

DEFINING AND EVALUATING AN EDUCATIONAL FRAMEWORK FOR VIRTUAL COMMISSIONING

Bachelor Degree Project in Production/Mechanical
Engineering
G2E, 30 credits
Spring term 2022

Josu Barrena Aseginolaza
Mikel Mugica Agirre

Supervisor: Martin Birtic

Examiner: Stefan Ericson

Abstract

Automated systems are large, complex, and expensive systems; therefore, it is often a better option to use virtual models of these systems for educational purposes. This project focuses on the definition and evaluation of an educational framework for virtual commissioning. The framework itself consists of a group of different software and environments interconnected between them. This is done by following the design and creation method. Acting in accordance with the methodology, the project starts with a clear identification of the need for a framework and follows an iterative process until getting a valid result. For this, first, it has been needed to select the appropriate software for the framework. Then this tentative framework definition is evaluated. Once the integration is valid, the framework itself is used to conduct the virtual commissioning of a manufacturing process that produces two different product variants. This is done to evaluate the framework in terms of virtual commissioning. The result is a framework fully integrated, being able to exchange different data types through the different environments. The final implementation design consists of three different environments that allow PLC and HMI programming, 3D modelling and robot programming, and connection to the cloud via an API and the implementation of a smart algorithm for the optimization of the makespan of a client order. The resulting framework is also one that allows the user to experience different activities that are performed when performing virtual commissioning, such as the PLC and robot programming and other IT solutions like the implementation of APIs and smart algorithms.

Certification

This thesis has been submitted by First name Family name to the University of Skövde as a requirement for the degree of Bachelor of Science in Production/Mechanical Engineering.

The undersigned certifies that all the material in this thesis that is not my own has been properly acknowledged using accepted referencing practices and, further, that the thesis includes no material for which I have previously received academic credit.



Mikel Mugica Agirre



Josu Barrena Aseguinolaza

Skövde 2022-06-03

Institutionen för Ingenjörsvetenskap/School of Engineering Science

Acknowledgements

First and foremost, we would like to thank our supervisor Martin Birtic for making this work possible and for his guidance and advice during the whole process.

In addition, we would also like to thank the University of Skövde for all the provided facilities that were necessary for the development of this thesis.

Finally, we would like to express our gratitude to all of the people that has supported us during this year.

Skövde, June 2021

Josu Barrena Aseguinolaza and Mikel Mugica Agirre

Table of Contents

1	Introduction	1
1.1	Problem Statement	1
1.2	Aim and Objectives.....	1
1.3	Delimitations	2
1.4	Overview.....	2
1.5	Sustainable development.....	3
1.5.1	Social sustainability	3
1.5.2	Economic sustainability	4
1.5.3	Environmental sustainability.....	4
2	Theoretical Frame of Reference.....	5
2.1	Virtual commissioning	5
2.2	Industrial automation and education.....	5
2.3	PLC and robots.....	6
2.4	Industrial communication protocol.....	6
2.4.1	OPC UA	7
2.4.2	ETHERNET IP	7
2.5	Software tools	8
2.6	Cloud storage / retrieval.....	9
2.7	Smart algorithms and optimization solvers	9
3	Literature Review	10
3.1	Virtual Commissioning in manufacturing systems	10
3.2	Virtual labs and virtual commissioning for educational purposes	11
4	Method.....	12
5	Implementation.....	15
5.1	Selection of the software for the framework.....	15
5.1.1	Modelling software selection	15
5.1.2	PLC programming software selection	16
5.1.3	Smart Algorithm / Optimization Solver Selection	17
5.2	Make the connection between different software platforms	19
5.2.1	RobotStudio-Codesys connection	19
5.2.2	Codesys-PyCharm connection	20
5.2.3	PyCharm-Google Sheets connection	20

5.3	3D Model Selection and Modelling	21
5.3.1	Functional Description.....	21
5.3.2	Conveyors and Robots.....	22
5.3.3	OPC UA Connection to PLC environment	23
5.4	PLC, HMI, and Robot programming.....	23
5.4.1	PLC programming approach	23
5.4.2	HMI	23
5.4.3	Robot programming approach	24
5.5	Cloud Data and Optimization solver development	25
5.5.1	Cloud Data	25
5.5.2	Optimization problem	26
6	Analysis and Results	29
6.1	Evaluation of the software integration	29
6.2	Evaluation of the educational framework in terms of virtual commissioning.....	30
7	Discussion.....	33
7.1	Future Work	33
8	Conclusions.....	34
	References.....	35
	Appendices	38
	Appendix 1 Work Breakdown and Time Plan.....	38
	Appendix 2 Software versions.....	39

List of Figures

Figure 1 Venn diagram for sustainable development	3
Figure 2 An automated system in its most basic form.....	6
Figure 3 Framework diagram	8
Figure 4 'Design and Creation' research method.....	12
Figure 5 Flowchart of the methodology applied to the project.....	13
Figure 6 Specific framework diagram.....	19
Figure 7 Excel file example for OpcUaClient configuration.....	20
Figure 8 Layout of the plant	21
Figure 9 Virtual cell in RobotStudio.....	22
Figure 10 Operator panel HMI	24
Figure 11 Customer order page of the Google Sheet	25
Figure 12 Production information page of the Google Sheet	25
Figure 13 Operations information page of the Google Sheet.....	26
Figure 14 Main elements of the program	27
Figure 15 Machine number and production time layout.....	27
Figure 16 Initial time plan.....	38
Figure 17 Final time plan.....	38

List of Tables

Table 1 Weight decision matrix for modelling software selection.....	15
Table 2 Weight decision matrix for PLC programming software selection.....	16
Table 3 Weight decision matrix for optimization solver	18
Table 4 Data type sent between software	29
Table 5 Activities involved in virtual commissioning.....	30
Table 6 Used software versions	39

List of Symbols

VC	Virtual Commissioning
PLC	Programmable Logic Controller
MES	Manufacturing Execution System
IT	Information Technology
HMI	Huma Machine Interface
API	Application Programming Interface
DT	Digital Twin
FAQ	Frequently Asked Questions
GVL	Global Variable List
RL	Reinforcement Learning
CP	Constraint Programming
I/O	Input/Output
OPC UA	Open Protocol Communication Unified Architecture
COM	Component Object Model
DCOM	Distributed Component Object Model

1 Introduction

Automated systems are large, complex, and expensive making them unsuitable for educational use. By utilizing software students can engage in virtual engineering of such systems. They can design and develop systems virtually getting a close-as-possible experience. With the right selection of virtual tools, they can model an automated system and then program it using a real Programmable Logic Controller (PLC) and robot code. Also, there are possibilities to connect the models to external programming environments such as Python. External programming environments can be used to deploy smart algorithms for data optimization, collection and presentation of model performance data, and simulation of higher-level systems such as a Manufacturing Execution System (MES system).

1.1 Problem Statement

As automated systems are getting more complex, educational efforts must match the demands of the industry. In this project, a virtual educational framework is defined and evaluated, with which students can perform many real-life tasks regarding virtual commissioning (VC). Historically VC focused on verifying and improving PLC and Robot programs but may now also focus on verifying and improving Information Technology (IT) solutions why an important addition to this project is to connect the virtual system with such tools.

The educational framework should enable students to develop skills related to virtual commissioning such as:

- Building/modelling automated systems.
- Controlling components and model behaviour using real PLC code and Human-machine interface (HMI).
- Controlling robots in the model using real Robot code.
- Connecting the model to an external programming environment.

1.2 Aim and Objectives

The aim of this project is to define and evaluate an educational framework for virtual commissioning that enables a close-to-real student experience. The main objectives are the following:

- Defining a software framework.
- Selecting software.
- Setting up and testing the framework.
- Analysis and evaluation of the framework.

1.3 Delimitations

Regarding delimitations, because the project will be carried out in 4 months, time is a limitation. Programming a Robot and the PLC of a large system model is time-consuming and modelling the system is not the only task that is going to be developed, so models will not be as complex as they are in the real world. Moreover, due to the limited time, the framework won't be evaluated in a student education situation which would be ideal and could be interesting for future work. Instead, it will be evaluated as it is explained in Chapter 4.

In addition, real hardware will not form part of the framework, since the project will be developed entirely virtually. It is true that using a real PLC to simulate the program, for example, has its benefits. However, the framework presented in this project can be implemented just with the use of a single computer.

1.4 Overview

This section goes over each of the report chapters briefly in order to introduce the reader to their content. This thesis is composed, in addition to the introduction chapter, of another 8 chapters.

Chapter 2 includes the theoretical frame of reference, where the main relevant concepts and tools for this project are described. These concepts are necessary to properly understand the development of the project. In chapter 3, recent scientific work related to the project is analysed and conclusions are drawn from their operation methods and terms that can be applied to this project. Chapter 4 describes the methodology followed to carry out this project. Chapter 5 documents the selection of the software that will form the framework, how the software are integrated, and the entire building process of a virtual commissioning model of a manufacturing plant. In Chapter 6, the framework is evaluated from two different points, evaluating the software integration that forms the framework and the educational framework in terms of virtual commissioning. Chapter 7, apart from discussing the followed methodology, also discuss the limitations that have arisen when evaluating the framework. Moreover, this chapter also considers future work that could be done in order to continue the research and outcomes of this project. Finally, the project's overall conclusions are presented in chapter 8.

1.5 Sustainable development

Sustainable development as (Commission on Environment, 1987) defines, “Sustainable development is a development that meets the needs of the present without compromising the ability of future generations to meet their own needs”. This article comprises two important principles:

- the concept of 'needs', in particular, the essential needs of the world's poor, to which overriding priority should be given;
- and the idea of limitations imposed by the state of technology and social organization on the environment's ability to meet present and future needs.

For achieving that sustainable development, it is important to obtain an equilibrium between humans and the environment. This requires a balance of the three major sustainability factors: environmental, economic, and social (Figure 1).

