

COLLABORATIVE ROBOTS INTEGRATION AND OPERATION IN ICAI'S MINI-FACTORY

M. B. Lettieri Palencia

ICAI – Universidad Pontificia Comillas

ABSTRACT This Master's Thesis arises from the need to provide the student with a completely virtual and remote tool to design and program a factory from the point of view of industrial automation. In addition, the developed platform will also be used to complement the learning and improve the understanding of the practices in the laboratory.

For this purpose, a simulation of the complete plant of the ICAI Mini-factory was performed in the offline programming and simulation tool for robotic applications, RobotStudio [1]. The system has four collaborative robots; GoFa CRB 15000 [2] and SWIFTI CRB 1100 [3]. The user will have the ability to modify the modules to define the operation of the reprogrammable automatons, as well as the set of actuators and sensors that make up the laboratory. In addition, thanks to the systematic and adaptable programming that has been developed, this project can be scaled to any other environment.

KEY WORDS Collaborative Robots, *GoFa CRB 15000*, *SWIFTI CRB 1100*, *RobotStudio* and Simulation.

I. INTRODUCTION

Today, the adoption of industrial robots and digitalization has transformed various industrial sectors, optimizing process management and reorganizing workplaces. These robots seek to increase efficiency, precision and productivity, while protecting workers in harmful environments. However, they face challenges such as high costs and security and privacy concerns.

At Universidad Pontificia Comillas, the ICAI Mini-Factory has incorporated ABB's advanced Cobots [4], giving students the opportunity to work with these machines and develop skills in the field of Industrial Automation. The project focuses on didactically integrating these Cobots in the university, deepening their systems and applications in order to achieve precise control through the RobotStudio tool.

This project focuses on three main objectives. The first one aims to improve the physical integration of the new Cobots in the laboratory of Universidad Pontificia Comillas. Several designs and tests of the grippers for the GoFa CRB 15000 Cobot were performed with the Solid Edge computer-aided design software and printed with a 3D printer.

The second objective focuses on performing a complete simulation of the Mini-factory using ABB's offline simulation and programming software, RobotStudio. The accuracy and productivity of the lab was taken into account,

performing measurements of the actual stations and analyzing the sensors and actuators for effective operation. The third objective is related to the study of protocols and safety levels of each robot. Simulated safety measures were implemented in the plant to ensure the correct operation of the project.

All objectives were developed following an iterative methodology, involving research, design, implementation, programming and testing processes to move towards the next objective. The objectives defining the project are attached in *Figure 1*.

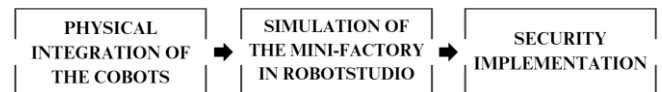


FIGURE 1: PROJECT OBJECTIVES

II. STATE OF THE ART

Industrial robotization, although considered a modern invention, was originated in the United States in the 1950s at the General Motors factory in New Jersey [5]. The first industrial robot, called Unimate 001 [6], shown in *Figure 2*, was designed to place and stack metal castings. Over time, the technology evolved quickly and more robots were implemented in the plant.



FIGURE 2: UNIMATE 001, THE FIRST INDUSTRIAL ROBOT

In the last few decades, collaborative robots or Cobots [7], invented in 1996 by J. Edward Colgate and Michael Peshkin, have emerged [8]. These Cobots enabled greater safety by working in close proximity to employees and easier, more intuitive programming. They were also more affordable and flexible, opening up a new market for small and medium-sized manufacturers.

In 2008, the first Cobot was installed at the facilities of Linatex [9], a plastics and rubber supplier. Universal Robots' Cobot UR5 marked the beginning of a growing adoption of Cobots in various industries around the world.

In addition to industrial applications, robots have found their way into sectors such as education, food and medicine due to their hygiene and process flexibility.

