easy to determine and this process would need to be assisted with external sensors like, e.g., a piezoelectric force sensor. Furthermore, the system could also only require a simpler controlling loop, like a proportional controller for the desired efficacy to be achieved.

BIBLIOGRAPHY

ABB. (n.d.). *IRB 140 - Industrial Robots (Robotics) - Industrial Robots from ABB Robotics*. Retrieved October 22, 2020, from https://new.abb.com/products/robotics/industrial-robots/irb-140

Agrawal, M., Eloot, K., Mancini, M., & Patel, A. (2020). Industry 4.0: Reimagining manufacturing operations after COVID-19. *McKinsey*, *July*. https://www.mckinsey.com/business-functions/operations/our-insights/industry-40-reimagining-manufacturing-operations-after-covid-19

Bahrin, M. A. K., Othman, M. F., Azli, N. H. N., & Talib, M. F. (2016). Industry 4.0: A review on industrial automation and robotic. In *Jurnal Teknologi* (Vol. 78, Issues 6–13, pp. 137–143). Penerbit UTM Press. https://doi.org/10.11113/jt.v78.9285

Carvalho, J. (2020). "*Desenvolvimento um sistema de controlo ativo de força para a indústria - projeto mecânico e construção"*. Tese de Mestrado em Engenharia Mecânica na especialidade de Produção e Projeto, Departamento de Engenharia Mecânica, Faculdade de Ciências e Tecnologias, Universidade de Coimbra.

Ciraldo, J. (2020). How IIoT Maximizes Manufacturing Efficiency. *Https://Www.Beekeeper.Io/*. https://www.beekeeper.io/blog/how-iiot-maximizes-manufacturing-efficiency/

FerRobotics Inc. (n.d.). *ACF - Ferrobotics*. Retrieved October 16, 2020, from https://www.ferrobotics.com/en/technologie-produkte/produkte/acf/

Festo. (n.d.). *Digital pneumatics | Festo Motion Terminal*. Retrieved October 15, 2020, from https://www.festo.com/vtem/en/cms/10169.htm

Garcia, E., Jimenez, M. A., De Santos, P. G., & Armada, M. (2007). The evolution of robotics research. *IEEE Robotics and Automation Magazine*, *14*(1), 90–103. https://doi.org/10.1109/MRA.2007.339608

Gibb, A. M. (2010). *NEW MEDIA ART, DESIGN, AND THE ARDUINO MICROCONTROLLER: A MALLEABLE TOOL*. Pratt Institute.

Gomes, N. (2020). *"Desenvolvimento um sistema de controlo ativo de força para a indústria – simulação, software de operação remota e teste"*. Tese de Mestrado em Engenharia Mecânica na especialidade de Produção e Projeto, Departamento de Engenharia Mecânica, Faculdade de Ciências e Tecnologias, Universidade de Coimbra.

Hermann, M., Pentek, T., & Otto, B. (2015). *(PDF) Design Principles for Industrie 4.0 Scenarios: A Literature Review*. https://www.researchgate.net/publication/307864150_Design_Principles_for_Industrie_40_Scenarios_A_Literature_Review

International Federation of Robotics. (2020). *Executive Summary World Robotics 2020 Industrial Robots*.

Li, J., Guan, Y., Chen, H., Wang, B., Zhang, T., Liu, X., Hong, J., Wang, D., & Zhang, H. (2020). A High-Bandwidth End-Effector With Active Force Control for Robotic Polishing. *IEEE Access*, *8*, 169122–169135. https://doi.org/10.1109/access.2020.3022930

Mailah, M., & Rahim, N. I. A. (2000). Intelligent Active Force Control of a Robot Arm

Using Fuzzy Logic. In *ieeexplore.ieee.org*. https://ieeexplore.ieee.org/abstract/document/888750/

Meon, M. S., Mohamed, T. L. T., Ramli, M. H. M., Mohamed, M. Z., & Manan, N. F. A. (2012). Review and current study on new approach using PID Active Force Control (PIDAFC) of twin rotor multi input multi output system (TRMS). *SHUSER 2012 - 2012 IEEE Symposium on Humanities, Science and Engineering Research*, 163–167. https://doi.org/10.1109/SHUSER.2012.6268848

Mohsin, I., He, K., Li, Z., & Du, R. (2019). Path planning under force control in robotic polishing of the complex curved surfaces. *Applied Sciences (Switzerland)*, *9*(24). https://doi.org/10.3390/app9245489

Pecina, E. (2018, October 11). *An Introduction to Arduino's MKR Family and IoT Development Boards | Arduino | Maker Pro*. https://maker.pro/arduino/tutorial/an-introduction-to-arduinos-mkr-family-and-iot-development-boards

Pires, J. N. (2019). *Apontamentos da disciplina de Control Indústrial inserida no Mestrado Integrado em Engenharia Mecânica na Universidade de Coimbra*.