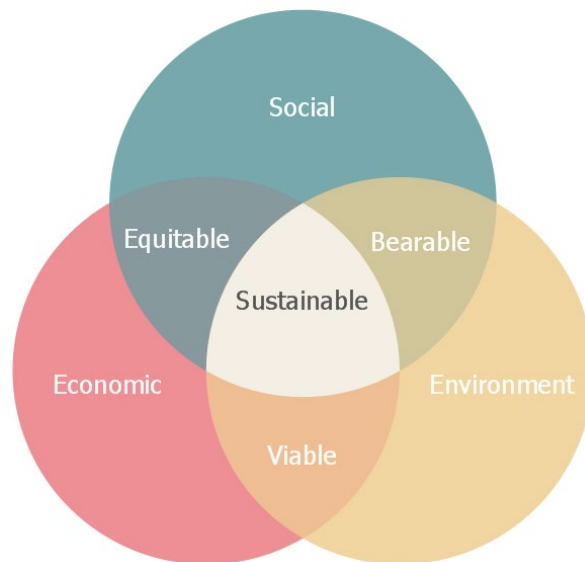


Figure 1 Venn diagram for sustainable development

1.5.1 Social sustainability

The main objective of social sustainability should be the continuous research of a future liveable community. In the case of this project, defining an educational framework for virtual commissioning not only helps students to have a close-to-real experience with automated systems, that are not suitable for educational use but also will be really helpful for workers' training. If the project's main objective is carried out successfully, students' and workers' knowledge would be widened, becoming more competitive. This has a beneficial impact on society since competition may lead to the development of new products that make people's lives better. Furthermore, automated systems improve workers' safety, since they can perform risky work that would be dangerous for workers. Adding this to the fact that they can carry out processes that cannot be done manually, the contribution to social sustainability is large.

1.5.2 Economic sustainability

This type of sustainability, as the name implies, is centered on capital. Its goal is to ensure resource conservation for future generations. In the case of the virtual educational framework, in the education environment, allows students to have a close-to-real experience with automated systems. Without the virtual environment, this would be impossible, since the costs of the physical systems are much higher than the framework software ones, being the number of resources needed much lower. Moreover, in the work environment, virtual commissioning reduces the wasted time on physical commissioning, being more effective and reducing the costs.

1.5.3 Environmental sustainability

The goal of environmental sustainability is to protect the world's resources by not squandering them unnecessarily. However, in the last decade, together with the growth of the human life quality, the consumption of the resources has brought the earth to a limit state and as previously said, maintaining a balance between human consumption and natural resources is crucial for sustainable development. That's why our project promotes environmental sustainability by saving energy, materials, and time by not depending on physical resources to simulate any type of workplace situation.

2 Theoretical Frame of Reference

2.1 Virtual commissioning

VC is a tool that serves to test manufacturing systems and their control programs through simulation on a virtual plant before the actual commissioning of the physical plant is realized (Hoffmann et al., 2010). In industry, the commissioning part usually takes 25% of the total time of the project. Moreover, 60% of that commissioning time is spent debugging the control software (Reinhart & Wünsch, 2007). Even if VC has been around for about two decades now, it is still not widely used in the industry. This is partly due to the lack of competencies and experience when integrating such virtual systems (Makris et al., 2012; Wöhlke & Schiller, 2005). That is why it is important to get an educational framework to train both the academic and industrial stakeholders on virtual commissioning.

There is no formal definition of what virtual commissioning should consist of, but there are different steps that need to be taken in the process. First, a complete model of the production system is done in a 3D environment, with its conveyors, robots, and other components. This model has all the dynamic and kinematic properties. Then a structure of all the signals is created that will later be connected to the PLC. Once all the connections are done, it is possible to test the PLC program, HMI panels, and different production parameters such as cycle time, and it can also check if any collisions happen.

Traditionally the scope of these models has been limited to the testing of the PLC and robot programs. However, the aim for the educational frame presented in this paper is to take a more holistic approach and include other elements of manufacturing systems, such as production optimization through different smart algorithms and the possibility to retrieve and collect data from the cloud.

2.2 Industrial automation and education

The qualifications and skills of engineers in future manufacturing systems will be one of the keys to the success of companies (Gehrke et al., 2015). Companies are using more and more new technologies that affect how the whole manufacturing process works. Changes in the industry need to be reflected in education. These new skills and qualifications are mainly in the area of new IT systems and services (Benešová & Tupa, 2017).

In order to face these changes, universities have come up with different ideas. Virtual commissioning and other kinds of virtual labs or factories have become an essential part of the educational process (Brazina et al., 2020; Mortensen & Madsen, 2018). These learning platforms have helped students get a multidisciplinary set of skills and they have also helped to achieve a better understanding of manufacturing systems (Mortensen & Madsen, 2018).

The opportunities given by these kinds of learning platforms are endless. The goal is to cover the most important and useful skills and qualities.

2.3 PLC and robots

There are four sorts of components in every automated system: sensors, actuators, industrial controllers, and the process or system. The most common type of industrial controller used in automated processes is PLC. It's a type of microprocessor-based controller that employs programmable memory to store instructions and perform activities like logic, sequencing, timing, counting, and arithmetic in order to operate machines and processes (Bolton, 2004). As a result, the major duty of the PLC is to manage the signals collected from the sensors, perform some operations based on the established logic, and then send commands to the actuators.

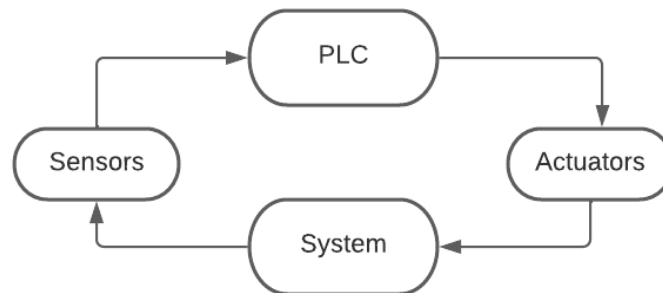


Figure 2 An automated system in its most basic form.

Robots, on the other hand, are programmed automatic machines that can execute some tasks autonomously and replace humans in particular jobs, especially those that are heavy, repetitive, or dangerous. Different types of robots can be utilized in the industry depending on the characteristics and requirements of the process.

Robots fall in the complex type of actuators. They cannot be actuated with a single signal from the PLC, they need to be programmed. This can be done in two ways: online and offline. The first step is to physically program the robot. Because production is halted during this time, this method is typically used to perform minor code changes. Offline programming, on the other hand, is done in a virtual environment. It is also used to run various simulations of the process, make more important changes, and test the robot.

2.4 Industrial communication protocol

Automated systems are composed of different elements such as sensors, actuators, PLCs... The PLC is the control logic of the system that collects data from the sensors, processes this information, and sends the corresponding command to the actuators. The feedbacks of the sensors are really important to have a clear view of what is happening in the system almost in real-time and send commands to the actuators depending on these feedbacks. Also, important information of the system performance (cycle-time, nº of ended products, nº of machines working...) is monitored by sending this information to a panel. This is why communication protocols are always integrated into every automated system.

Before, field buses, protocols that use serial communication between all components, were used to “replace the starlike point-to-point connections between the process control computers and the sensors or actuators” (Sauter & Member, 2010). However, in the last decade, new communication protocols have been developed being Ethernet-based solutions the most used ones. Nowadays, two of the most popular solutions are Open Protocol Communication Unified Architecture (OPC UA) and Ethernet IP.

2.4.1 OPC UA

The OPC Foundation published the OPC DA standard in August 1996, to resolve data communication issues between devices with diverse interfaces and protocols, as well as complete interconnection and interoperability between them.

Although the standard became widely accepted, it has some limitations (Ren et al., 2019):

- Platform limitations: the standard is very dependent on Component Object Model (COM) and Distributed Component Object Model (DCOM) technology of Microsoft platforms, making it difficult to implement OPC in platforms other than Windows.
- Security dependence on COM and DCOM: OPC specification lacks its security design, relying solely on the security of COM and DCOM.

To solve the limitations mentioned before, substitute the traditional OPC standard and meet automatic control systems requirements OPC UA standard has been developed. Compared with OPC, OPC UA has the following advantages (Ren et al., 2019):

- Platform independent: OPC UA can be developed on other systems apart from Microsoft.
- Safety performance: unlike OPC, the OPC UA standard integrates a completely secure communication mechanism.

2.4.2 ETHERNET IP

Ethernet IP is an industrial Ethernet protocol developed by Rockwell Automation. Ethernet IP is an application-layer protocol built on top of TCP/IP and employs the Common Industrial Protocol (CIP) over TCP/IP (Lin & Pearson, 2018). CIP is an object-oriented protocol that ensures the effective transfer of both implicit and explicit real-time network I/O information (Niu et al., 2021).

The communication model is based on a master-slave architecture, with slaves being able to communicate with one another since each device is connected to a switch. Every node filters the packets that are transmitted, so the switch's only task is to forward the ones that come in. Furthermore, the device's IP address could be used to establish direct communication (Salazar & Río, 2019).

2.5 Software tools

To build a virtual model of an automated system and achieve virtual commissioning, some advanced simulation software is needed (Figure 3):

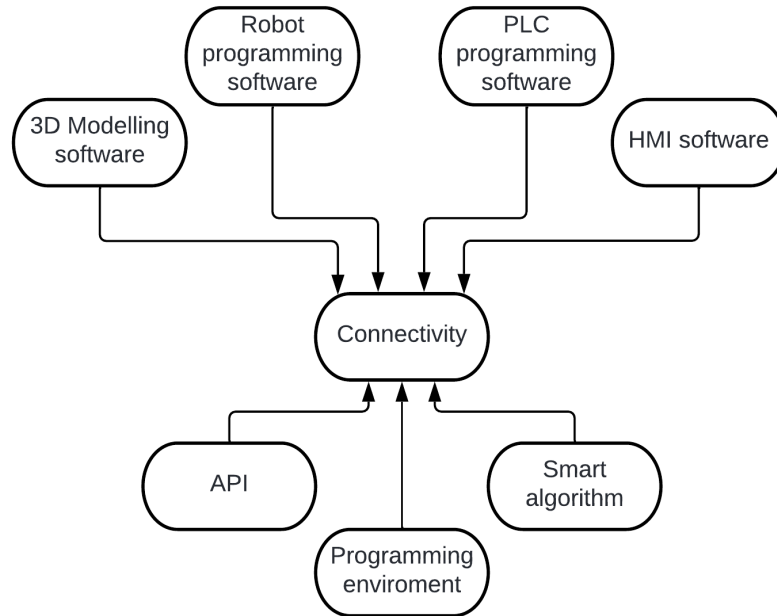


Figure 3 Framework diagram

First, it is necessary to develop a 3D model of the system, including all the elements, such as conveyors, robots, sensors, actuators... These elements are crucial to simulate the behaviour of the system correctly. In case the automated system contains an industrial robot, robot manufacturers such as ABB, Kuka... provide their own software. Apart from the software the robot manufacturers provide, they also are manufacturer-independent robot simulation software like Process Simulate, FastSuite... that allow programming and simulation of robots from various manufacturers.