III. DESCRIPTION OF THE TECHNOLOGIES

First, all the specific characteristics of the Cobots used in this project will be developed below:

- **Load Capacity and Range:** Cobots, including the ABB GoFa CRB 15000 and the SWIFTI CRB 1100, generally have a lower payload capacity compared to traditional industrial robots. They can handle lighter loads, typically up to several kilograms, and their reach is also usually shorter, making them ideal for tasks closer to the human operator.
- **Vision System, Advanced Sensing and Safety:** Cobots are equipped with a vision system and advanced sensors that allow them to interact safely with their environment. Thanks to the application of systems such as artificial vision, they can detect nearby objects and people, which facilitates their collaboration with human workers without risking their safety.
- **Easy and Intuitive Programming:** Cobots are designed to be programmed easily and quickly, which facilitates their implementation and improves operational efficiency in different applications. They often feature user-friendly interfaces that allow operators to teach tasks intuitively.
- **Compact and Ergonomic Design:** Its compact and ergonomic design allows the robot to take up less space and is easy to integrate into different work

environments, providing greater flexibility in work area layout.

- **Versatile Applications:** These collaborative robots are suitable for a wide range of industrial applications, from assembly and material handling tasks to inspection and quality control.
- **Human-Machine Collaboration:** The main distinctive feature of Cobots is their ability to collaborate directly with human workers in the same workspace. This allows for greater flexibility in production and improves efficiency in certain tasks.

The programs used in the project are presented as follows.

First, Solid Edge [10], a 3D computer-aided design (CAD) software, was used to design the GoFa CRB 15000 robot gripper and the loaders for Stations 2 and 3. Solid Edge offers a set of tools for solid modeling, assemblies and 2D orthographic views, as well as linking to Product Lifecycle Management (PLM) technologies. Although it is not free, the student obtained the license through Comillas Pontifical University. The software is exclusively for Windows operating systems.

For the printing of the GoFa CRB 15000 grippers, Ultimaker Cura [11], a free application designed for 3D printers, was used. This 3D printing software is versatile and widely compatible with different printers. With an intuitive interface, it allows adjusting settings to optimize print quality. The version used was UltiMaker Cura 5.4, and it was considered essential to improve the physical integration of the robots in the ICAI Mini-factory.

The most relevant software of the project was RobotStudio [12], an offline programming and simulation tool for robotic applications from ABB. It allows robots to be programmed without interrupting actual production, and its similarity to the actual robot controller enables accurate and realistic simulations. Although it is not free, licenses were obtained through the University. RobotStudio proved crucial in minimizing risks and costs associated with errors, as it allows simulating the robot's behavior before implementing changes in the real production environment.

In general, these programs have played fundamental roles in the design, printing and simulation of the robots and components of the ICAI Mini-factory.

IV. PROJECT CONFIGURATION AND DEVELOPMENT IN ROBOTSTUDIO

As discussed in previous sections, the first task was to design a more suitable gripper for the GoFa CRB 15000. The main problem lay in the ability of the old gripper to hold the lego pieces in the laboratory. This caused problems in the numerous work sessions and research projects that were carried out with this Cobot, so the first objective of

this Master's Thesis was to develop the physical integration of the reprogrammable manipulators. Below, the design of the gripper and its implementation in the laboratory are shown in *Figure 3* and *Figure 4*, respectively.