Quintero, S. (2019, October 4). *Control Engineering | New ways to increase productivity with smart systems*. https://www.controleng.com/articles/new-ways-to-increase-productivity-with-smart-systems/

Reljic, V., Seslija, D., Milenkovic, I., Bajci, B., Dudic, S., & Sulc, J. (2019). Concept of remotely controlled digital pneumatic system in accordance with Industry 4.0 approach. *Proceedings of the 2019 5th Experiment at International Conference, Exp.at 2019*, 419–423. https://doi.org/10.1109/EXPAT.2019.8876544

Rojko, A. (2017). Industry 4.0 Concept: Background and Overview. *International Journal of Interactive Mobile Technologies (IJIM)*, *11*(5), 77–90. https://doi.org/10.3991/ijim.v11i5.7072

Thames, L., & Schaefer, D. (2016). Software-defined Cloud Manufacturing for Industry 4.0 | Elsevier Enhanced Reader. *ELSEVIER*. https://reader.elsevier.com/reader/sd/pii/S2212827116307910?token=7A2B204DB7FEDF6BEEF3EB13034118A09B6D472B31BF086D14C9A0DD5EE48FA72F273A18D2EDB4003DA612586B971C3C

Xu, L. Da, Xu, E. L., & Li, L. (2018). Industry 4.0: State of the art and future trends. *International Journal of Production Research*, *56*(8), 2941–2962. https://doi.org/10.1080/00207543.2018.1444806

APPENDIX A

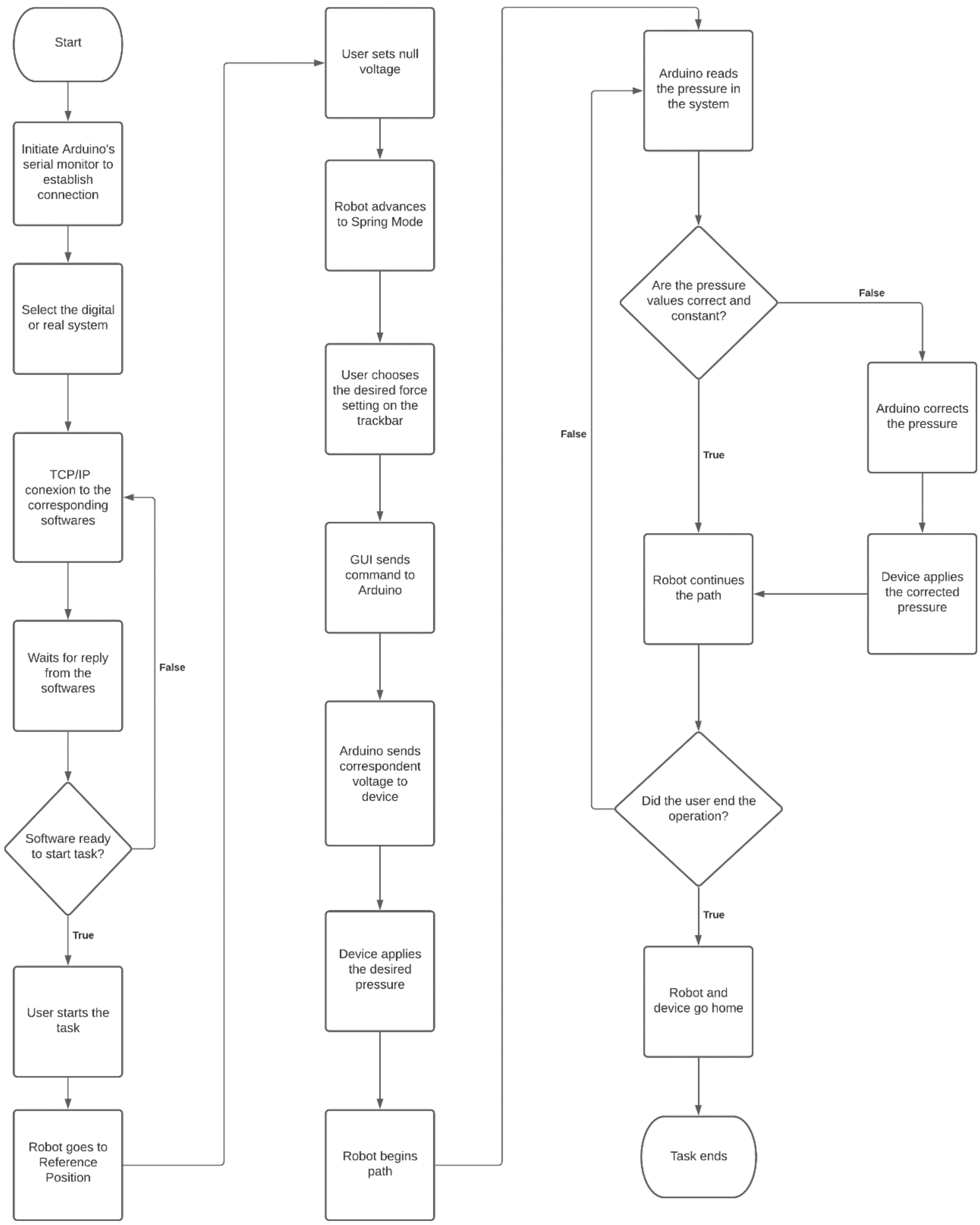

Figure 8.1. Logigram of the system's communication protocol.

APPENDIX B

Table 8.1. Set Pressure Table.

Force setting	12-bit range (0 - 4095) bits	Arduino Tension (0 - 3.3) V	Amplified Regulator Input Tension (0 - 10) V	Set Pressure (0.001 - 0.9) MPa	Theoretical cylinder force output (N)	Theoretical cylinder force output (kg.f)
0	0	0.000	0.000	**0.001**	1.257	0.128
1	12	0.010	0.029	**0.004**	4.566	0.465
2	24	0.019	0.059	**0.006**	7.876	0.803
3	36	0.029	0.088	**0.009**	11.186	1.140
4	48	0.039	0.117	**0.012**	14.496	1.478
5	60	0.048	0.146	**0.014**	17.805	1.815
6	72	0.058	0.176	**0.017**	21.115	2.152
7	84	0.068	0.205	**0.019**	24.425	2.490
8	96	0.077	0.234	**0.022**	27.734	2.827
9	108	0.087	0.264	**0.025**	31.044	3.165
10	120	0.097	0.293	**0.027**	34.354	3.502