Once the model is built the PLC must be programmed in order to control the system behaviour. As in the previous case, PLC manufacturers such as Siemens and Allen-Bradley provide their own software, TiaPortal, and Studio5000, while in the case of Codesys, it is an independent software and allows programming PLCs from many manufacturers.

Regarding HMI software, all PLC programming software mentioned before integrates HMI software but they also are software like FactoryTalk View Studio that is focused solely on HMI programming.

With the software mentioned above, a basic simulation of a production system can be developed, however, to model a higher-level system such as an MES system, a programming environment is needed in order to use cloud storage, smart algorithms...

Before talking about different programming environments, the most known programming language that needs to be mentioned is Python (Google, 2022a). As (IRJET Journal, 2017) defines, "Python is a very high-level, dynamic, object-oriented, general-purpose programming language that uses interpreter and can be used in a vast domain of applications". Moreover, Python has replaced Java as the most popular introductory language due to its appeal as a beginner-friendly language.

For Python developers, two of the most used programming environments are PyCharm and Spyder. Both software is Integrated Development Environments (IDE), which are full-featured environments that provide all of the necessary tools for software development. Apart from handling the code (write, edit...), debugging, execution, testing, and code formatting are among the various tools available to programmers.

Regarding smart algorithms, they include all Artificial Intelligence-based algorithms used in Digital Assistants, Robotic Process Automation, and Machine Learning. In an automated production system, smart algorithms could be implemented for energy reduction of the production plant, job shop scheduling to optimize and minimize the makespan...

To obtain the data from the system and store it on a file or to use the file information on the system an Application Programming Interface (API) is needed. An API is an intermediary software that allows two applications to talk to each other.

2.6 Cloud storage / retrieval

Cloud storage allows users to access their cloud storage space through Internet from anywhere and at any time using a desktop or mobile device, allowing them to view and back up their data flexibly. In a production system, this is really useful to store production data, such as finished product number, cycle time, and also to store the orders of the customers. This type of information is normally used by many departments in a company, not only the manufacturing system, so the cloud is crucial for accessibility.

2.7 Smart algorithms and optimization solvers

Smart algorithms are being introduced to the industry to optimize different aspects of manufacturing processes. There are many applications, such as defects detection in production lines, predictive maintenance, energy consumption forecasting, or manufacturing scheduling optimization. This project will try to implement a scheduling optimization algorithm in the manufacturing system. The scheduling of a manufacturing system is a very difficult scheduling problem because it involves all the aspects of the processes: order, resources, transportation system, perturbation factors such as breakdowns of a machine, etc. Typically, the scheduling problem is a problem modeled in mathematical form. The goal is to solve this equation in a reasonable time with an optimal solution, and of course with minimal resources. Therefore, to run these algorithms, it is necessary to connect the model to an external programming environment.

Additionally, there are software packages called optimization solvers that solve even more complex optimization problems than the afore mentioned smart algorithms. In practice, these solvers have algorithms embedded in them and can be used to solve many optimization problems. A programming environment is also needed to deploy them.

3 Literature Review

3.1 Virtual Commissioning in manufacturing systems

VC, as defined in chapter 2.1, is a tool that serves to test manufacturing systems and their control programs through simulation on a virtual plant before the actual commissioning of the physical plant is realized.

In 2015, some researchers developed a reconfigurable assembly system using virtual commissioning to plan, validate and optimize it (Vermaak & Niemann, 2015). In their research, they explained how “DELMIA” software was employed to develop a virtual simulation environment in order to verify an assembly cell. They carried out simulations to check software functionalities, device motions and operations, and the system’s control software. The findings of the research confirm that virtual commissioning can be used to speed up the planning, verification, and optimization of a system without endangering the real system equipment. It also proved that apparent design flaws can be fixed early in the design process, analysis can be performed to validate modifications to a system, and it can be determined if building a system is economically worth it.

Another paper that shows how useful VC can be is (Barbieri, Bertuzzi, Capriotti, Ragazzini, David Gutierrez, et al., 2021). The researchers proposed a methodology based on VC to respond to the issue of developing a Digital Twin (DT) and integrating it into a manufacturing system. This methodology was validated through the integration of a DT into a flow shop for the implementation of a scheduling reactive machine breakdown. The study was concluded by noting three main benefits that the methodology provided in comparison to the direct implementation of the DT into the manufacturing system:

- “Algorithms for the decision-making: the virtual interface between the intelligence layer and the DT enables the selection and tuning of the algorithms utilized for the decision-making”.
- “Changes in the original manufacturing system: the VC simulation enables the verification of the changes implemented in the original manufacturing system to integrate the DT”.
- “DT architecture: the virtualization and interface of all the actors allow the generation of a virtual environment to simulate the DT architecture and to identify possible issues that would occur in the physical implementation”.

Both articles, although they have implemented VC for different purposes, they have reached the same conclusion. VC not only allows to validate modifications before implementing them in the real system but also helps to identify possible issues in the design process or problems that could arise during physical implementation. Moreover, the second article (Barbieri, Bertuzzi, Capriotti, Ragazzini, David Gutierrez, et al., 2021) introduces algorithms for decision making which is a feature that is emerging during the last few years and it is not so common in VC. However, the VC of this project would include not only smart algorithms but also cloud storage/retrieval which is also an important part of an MES system.

3.2 Virtual labs and virtual commissioning for educational purposes

The introduction of software tools for testing control programs and virtual commissioning in industrial automation projects has reduced lead times and improved product quality, but it has also highlighted the need for skilled experts in these areas. By recreating and solving industrial problems in the classroom, the academic setting can help in the education of future professionals. Research has been conducted on how these skills should be taught and how effective these systems are.

An educational case study that was conducted in 2019 compares traditional working methods against the usage of a digital twin, highlighting the advantages of the latter (Fernández et al., 2019). They used the virtual commissioning of a robotic cell as a case study. The virtual model was used to verify the control PLC program developed by students. They concluded that by using emulation tools, students got a better understanding of the process, and the PLC program was more rigorously tested. Also, after validation, the use of the real equipment was more effective, and the commissioning time was shorter.

In another paper (Mortensen & Madsen, 2018) developed an industry 4.0 learning factory called the AAU Smart Production Lab. The paper shows the setup of a virtual commissioning platform and shows how various students have worked with it. The platform aims to help students in two main aspects: the modelling of physical and logic devices and process planning and system control modelling. The paper shows two learning activities in which the platform was used. The platform consists of three main parts, MES, PLC racks, and a virtual model. They also had a physical learning factory that offered the opportunity to perform the real commissioning. In this case, they reached similar conclusions, students got a deeper understanding of the process and acquired a multidisciplinary set of skills.

(Brazina et al., 2020) presents virtual commissioning as part of the educational process. They focused on the virtual commissioning of a robotic production system with elements of industry 4.0. They present different software tools to achieve virtual commissioning and they also present several steps to carry out the teaching of VC:

1. Component and layout design
2. Operation and component kinematic creation
3. Communication preparation
4. PLC program creation and virtual commissioning
5. Validation

In this case, also, they could validate the project on a physical system with real hardware. They concluded that students acquired knowledge and skills required in Industry 4.0

All the related work demonstrates that the use of VC in the learning process is very beneficial for the students. They feel more engaged with the work they are doing and acquire a deeper understanding of the processes they are automating. As a result, they get better knowledge and skills. All the approaches covered the aspects of PLC and robot programming. However, not that many cases worked on activities such as process planning or MES. The framework presented in this paper will not only cover the PLC and robot part but will also try to cover other aspects that are key in Industry 4.0

4 Method

The aim of this project is to define and evaluate an educational framework for virtual commissioning. This project does not only have a technical part which is the development of the framework, but it is also a research project since it examines the current state of knowledge and implementations in this field of study. This sort of project, according to (Oates, 2006), helps in the generation of knowledge. In this case, this is done by establishing a new framework for virtual commissioning. The methodology described in this chapter is used to develop this project.

As previously mentioned, the project's main goal is to define an educational framework for virtual commissioning that consists of different software tools and technologies, therefore the followed methodology will be the one named "Design and Creation" (Oates, 2006). This approach focuses on the creation of new IT products, also called artefacts, in order to contribute to knowledge. In our case, the educational framework itself will be the so-called artefact.

The design and creation problem-solving approach is an iterative process that involves five steps as shown in Figure 4. These steps are not followed strictly, they describe a fluid iterative cycle (Oates, 2006).

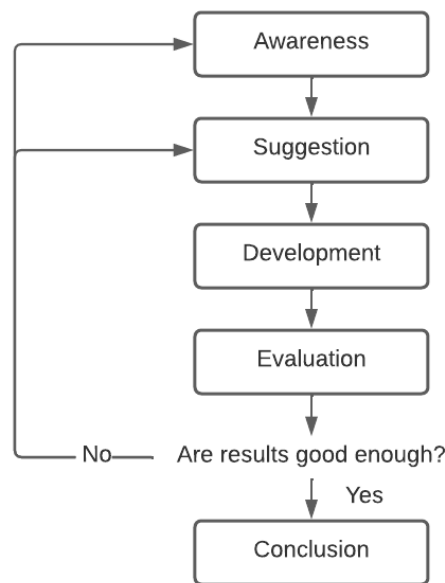


Figure 4 'Design and Creation' research method

Here are the steps that need to be followed a little bit more in-depth:

- Awareness: is the recognition and articulation of the problem. This is carried out with the help of the frame of reference and the literature review in which authors identify areas for future research or from new technological advancements.
- Suggestion: offering a possible idea that might work as a solution for the problem and deciding how the project might be handled. In this step, brainstorming is vital for generating as many ideas as possible. After all the proposals have been presented, a tentative design is made, to see if the entire project can be developed.