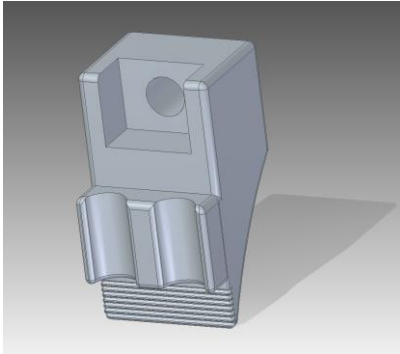


FIGURE 3: SOLID EDGE DIAGONAL VIEW OF THE GoFA CRB 15000 GRIPPER DESIGN

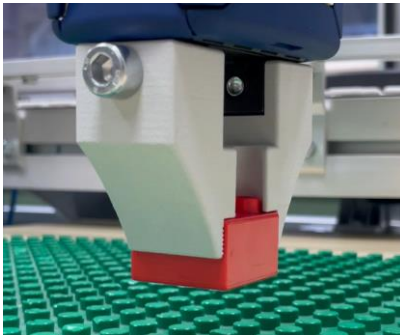


FIGURE 4: GoFA CRB 15000 REAL GRIPPERS WITH WORKING BASE

Next, the 3D design of the chargers for Stations 2 and 3 of the plant, which were recently introduced in the university laboratory and were not previously implemented, will be discussed. To achieve an accurate design, several measurements were taken considering the real component of the project. The biggest challenge was to integrate the Solid Edge designs into the robotic application software, RobotStudio. After exactly replicating the loaders in a three-dimensional environment, they were implemented into the complete plant model in the appropriate positions and orientations. The charger design is shown in *Figure 5* and the factory simulated charger is presented in *Figure 6*.

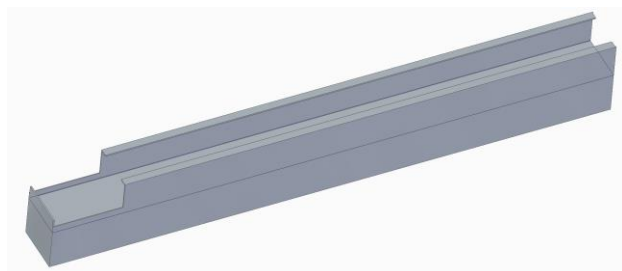


FIGURE 5: DIAGONAL VIEW OF CHARGER DESIGN

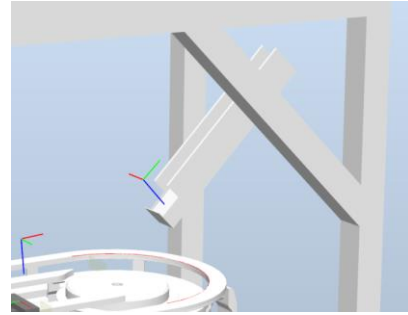


FIGURE 6: MINI-FACTORY CHARGER IN ROBOTSTUDIO

So, in general terms, the model is composed of a general structure, the Cobots, the robot stations and other components, such as elevators, loaders, lego pieces, pallets...

A detailed explanation will be given of the successive programming of each of the stations that make up the laboratory. It should be noted that, first of all, all the research, design and implementation of one of them was carried out, in order to scale the model to the rest of the cells. For this reason, it can be seen how the development of Station 1 is much more extensive than the rest, since all the programmed codes and the implemented logic are detailed in more depth.

Although the complete code of Station 1 is attached in the *Appendix A*, the main functions that compose the whole programming of the entire laboratory will be explained below:

- **main()**: Main function that coordinates the flow of the program, from the simulation setup to the operation at the station.
- **setup()**: Initialization of actuators in the stations, such as elevators, retainers and rail change arms.
- **esperaPale()**: Function to detect and wait for the arrival of a pallet in the work area, checking if it has assembled lego parts.
- **montaFigura()**: Defines the positions to which the robot must take the pieces to assemble a figure.
- **desmontaFigura()**: Defines the positions to which the robot must take the pieces to disassemble a figure.
- **bajaPale()**: Configures stations with elevators and manages the movement of pallets.
- **liberaPale()**: Manages the main and secondary conveyor to allow pallet flow.
- **colocaLego()**: Defines the robot movement to transport a lego piece to its final position, considering specific transformations and orientations.
- **cogeLego()**: Similar to *colocaLego()*, but designed to disassemble a part and leave it in its storage location.