- Development: is where the tentative idea is developed and implemented, whenever the idea proves feasible. How the idea is developed depends on the type of IT artefact being proposed. This part will lead to the development of the final solution.
- Evaluation: this step involves analysing whether the objectives established at the start of the research are obtained after the artefact is developed. If this is the case, the project can be validated
- Conclusion: the results of the design process are compiled and written up, and the knowledge obtained is highlighted, along with potential study topics for future work.

In Figure 5 we can see a more detailed flowchart of the methodology applied to this specific project.

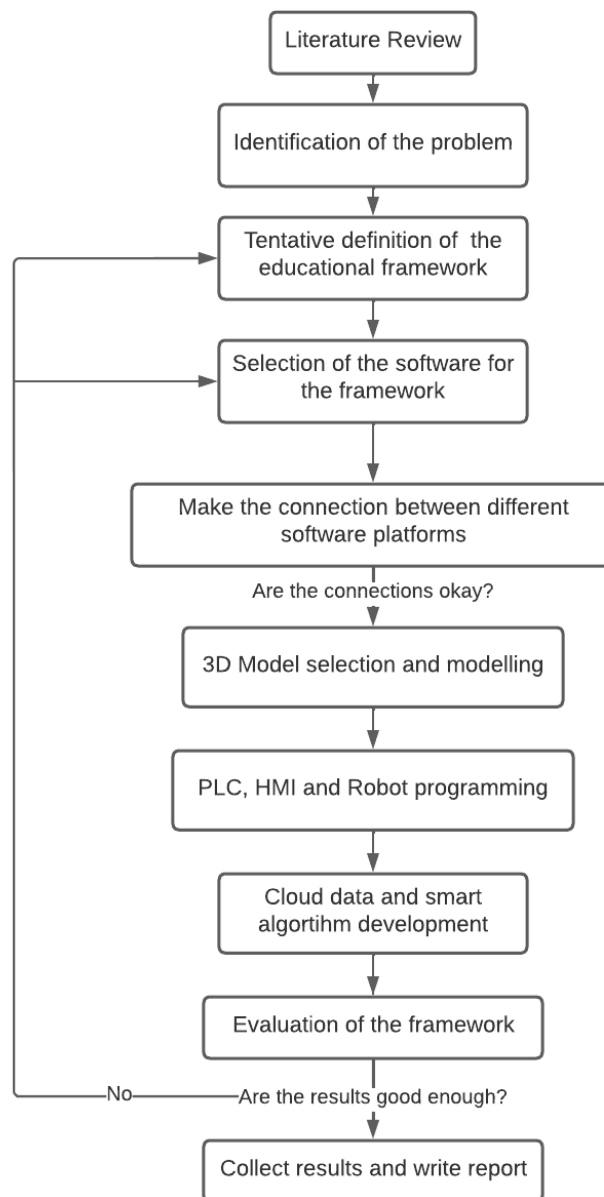


Figure 5 Flowchart of the methodology applied to the project

Awareness, in this case of study, comes due to the gaps that exist between education and the industry when it comes to the development of automated systems. To address this issue, first, a literature review has been done, being the researched areas are the application of virtual commissioning in education and the gaps between education and the industry that has been mentioned before.

Once a better understanding of the different research fields covered by this project is obtained and the problem is identified, a definition of the educational framework is suggested, that aims to cover different areas of automation and automated systems.

After a tentative definition of the framework is done, since the framework will be developed by using different software, the needed software has to be selected. This software will enable all these functionalities such as the modelling of the system, the PLC and robot programming, the HMI functionality, retrieving data via an API, and applying different kinds of smart algorithms. And on the other hand, there will be the connectivity part that enables the communication between all these functionalities.

The selection will be done using a weight table that will include the main software that are offered and following some criteria the best options will be selected. After all the needed software are selected the connection between them must be created and it is needed to check whether these connections work for virtual commissioning or not.

Then, in order to evaluate the framework, the defined framework will be used to do the actual virtual commissioning of a manufacturing process (integrating software that allows system modelling, PLC programming, Robot programming, cloud actions, and using an external programming software). It will also be checked if the framework and software selection allow the user to experience different activities that are performed when performing virtual commissioning and if the model covers all the aspects of virtual commissioning.

Once the results are good enough the conclusions and the process will be written and summarized in the report. If the results reveal anything unexpected or anomalous, it must be also documented so that it can be used in future research.

5 Implementation

5.1 Selection of the software for the framework

5.1.1 Modelling software selection

As explained in Chapter 2.1, VC is a tool that is used to simulate manufacturing systems and associated control programs on a virtual plant before the physical plant is actually commissioned. Different steps need to be taken in order to achieve VC, being the first one the development of a complete model of the production system in a 3D environment, formed of conveyors, robots...

Nowadays, there are many environments where a manufacturing process can be simulated and make it look as real as possible. Some of the most common software are KUKA, RobotStudio, and RoboDK. To choose the software that fits better for our case study, the products are compared using the following weight decision matrix (Table 1), in which the software are evaluated using different criteria. Each criterion has a weighting factor between 1...5, whereby 5 is the highest weight and 1 is the lowest.

Table 1 Weight decision matrix for modelling software selection

	Weight	KUKA Sim	RobotStudio	RoboDK
Features	5	4	4	4
Training	4	3	4	5
Customer support	2	3	5	4
Operating specifications	1	4	4	5
Pricing	5	2	5	3
Sum		52	75	68
Rank		3	1	2

As for the features offered by the three software, they are all very similar. All three offer a powerful simulation environment for robots. Ensuring collision-free paths, advertising of any singularities or joint limits... In the case of RobotStudio and KUKA Sim, they are not brand-independent software. RobotStudio is ABB's simulation and offline programming software, just as KUKA Sim is KUKA's one. This means that in RobotStudio only ABB robots are available and KUKA robots in the case of KUKA Sim. On the contrary, RoboDK is brand-independent and it supports over 30 different robot brands. However, unlike the other two, this software doesn't support the physics behavior of the objects, which is a must for some types of robot simulation.

Regarding the training that these software offer, all of them have online training on their official website. The main difference between them is the amount of content of those training services. RoboDK is the one with the more extensive catalog, offering not only documentation but also adding videos to the documentation, facilitating the understanding. Is followed by RobotStudio its online training is more focused on video tutorials just like KUKA sim but being KUKA's catalog more limited.

As regards customer support, all three provide online customer service, including a contact form, email, and a phone number. Moreover, RobotStudio and RoboDK, unlike KUKA sim, have a forum where the customer can ask specific questions or discuss any problem they are facing during a project... Although both software offer this forum, the RobotStudio one is bigger and is better structured than the RoboDK one.

Concerning the operating specifications, RoboDK, apart from supporting Windows 7 and 10 operating systems, also supports Mac OS and Linux, unlike the other two, that only support Windows 7 and 10.

As for pricing, in our case, as the university already has a RobotStudio license, the cost has been 0, which is why it is the best option. However, in case of not having the license, RoboDK offers a 2-year educational subscription for 145€, which is only for educational use. The professional version costs 2995€ but it is only a one-time payment, while KUKA sim costs 1200€/year and RobotStudio 1500€/year. Moreover, the last two offered additional PowerPacs like ArcWelding, Cutting... that have an additional cost.

Taking into account all the scores and the sum, the winning alternative is the RobotStudio software, being the price the differentiating point.

5.1.2 PLC programming software selection

Once the modelling software has been selected, the software where the PLC is programmed needs to be selected. As explained in chapter 2.3, the major duty of the PLC is to manage the signals collected from the sensors, perform some operations based on the established logic, and then send commands to the actuators. All this has been done in the selected software.

As in the case of modelling software, nowadays, there are many software that apart from programming the PLC, it is possible to program the HMI. Some of the most common software are TiaPortal, Codesys, and Studio5000. Like in the previous chapter (5.1.1), the best software is selected using a weight decision matrix (Table 2) and following the same criteria.

Table 2 Weight decision matrix for PLC programming software selection

	Weight	TiaPortal	Codesys	Studio5000
Features	5	4	5	3
Training	4	5	4	2
Customer support	2	5	5	4
Operating specifications	1	5	5	5
Pricing	5	2	5	2
Sum		65	81	46
Rank		2	1	3

Regarding the features offered by the software, all three provide similar ones. The main differences are on one hand, that Codesys, unlike the other two environments, is brand-independent, while TiaPortal is Siemens's software and Studio5000 the Rockwell Automation software. On the other hand, Studio5000 is the only one of the three that requires a separate software package for the HMI programming.

As for the training service provided, Studio5000 only offers non-free E-learning courses. Codesys also provides these types of courses, but it also offers a help site where much documentation from all Codesys areas is available and is complemented with video tutorials. In the case of TiaPortal, it also offers similar training services to those of Codesys but the content is more extent.

As far as customer support is concerned, all of them provide online customer service with contact forms, phone calls, frequently asked questions (FAQ)... Moreover, unlike Studio5000, both TiaPortal and Codesys, have a forum where users can open discussions or check already solved problems.

As regards the operating specifications, all three support Windows 10, TiaPortal also supports Windows 7, Codesys Windows 8, and Studio5000 both of them.

Concerning the price, Codesys is the only one that is totally free, since it is an open-source programming language. On the contrary, both TiaPortal and Studio5000 have non-free licenses and although they are different licenses editions the cheapest ones cost more than 500€/year.

Taking into account all the scores and the sum, the winning alternative is the Codesys software, being the price again the differentiating point.