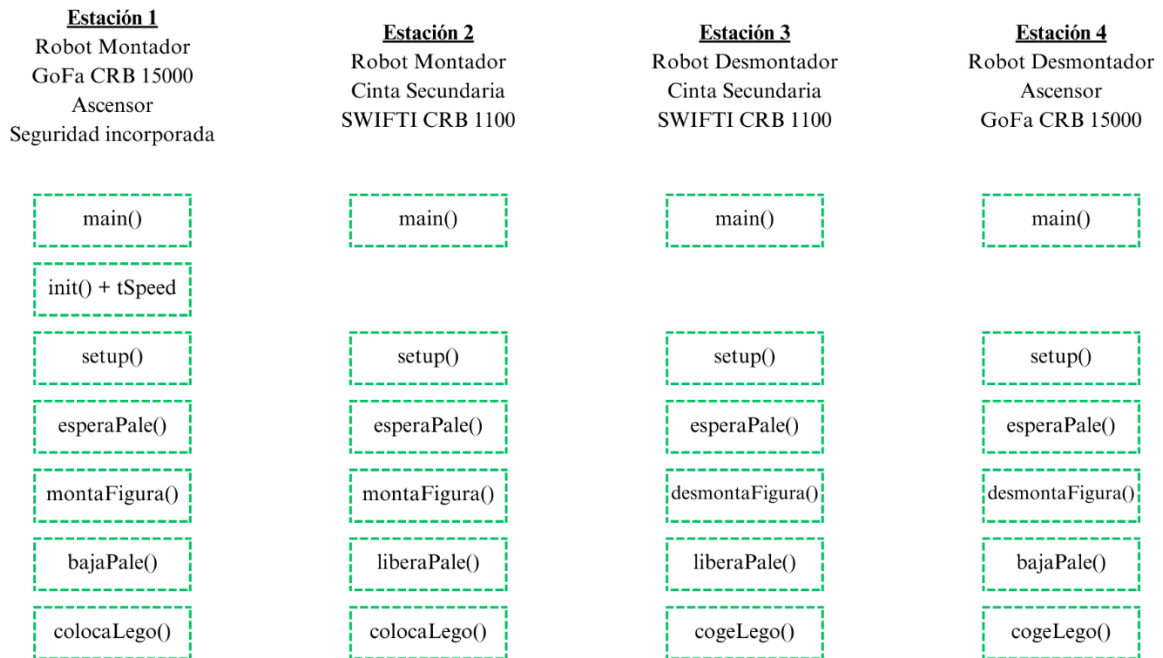


FIGURE 7. ROBOT PROGRAMMING DIAGRAM

In conclusion, we want to emphasize that all programming has been done to have a large component of scalability. The most significant purpose is to provide the student with a simple and effective tool to be able to reprogram any of the robots in the laboratory. For this reason, all stations have a homogeneous and systematic code hierarchy, which is represented in *Figure 7*.

Block programming is crucial to comply with the sequential operation of the Mini-factory between the robots and the plant as a whole. For this reason, in this point the logic that had to be built to design the operation of the actuators, the sensors, the creation of the lego pieces and their corresponding interactions with the automatons will be explained.

Although, the logic of the general station is attached in *Appendix B*, the following reflections are required.

The introduction of the robots and their controllers into the station logic was crucial for the laboratory workflow. Specific blocks were used for each robot, such as "CRB15000_5_95", "SWIFTI_2", "SWIFTI_S3" and "GOFAS4", and the necessary inputs and outputs were configured for the correct operation of the signals.

The "General" block turned out to be the most voluminous, as it managed all the actuators, sensors and pallet movement in the factory. Inherited from a previous project, it was adapted to include three new robots and replace the IRB 120 robot with a GoFa CRB 15000, which

considerably increased the number of inputs and outputs in the logic.

A major update in the "General" block focused on the creation of sensors to detect lego parts on pallets (*Figure 8*), essential for organizing the flow between stations with assembler or disassembler robots. This real simulation of industrial movements in the laboratory required the accurate detection of pieces for the correct operation of the robots and the coordination of pallets between stations.

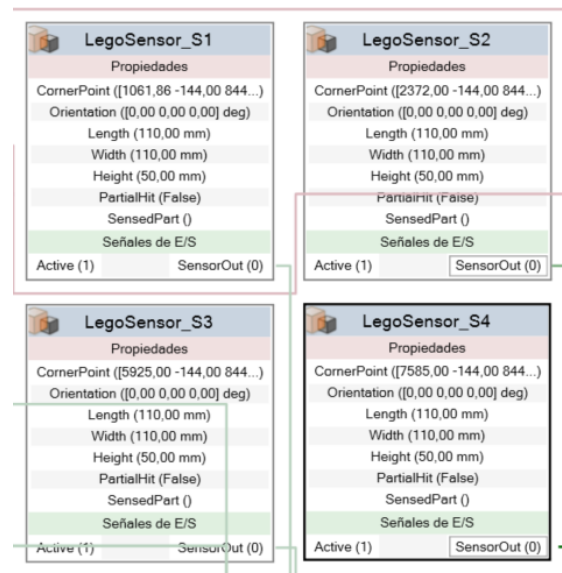


FIGURE 8: SENSOR BLOCKS FOR LEGO PIECES

Now, reference is made to the "Lego Pieces" blocks. The main function of these blocks is to manage the creation, movement and removal of the lego pieces. It was also necessary to program in these blocks the logic that ensures that the lego pieces will follow a movement analogous to that of the pallets or the robot grippers at the relevant moments. This logic is shown in *Figure 9*.