5.1.3 Smart Algorithm / Optimization Solver Selection

Due to the intense competition caused by globalization, companies are looking for the best ways to improve their manufacturing processes and obtain higher-quality products. Moreover, products need to be manufactured at competitive prices and need to take into account things such as environmental impacts, production costs, and material consumption.

Working under optimal conditions is critical for lowering manufacturing costs, increasing productivity, and improving the quality of manufactured goods. In general, optimization is the process of picking the optimal solution among a set of viable solutions for a given problem, based on a set of criteria. The optimization problem consists of three components (Afteni & Frumușanu, 2017):

- The function to minimize or maximize.
- The variables that are part of that function.
- The restrictions or constraints.

Optimization solvers assist decision-making in the areas of resource planning, allocation, and scheduling. They include sophisticated algorithms for solving mathematical programming, constraint programming, and constraint-based scheduling problems.

First, the parameter to optimize needs to be selected. The objectives could be to optimize energy consumption, manufacturing costs, productivity, or cycle time. In this case, the objective will be to select an algorithm or an optimization solver that optimizes the makespan of a client's order, what is also known as the job shop problem.

Consequently, an optimization solver that can be implemented in PyCharm needs to be selected. There are many different open source options. Three options have been considered and the selection has been made following the decision weight matrix shown in Table 3. Each solver is given a point in the range of 0-5. And later that values is multiplied by the wight of the criteria.

Table 3 Weight decision matrix for optimization solver

	Weight	PuLP	OR tools – CP SAT	FICO - XPRESS
Ease of use	4	3	4	4
PyCharm compatibility	3	5	5	4
Available examples	4	3	4	3
Community support	3	3	4	3
Sum		48	59	49
Rank		3	1	2

Taking the matrix as a guide, that Google's constraint programming solver (CP-SAT) is the best option. The given points do not follow a grading criteria but are given according to the comparison between the three of them. All of them are open source except Xpress, but this one is free for academic use. When it comes to the ease of use, the CP-SAT solver and the FICO's solvers are the easiest ones thanks to the available extent documentation. They can all be easily implemented on Python. OR tools are quite popular, it has a great community support. Consequently, CP SAT solver has been selected.

5.2 Make the connection between different software platforms

Once the software that forms the framework has been selected, the connectivity between them needs to be tested, to see if it is possible to integrate all. Following the framework diagram (Figure 3) defined in chapter 2.5, a more specific diagram has been made (Figure 6):

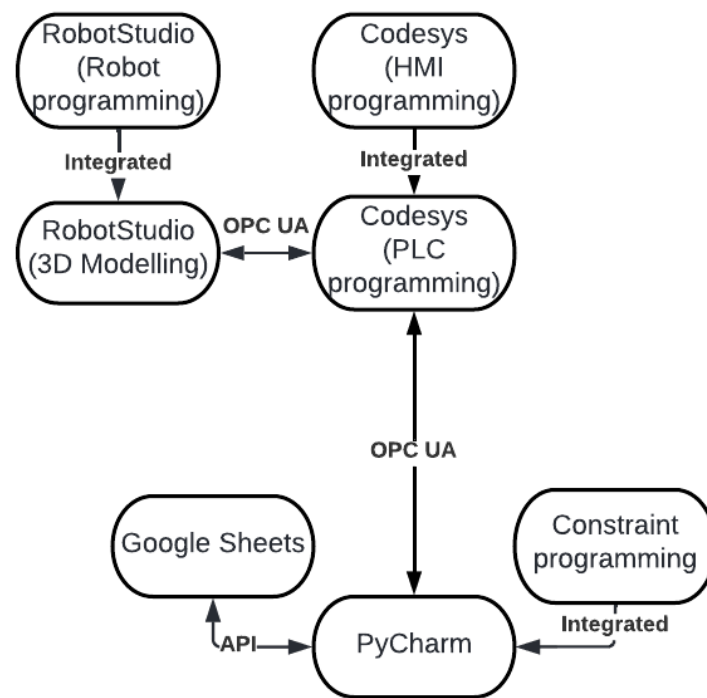


Figure 6 Specific framework diagram

5.2.1 RobotStudio-Codesys connection

The connection between Codesys and RobotStudio has been established via OPC UA (chapter 2.4.1). To establish the connection, first, it is necessary to declare the signals that are used in the process inside Codesys, in a Global Variable List (GVL) object. These signals simulate the input and output signals of a real PLC, so it is important to make sure that each signal has a different directory in the PLC's memory. Once the signals have been declared, the Symbol Configuration object must be added to the Application. When this object is added, a window pops up, where it is necessary to check the option that says "Support OPC UA features". With these symbols, you can access the variables from outside, in this case from an OPC server.

After the GVL has been created, the signals are declared and the Symbol Configuration is also created, is the time to include the declared variables in the Symbol Configuration object. For that, the option "Link always" has to be checked. This option can be found inside the configuration window of the GVL, in the Build tab. Build the project and the GVL appears in the Symbol Configuration object. Before Login the program into the PLC, the PLC must be started and the GVL checked in the Symbol Configuration tab.

Once Codesys is configured, the OPC UA server needs to be configured in RobotStudio. For that, OpcUaClient Smart Component is used. First, the Server address is defined. Since the virtual PLC is in its own computer, the address is: `opc.tcp://localhost:4840`. To establish the connection the “Secure connection” option needs to be unchecked and the “Auto connect” one doesn’t matter. After configuring the component an excel file must be created and imported to the component, selecting the file after clicking “import configuration”. The excel file must include 8 columns (Figure 7):

A	B	C	D	E	F	G	H
NamespaceUri	IdentifierType	Identifier	ReadWrite	Signal	SignalType	Controller	Device
CODESYS/3S/lecVarAccess	String	var CODESYS	Write	SOP010_FQ1	Digital		

Figure 7 Excel file example for OpcUaClient configuration

The first column needs to be named as NamespaceUri and the value always is CODESYS/3S/lecVarAccess, as long as the Codesys version is the 3 one. The second column, IdentifierType, specifies the type of the Identifier that is the next column. Since the Identifier is |var|CODESYS Control Win V3 x34.Application.GVL.DeclaredVariable, the IdentifierType is String. The .GVL. part of the identifier is the name of the GVL object created before and the .DeclaredVariable is the name of the signal, the same as declared in the GVL. The 4th column determines if the variable is an Input variable or an Output one. Write means that the signal is written in the PLC so is identified as an input and in the case of Read, it is identified as an output. Then comes the Signal column, being its value the same signal name as the one declared on the GVL. The SignalType column refers to whether the signal is Digital, Analog, or Group. Finally, the last two columns must be empty for station signals, which are non-robot signals that connect the PLC and various station components such as sensors and actuators.

Once the file is imported and the smart component is configured correctly (address...), by clicking connect it connects to the OPC UA server and RobotStudio is able to communicate with Codesys and vice versa.

5.2.2 Codesys-PyCharm connection

PyCharm and Codesys connection, as in the previous case, is formed via OPC UA. Codesys is already configured, so it is only necessary to work in the PyCharm part. For that, a publication uploaded to GitHub has been followed (Gehring, 2018). There it is explained how to establish the connection with the OPC server and how to access the variables and how to read the accessed variable or write another value.

5.2.3 PyCharm-Google Sheets connection

Finally, the connection between PyCharm and Google Sheets has been carried out following a video tutorial available on YouTube (Learn Google Spreadsheets, 2020). This connection is formed through the Google Sheets API of the Google Cloud Platform. This API allows access to a specific Google Sheet from PyCharm and once you access the sheet, it includes many functions to read/write data, change the format of the cells, clear the sheet data... Many of these functionalities are explained in the video mentioned before, but apart from the video, the API has its own documentation (Google, 2022b) where all the functions are explained in detail.

5.3 3D Model Selection and Modelling

In virtual commissioning, it is necessary to have an environment where we can simulate the manufacturing process and make it look as real as possible. For that, RobotStudio is used. RobotStudio is ABB's PC-based, 3D, virtual robot programming and simulation software. Apart from programming robots offline it also supports advanced real-time physics simulation.

There are many aspects that need to be considered when making the selection of the model. In this project, one of the most important things is to select a model where the selected smart algorithm is applicable. That is why, it has been decided to select a model where various product variants could be produced, and each product has a different sequence of operations. In this way, the smart algorithm can later optimize the best order to produce the different variants. A similar layout is used by (Barbieri, Bertuzzi, Capriotti, Ragazzini, Gutierrez, et al., 2021). In this case, they use a genetic algorithm for the optimization of the makespan.

Moreover, there are other aspects that are considered in the selection process. The selected model must also promote PLC and robot programming. Therefore, the model consists also of many conveyors and different robots, the layout is shown in Figure 8.

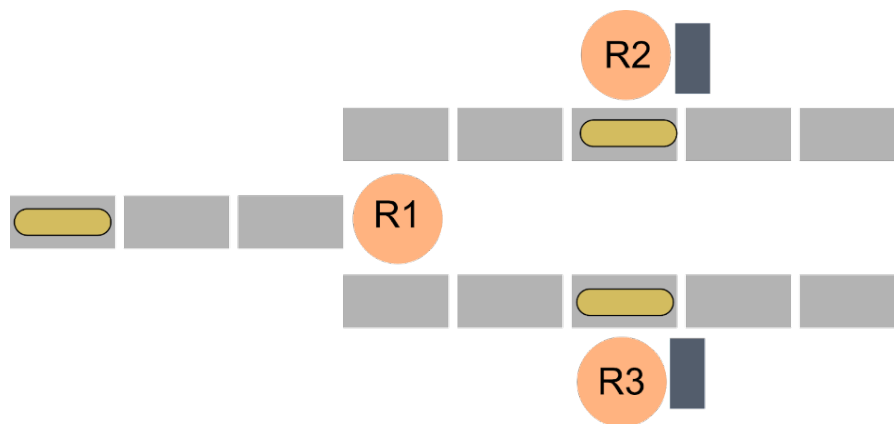


Figure 8 Layout of the plant

5.3.1 Functional Description

Taking all the mentioned things into account, the layout of the selected model is the one shown in Figure 8. The manufacturing process is mainly composed of three elements: the products, the conveyors that move the product through the system, and three robots. The process will produce skateboards and two different product variants are produced in this plant.

When working in automatic mode, first, the operator introduces the number of products that need to be produced from each variant. Then the smart algorithm computes the optimal production order, and the production starts.

Initially, both variants enter the plant through the same conveyors. Once they get to the first robot the products are sorted to the upper or lower conveyors. In the case of product variant A, the product will be moved to the upper conveyors. When it gets to robot two the robot will perform some drilling operations on the product and once it is finished it will place it back on the conveyor and the product is sent to the next operation. In the case of variant B, the board is moved to the lower conveyors and the robot here performs some edge cleaning on the product.

There are also some additional elements in the plant, safety elements such as fences, there is also the operator's panel. The goal here is to make it look as similar to the real plant as possible, the resulting virtual cell is the one shown in Figure 9 Virtual cell in RobotStudio.

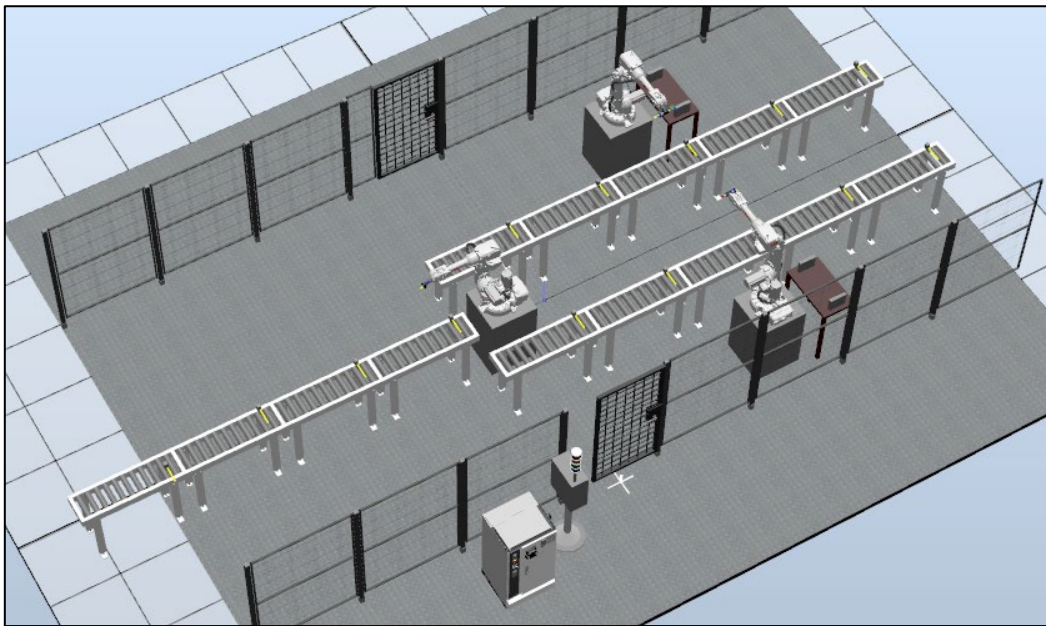


Figure 9 Virtual cell in RobotStudio

5.3.2 Conveyors and Robots

Smart Components are RobotStudio objects that have built-in features and logic. RobotStudio comes with a set of Base Smart Components that may be used for basic motion, signal logic, arithmetic, parametric modelling, sensors, and more. In the case of the conveyors, the Smart Component has been created with the use of other Base Smart Components. In this way, the product moves along the conveyor and there is also a sensor to detect the product.

Apart from that, every robot needs a tool for material handling purposes or to make any other modifications to the product. These tools can easily be made in RobotStudio or be imported into the system as libraries. Each tool has its own Tool Center Point (TCP) which determines the exact working point of the tool. The robots can be programmed directly in RobotStudio. This programming technique is called off-line programming as the robot is programmed apart from the physical cell.

5.3.3 OPC UA Connection to PLC environment

In this case, there is an external PLC that controls all the logic of the plant. That is why a connection is needed between RobotStudio and the automation software Codesys. The connection is made by using the “OpcUaClient” Smart Component.

5.4 PLC, HMI, and Robot programming

As mentioned before, the PLC is the main controller of the process. It controls most of the activities of the plant. To program the PLC, Codesys is used. In our educational framework, the computer itself works as a real PLC.

5.4.1 PLC programming approach

Different approaches can be taken when creating the PLC code for a manufacturing process. In this case, the program is divided into suboperations, each conveyor and robot are considered a suboperation. The PLC code provides the actuators of the plant with logic.

First, we have an IO list with all the signals of the process. This list is created with the help of an excel sheet. It is important to make sure that each signal has a different directory on the PLC’s memory.

The programming approach has been to use function blocks. Functions blocks are basically code pieces represented as boxes with some inputs on the left side and outputs on the right. These blocks can be reused in the same project and in new ones. Function blocks have been used for the conveyors. Some conveyors must fulfil different functions. In these cases, the function blocks have been modified to fit the goals in each case.

Regarding the language, there are different programming languages, in this case, the language called ladder logic is used in most parts of the project. It is a graphical PLC programming language that expresses logic operations with symbolic notation. The pieces of code that were not covered by function blocks have been programmed by using ladder.

5.4.2 HMI

For the HMI part, an operation panel has been created with different buttons as shown in Figure 10. As has been mentioned before, the plant can work both in manual and automatic modes. For accessing the automatic mode, the switch must be on the “AUTO” side and the control voltage must be on. The “Cycle Start” button must be pressed for three seconds to start the production on automatic mode.

And for accessing the manual mode, the switch must be on the “MAN” side and the control voltage must be on. Then each conveyor can be actioned from another panel in the HMI.



Figure 10 Operator panel HMI

5.4.3 Robot programming approach

The robot programming technique that has been employed is the one called offline programming. It is called this way because the robot is programmed in a virtual environment and not in the physical cell.

When programming a robot, firstly, it is necessary to create the paths that it will follow. This has been done in RobotStudio. In the case of the first robot, for example, the robot picks the boards from the central conveyors and places them on the upper or lower conveyors.

As for the interaction with the PLC, in this case, it has been decided that the PLC will just tell the robot which path it needs to follow through the activation of a digital signal. Taking robot one as an example again, the PLC just activates the signals "Robot1UP" or "Robot1DOWN".

Therefore, when one product arrives to the conveyor located just before robot one the code in the PLC checks the variant of the product, and one of these signals is activated, then the robot just has to follow one of the two paths that have been created previously depending on what signal was activated.

Regarding the control of the robot tools, it is the robot controller that controls them and not the PLC.

5.5 Cloud Data and Optimization solver development

5.5.1 Cloud Data

As mentioned before in chapter 2.6, Cloud storage is really useful to store production data, such as finished product numbers, cycle time, and also to store the orders of the customers. This type of data is normally used by many departments inside a company, not only the manufacturing system, therefore, cloud access is critical.

In this project, Google Sheets has been used to store the system data, since it is quite simple to read/write the data following the API mentioned in chapter 5.2.3. Moreover, the sheet can be shared with any person, and it allows access from anywhere on the planet since it can be accessed via mobile phone, as long as the device has an internet connection.

Regarding the data that is going to be retrieved and written, the sheet is formed by 3 pages, CustomerDemand, DemandSequence, and ProductionInf.

In the first one, the customer demand is written (Figure 11), the amount of VariantA and VariantB wanted, simulating an order, in PyCharm this information is read using an API function and then is sent to Codesys via OPC UA (chapter 5.2.2).

CUSTOMER ORDER	
Set the number of products to be produced for each variant.	
Type of Product	Amount
Variant A	5
Variant B	3

Figure 11 Customer order page of the Google Sheet

As for the DemandSequence page (Figure 12), here the optimal product sequence obtained from the smart algorithm program is written, using again an API function. The schedule is updated when the customer demand data changes since the schedule also change. In addition, the status of the product is also shown on this page, indicating if the product is in process or has already been produced.

PRODUCTION INFORMATION	
PROD. ORDER	STATUS
2	
1	
1	
2	
1	
1	
2	
1	

Figure 12 Production information page of the Google Sheet

OPERATIONS INFORMATION

Set times for every operation in the production process.

	Time length (s)
Conveyor	5
Robot1	10
Robot2	15
Robot3	28

5.5.2 Optimization problem

In this case, the manufacturing process produces two different product variants. The amount of each variant is different in every client order. And the production order that gives the shortest production time depends on the amounts to produce for each variant. Therefore, when an order is received from a client the algorithm needs to compute the best sequence to produce the products variant A and B in the shortest time.

The final program is very similar to the one in the example, but some code lines have been added in order to fit our problem. The program is composed by the elements shown in Figure 14

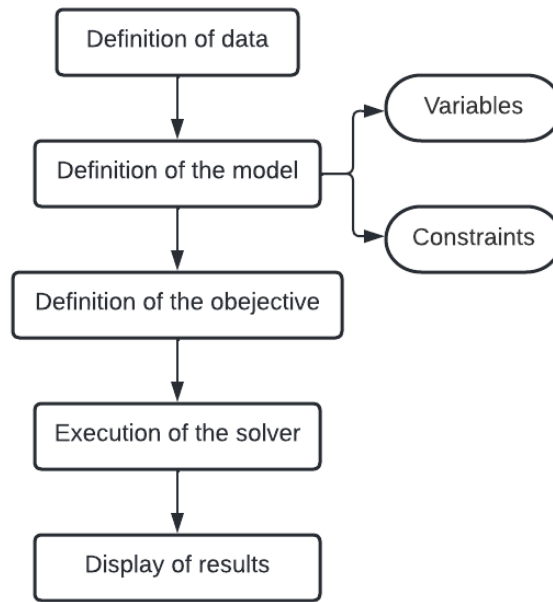


Figure 14 Main elements of the program

First, it is needed to describe each product variant in terms of “tasks”. Each task is labelled by a pair of numbers (m,p) where m is the machine number and p is the processing time of the product on that particular machine. Therefore, the layout needs to be represented in the form of machines and their production times as shown in Figure 15. In real life, these times need to be measured in the physical plant, however if the elements are modelled correctly, the times in the virtual simulation match the physical ones.

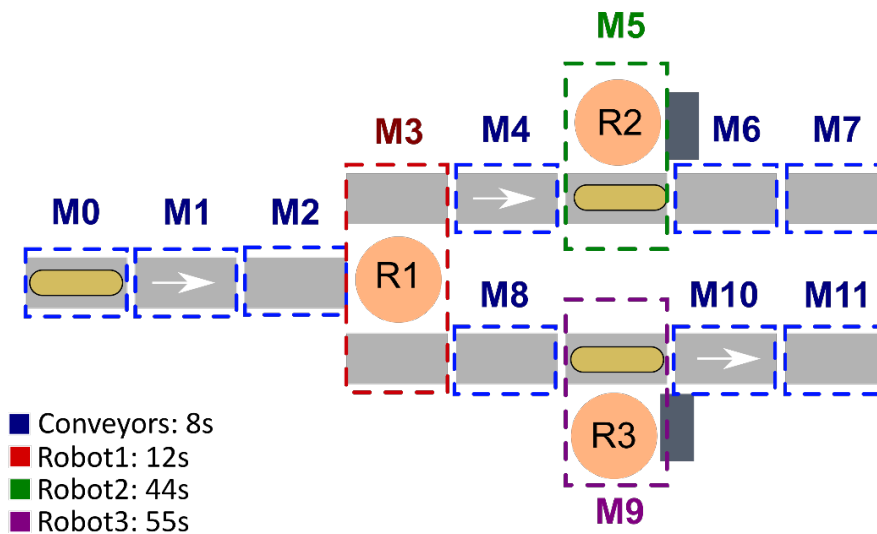


Figure 15 Machine number and production time layout

In this case, each conveyor and robot have been considered as machines and the product spends a different amount of time in each of them, 8s in the conveyors, 12s in the R1 (Figure 15), 44s in R2, and 55s in R3. Depending on the product variant, A or B, as mentioned in section 5.3.1, the product follows a different path, being processed by different machines. The list for each variant is defined as follows, indicating in each parenthesis, the machine number and the time the product spends in the machine:

- VariantA = [(0, 8), (1, 8), (2, 8), (3, 12), (4, 8), (5, 44), (6, 8), (7, 8)]
- VariantB = [(0, 8), (1, 8), (2, 8), (3, 12), (8, 8), (9, 55), (10, 8), (11, 8)]

Following the chart in Figure 14, first some data needs to be defined. A list of lists called “jobs_data” is defined. This list is created according to the client order. It includes the list VariantA as many times as the number of A type variants that are to produce and the same for VariantB.

Next, the model is declared. With the information obtained from the “jobs_data” some variables are created. In the example program some constraints come predefined:

- No task for a job can be started until the previous task for that job is completed.
- A machine can only work on one task at a time.
- A task, once started, must run to completion.

However, in the process described in Figure 15, there is one additional constraint that must be fulfilled:

- The first product entering the machine 0 must be the first one to enter the machines M1, M2 and M3.

This constraint has been added by making the start of the time of the conveyors match the end time of the previous one. This constraint is only applied to the machines 0-3.

Regarding the definition of the objective, here it is defined that the makespan is what needs to be optimized. And finally, the solver is executed and the results are displayed. The result of the example has been adjusted so the modified result is a list that consists of a sequence of the numbers 1 and 2, and this list is sent to Codesys via OPC UA.

6 Analysis and Results

Once the development of the educational framework is finished is time to evaluate whether it serves as a tool for virtual commissioning. For doing that the framework is evaluated from two different points.

6.1 Evaluation of the software integration

To evaluate the software integration, a matrix (Table 4) has been created with all data types needed to send between the different software in order to perform virtual commissioning. In this case, it has only been necessary to send Boolean (B in the table) and Integer (I in the table) type data since the majority of the data are sensor and actuator signals which are all Boolean-type, and the customer demand, production information... are sent as Integers. Although in this case, it has not been necessary to send any String-type data, if it is needed, it is possible to send this type of data.

Table 4 Data type sent between software

	Codesys	RobotStudio	PyCharm	Google Sheet
Codesys		I, B	I, B	Via PyCharm
RobotStudio	I, B		Via Codesys	Via Codeysy & PyCharm
PyCharm	I, B	Via Codesys		I, B
Google Sheet	Via PyCharm	Via Codesys & PyCharm	I, B	

After testing all the connections, the conclusion is that all software can be integrated with no problem. RobotStudio-Codesys, Codesys-PyCharm, and PyCharm-Google Sheet communicate directly, while RobotStudio communicates with PyCharm via Codesys, meaning that the variable is first sent to Codesys and is from Codesys that PyCharm reads/writes the variable. This is also the case with the Google Sheet-Codesys and Google Sheet-RobotStudio communications, that they have one or two software as an intermediary, allowing the integration of all software.

6.2 Evaluation of the educational framework in terms of virtual commissioning

Once the integration of the framework is done, it is time to evaluate whether the framework serves the user as a tool to perform different activities that are performed when performing virtual commissioning.

To evaluate this aspect of the framework, an actual virtual commissioning of a manufacturing plant has been carried out as shown in the implementation chapter 5. The difficulty here is that there is no single standardized way of doing virtual commissioning.

However, certain things are always included. Table 5 shows some of the steps that are involved in virtual commissioning. These steps have been gathered from different sources. On the one hand, having a look at the literature, main activities in VC have been identified (HJVermaak, 2015; Ugarte et al., 2022). Also, the authors have previously gone through a course (Industrial Control Technology) where virtual commissioning was involved. The other activities on the list were collected from this experience and by talking to an expert in the subject. On the right side of the table, there is a column that says whether the educational framework implemented in this project covers those activities.

Table 5 Activities involved in virtual commissioning

Activities involved in virtual commissioning	
Framework selection¹	✓
This has been the first step that has been carried out. It is not common step in virtual commissioning as most of the time the user performs the VC with the available software already in use in the company or school.	
Software selection¹	✓
The software that form the framework is selected using the weight decision matrix. ¹	
Layout prototyping⁴	✓
Initial model layout has been design to define the elements (conveyors, robots...) that are part of the model and define their distribution.	
System design^{2, 3}	✓
The layout prototype has been design and modelled in RobotStudio, using not only its own library equipments, but also imported ones.	

¹ Activity based on an expert on the field

² (Ugarte et al., 2022)

³ (HJVermaak, 2015)

Product modelling^{2, 3, 4}	✓
The product has been modelled in RoboStudio. However, it is also possible to import product models built in other environments to RobotStudio.	
Component modelling and testing⁴⁴	✓
Every component needs to be modelled and be given some properties to make it function as it would in the real world. This has been done in RobotStudio.	
I/O list handling⁴	✓
In the framework, there are many different softwares and even if the same signals all are found in different environments they are all declared in a different way in every case. In VC is common to have an excel sheet that handles this.	
PLC programming⁴	✓
Codesys is a robust environment for PLC programming. Most of the programming has been done by using function blocks.	
Robot programming⁴	✓
Robot programming has been done in RobotStudio. Online programming was used, using the virtual model for the creation of the paths.	
HMI design and programming⁴	✓
Codesys is the software which enables the creation of different HMI panels and the programming behind them.	
Optimization solver utilization	✓
Thanks to PyCharm the model is supported with external computation capabilities. This has enabled to deploy an optimization solver and solve a job shop problem, computing the best production sequence for a client's order.	
Virtual FAT testing⁴	✓
A FAT test ensures that the components and controls are working properly. In this case, since the project is developed entirely virtually, this test has been taken virtually.	

Apart from the table, some activities are not implemented with this framework. For example, the programmed PLC and robot codes are not tested in any physical equipment. This is also a crucial step in the commissioning of an automated system. But, as mentioned in the delimitations chapter, no physical equipment is needed for the defined framework apart from a PC.

⁴ Activity experienced in the Industrial Control Technology course.

On the whole, the educational framework presented in this paper allows the user to perform all the activities mentioned in Table 5. Therefore, it is possible to say that if the steps mentioned above are followed the user would get an experience of VC that is largely similar to the real one.

7 Discussion

The main goal of defining and evaluating an educational framework has been achieved successfully, enabling students to develop skills related to virtual commissioning. Nevertheless, this chapter contains discussions about the methodology followed to develop the project and about the limitations the authors faced when evaluating the defined framework. In addition, a Future Work section is also displayed, where some improvements are discussed in order to solve the mentioned limitations.

Regarding the used methodology to conduct the research, although there were alternative methods, such as the case study approach, the chosen one is an iterative process that allows backtracking when necessary, allowing to perform this project on good terms. Moreover, Design and Creation method has also helped to carry on a structured way of working.

As for the evaluation of the framework, it is important to mention that the framework has only been used to conduct the virtual commissioning of a single manufacturing process, so it is not possible to say that this framework will work for doing the virtual conditioning of any production process. However, this problem has been addressed, as mentioned earlier, by selecting a thorough plant with different elements and the ones that are highly common in the industry.

7.1 Future Work

Some improvements on the current project and interesting topics for future research are presented in the coming paragraphs. The first improvement comes for the evaluation part of the defined educational framework. In this case, the evaluation in terms of virtual commissioning has been done by using the framework and doing an actual virtual commissioning of a manufacturing process. However, it would be nice to consider doing the same thing but in a student education situation. The strengths and weaknesses of the framework could later be collected via a survey. Moreover, in this way the framework would be used for conducting the virtual commissioning of many different automated systems, addressing in this way the problem mentioned earlier.

As the objective is to give the student a close-to-real experience, it would be interesting to implement virtual reality on the virtual cell. Consequently, students could get even a closer experience of what is done in real life. Moreover, as mentioned above this project has done an initial step in introducing IT solutions to “conventional” virtual commissioning. These features could be developed more in the future and give them greater weight in future projects.

8 Conclusions

In conclusion, the project presented in this thesis has accomplished the primary goal stated in Section 1.2, defining and evaluating an educational framework for virtual commissioning that enables a close-to-real student experience. Moreover, as it can be seen in the following list, the results are in line with the initial objectives (chapter 1.1 and chapter 1.2), which have been met:

- An educational framework has been defined that enables students to develop the following skills related to virtual commissioning:
 - Building/modelling automated systems.
 - Controlling components and model behaviour using real PLC code and Human-machine interface (HMI).
 - Controlling robots in the model using real Robot code.
 - Connecting the model to an external programming environment.
- The needed software for the framework has been analysed and selected, using the weight decision matrix as a comparison tool, being the result the final implementation design shown in Figure 6.
- The framework has been set up and tested. Connections between software have been tested sending different data types used in VC among all the software.
- The framework has been analysed and evaluated. An actual virtual commissioning of a manufacturing plant has been carried out in order to evaluate whether the framework serves the user as a tool to perform different activities that are performed when performing virtual commissioning

Regarding the results obtained (chapter 6), it can be concluded on one hand, that the integration of the software has been carried out successfully and that the defined framework is valid in terms of connectivity (chapter 6.1). And on the other hand, considering the results obtained in chapter 6.2, the results show that the framework serves for VC purposes, confirming that the project has been developed successfully as the aim and objectives have been achieved as mentioned above.

This project has defined and evaluated an educational framework and has been an initial step in introducing IT solutions, such as cloud data retrieval and smart algorithm implementations, to virtual commissioning.

References

- Afteni, C., & Frumușanu, G. (2017). A Review on Optimization of Manufacturing Process Performance. *International Journal of Modeling and Optimization*, 7(3), 139–144. <https://doi.org/10.7763/ijmo.2017.v7.573>
- Barbieri, G., Bertuzzi, A., Capriotti, A., Ragazzini, L., Lorenzo, David Gutierrez, D., Negri, E., & Fumagalli, L. (2021). PRODUCTION MANAGEMENT A virtual commissioning based methodology to integrate digital twins into manufacturing systems. *Production Engineering*, 15, 397–412. <https://doi.org/10.1007/s11740-021-01037-3>
- Barbieri, G., Bertuzzi, A., Capriotti, A., Ragazzini, L., Gutierrez, D., Negri, E., & Fumagalli, L. (2021). A virtual commissioning based methodology to integrate digital twins into manufacturing systems. *Production Engineering*, 15(3–4). <https://doi.org/10.1007/s11740-021-01037-3>
- Benešová, A., & Tupa, J. (2017). Requirements for Education and Qualification of People in Industry 4.0. *Procedia Manufacturing*, 11. <https://doi.org/10.1016/j.promfg.2017.07.366>
- Bolton, W. (2004). PLC systems. In *Instrumentation and Control Systems* (pp. 156–178). Elsevier. <https://doi.org/10.1016/b978-075066432-5/50007-3>
- Brazina, J., Vetiska, J., Stanek, V., Bradac, F., & Holub, M. (2020). Virtual commissioning as part of the educational process. *Proceedings of the 2020 19th International Conference on Mechatronics - Mechatronika, ME 2020*. <https://doi.org/10.1109/ME49197.2020.9286613>
- Commission on Environment, W. (1987). Report of the World Commission on Environment and Development: Our Common Future Towards Sustainable Development 2. Part II. Common Challenges Population and Human Resources 4. *Oxford University Press*.
- Fernández, I. A., Eguía, M. A., & Echeverría, L. E. (2019). Virtual commissioning of a robotic cell: An educational case study. *IEEE International Conference on Emerging Technologies and Factory Automation, ETFA, 2019-September*. <https://doi.org/10.1109/ETFA.2019.8869373>
- Gehring, M. (2018, February 28). *OPCUA-Client-Simple/opcua_client_simple.py at master · Matthias-Gehring/OPCUA-Client-Simple · GitHub*. GitHub. https://github.com/Matthias-Gehring/OPCUA-Client-Simple/blob/master/opcua_client_simple.py
- Gehrke, L., Kühn, A., Rule, D., Moore, P., Bellman, C., Siemes, S., Dawood, D., Singh, L., Kulik, J., & Standley, M. (2015). Industry 4.0 - A Discussion of Qualifications and Skills in the Factory of the Future: A German and American Perspective. In *VDI*.
- Google. (2021). *The Job Shop Problem | OR-Tools | Google Developers*. Google Developers. https://developers.google.com/optimization/scheduling/job_shop
- Google. (2022a). *Python, Java, C#, JavaScript - Google Trends*. https://trends.google.com/trends/explore?date=all&q=%2Fm%2F05z1_%2Fm%2F07sbkfb,%2Fm%2F07657k,%2Fm%2F02p97

- Google. (2022b). *Sheets API* / Google Developers. Google Developers. <https://developers.google.com/sheets/api>
- HJVermaak. (2015). *Validating a reconfigurable assembly system utilizing virtual commissioning; Validating a reconfigurable assembly system utilizing virtual commissioning.* <https://doi.org/10.1109/RoboMech.2015.7359532>
- Hoffmann, P., Schumann, R., Maksoud, T. M. A., & Premier, G. C. (2010). Virtual commissioning of manufacturing systems a review and new approaches for simplification. *Proceedings - 24th European Conference on Modelling and Simulation, ECMS 2010.* <https://doi.org/10.7148/2010-0175-0181>
- IRJET Journal. (2017). (PDF) *Python – The Fastest Growing Programming Language* | IRJET Journal - Academia.edu. https://www.academia.edu/35592490/Python_The_Fastest_Growing_Programming_Language
- Learn Google Spreadsheets. (2020). *Google Sheets - Python API, Read & Write Data - YouTube.* Youtube. <https://www.youtube.com/watch?v=4ssigWmExak&t=1484s>
- Lin, Z., & Pearson, S. (2018). *An inside look at industrial Ethernet communication protocols Strategic Marketing Manager Texas Instruments Strategic Marketing Manager Texas Instruments.*
- Makris, S., Michalos, G., & Chryssolouris, G. (2012). Virtual commissioning of an assembly cell with cooperating robots. *Advances in Decision Sciences, 2012.* <https://doi.org/10.1155/2012/428060>
- Mortensen, S. T., & Madsen, O. (2018). A Virtual Commissioning Learning Platform. *Procedia Manufacturing, 23.* <https://doi.org/10.1016/j.promfg.2018.03.167>
- Niu, C., Xie, Z., & Zhou, T. (2021). *Application of Ethernet/IP Protocol in Equipment Lifecycle Management; Application of Ethernet/IP Protocol in Equipment Lifecycle Management.* <https://doi.org/10.1109/ICICSE52190.2021.9404137>
- Oates, B. J. (2006). Researching Information Systems and Computing. *Inorganic Chemistry, 37.*
- Reinhart, G., & Wünsch, G. (2007). Economic application of virtual commissioning to mechatronic production systems. *Production Engineering, 1(4).* <https://doi.org/10.1007/s11740-007-0066-0>
- Ren, H., Liu, Y., & Wang, H. (2019). *Research on Communication Method of OPC UA Client Based on ARM; Research on Communication Method of OPC UA Client Based on ARM.*
- Salazar, M. A., & Río, D. (2019). *EMULATION OF INDUSTRIAL FIELDBUS MODULES FOR VIRTUAL COMMISSIONING.*
- Sauter, T., & Member, S. (2010). The Three Generations of Field-Level Networks—Evolution and Compatibility Issues. *IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS, 57(11), 3585.* <https://doi.org/10.1109/TIE.2010.2062473>
- Ugarte, M., Etxeberria, L., Unamuno, G., Bellanco, J. L., & Ugalde, E. (2022). Implementation of Digital Twin-based Virtual Commissioning in Machine Tool Manufacturing. *Procedia Computer Science, 200, 527–536.* <https://doi.org/10.1016/J.PROCS.2022.01.250>

- Vermaak, H., & Niemann, J. (2015). *Validating a reconfigurable assembly system utilizing virtual commissioning; Validating a reconfigurable assembly system utilizing virtual commissioning*. <https://doi.org/10.1109/RoboMech.2015.7359532>
- Wöhlke, G., & Schiller, E. (2005). Digital Planning Validation in automotive industry. *Computers in Industry*, 56(4). <https://doi.org/10.1016/j.compind.2005.01.010>
- Zhang, J., Ding, G., Zou, Y., Qin, S., & Fu, J. (2019). Review of job shop scheduling research and its new perspectives under Industry 4.0. *Journal of Intelligent Manufacturing*, 30(4), 1809–1830. <https://doi.org/10.1007/S10845-017-1350-2>

Appendices

Appendix 1 Work Breakdown and Time Plan

The time plan defined at the start can be seen in the following Gantt chart (Figure 16). The workflow has been divided into 8 tasks and the deadlines are those with a yellow back colour.

Week	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
Specification draft and Signature				Submission of the specification																
Theoretical Framework																				
Develop of the model and program it																				
Develop the communication																				
Work on the API, cloud storage and smart algorithm																				
Validation																				
Report							Midterm report										Peer review			Final submission and public exhibition
Oral presentation								Midterm presentation										Presentation		

Figure 16 Initial time plan

The time plan below is the one that shows how the time has been distributed at the end, differing from the first plan. The work before the midterm presentation has been carried out as planned. However, after that, developing and programming the model and developing the communication has taken longer than expected. But, since working on the API, cloud storage, smart algorithm, and the validation process have taken much less time than planned, the project has been finished on time.

Week	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
Specification draft and Signature				Submission of the specification																
Theoretical Framework																				
Develop of the model and program it																				
Develop the communication																				
Work on the API, cloud storage and smart algorithm																				
Validation																				
Report							Midterm report										Peer review			Final submission and public exhibition
Oral presentation								Midterm presentation										Presentation		

Figure 17 Final time plan

Appendix 2 Software versions

The following table (Table 6) shows the software used during this project, and the version of each one. Is important to sepecify the versions used, since in case of Codesys for example, the newest version had some problems when communication to RobotStudio via OPC UA, that is why an older version has been used.

Table 6 Used software versions

SOFTWARE NAME	VERSION
RobotStudio	RobotStudio 2021.4 (64-bit)
Codesys	CODESYS V3.5 SP16 Patch4 (64-bit)
Pycharm	PyCharm 2021.3.2 (Community Edition)