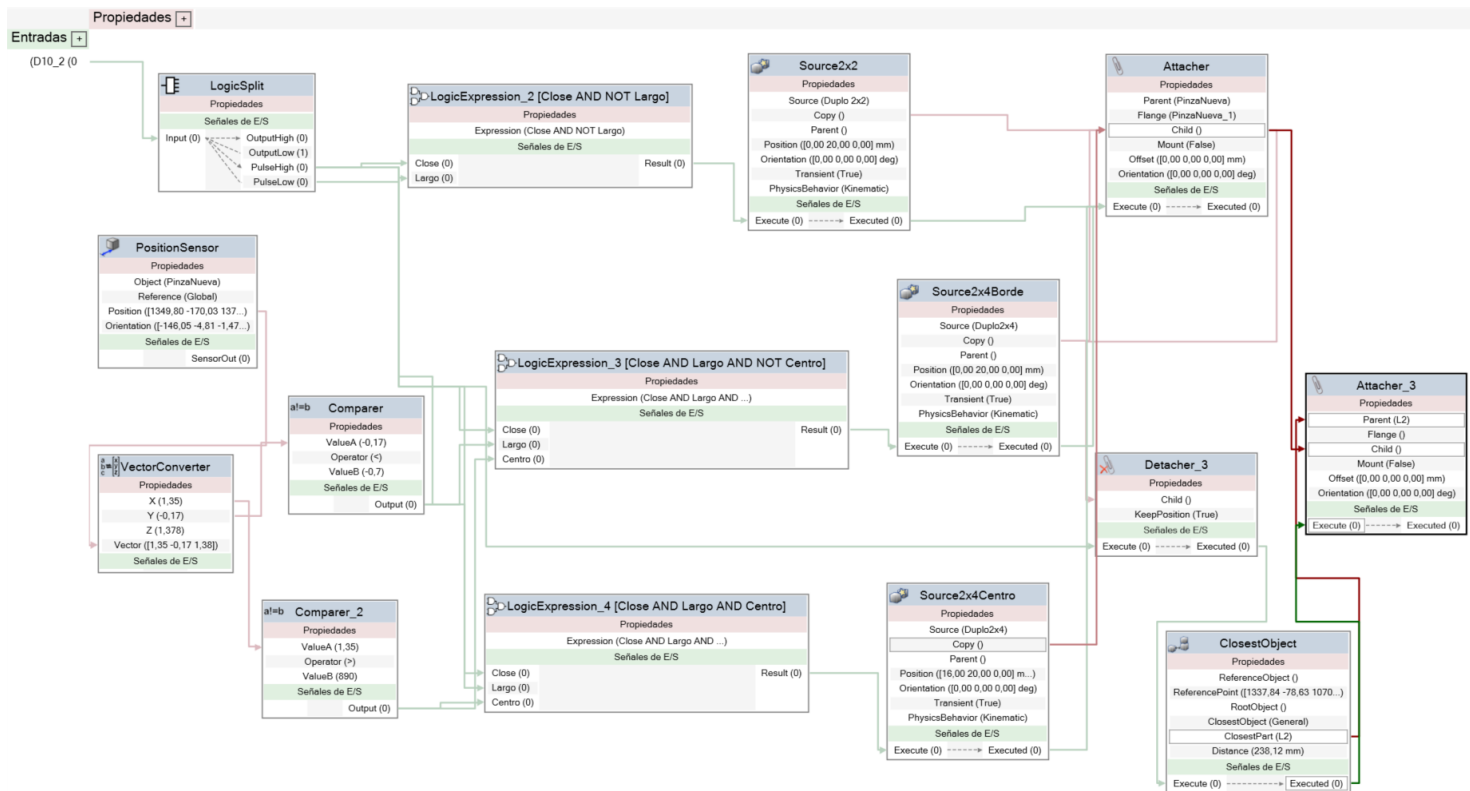


FIGURE 9: BLOCK "LEGO PIECES"

Also, a somewhat controversial issue with RobotStudio software must be tackled: exporting and importing files. A container is able to package all the code and dependencies of a project in a standard format for safe and agile execution on another device. Similarly, Pack and Go or Unpack and Work can be used in RobotStudio.

- **Pack and Go:** is used when all the elements of a project should be packed in order to be able to open it on another device.
- **Unpack and Work:** is the option to select when the objective is to open a Pack and Go on a device.

Of course, for the correct development of this and any other project, this way of exporting files is recommended. It is the safest option to make sure that all components of the work are safely transported from one device to another.

As the last part of this section, the main problems that arose when developing the tool in RobotStudio will be introduced.

- Poor curves in the pallet trajectories were corrected to solve problems of incorrect movement on the conveyor belt.
- In the simulation, a volumetric sensor was added to detect if the pallet is carrying lego pieces, since accurate cameras were not included as in the real Mini-factory.
- The SWIFTI CRB 1100 robot at Station 3 presented difficulties in its movements due to its limited space and inverted placement.
- The movements of the lego pieces proved challenging in the simulation, requiring additional skills and more complex sensors to achieve their movement along with the pallet.
- The physics of the pieces was a bottleneck, so kinematic behavior was chosen to allow better interaction with the plant and more control over the pieces.
- The inclusion of new 3D models in the plant design was complicated due to file compatibility issues with RobotStudio, resulting in a final representation that was not completely faithful to the real model, but functional for the simulation objectives.

V. SAFETY PROTOCOL

Once the programming and simulation of the four robots in the Minifactory was completed, it was decided to perform an investigation on the possible implementation of safety on the GoFa CRB 15000 Cobots. The main idea was to implement the solution provided by ABB to enable closer collaboration between robots and factory workers, SafeMove

SafeMove, ABB's innovative safety proposition, offers increased versatility, space optimization and advanced setup tools that increase productivity and reduce investment costs. This exceptional solution ensures highly robust safety, fostering close collaboration between robots and factory workers, while lowering total investment by up to 30%.

First of all, it was decided to look for enough information to have a formed idea of the possibilities of implementing the safety protocols in RobotStudio. Since, generally speaking, programming in the physical laboratory has much more updates and documentation than development in a simulated environment.

After reviewing many manuals on the subject, one was found that defined the user interface of Visual SafeMove in RobotStudio. This documentation explains that a safe zone, that is, a volume, can be defined. In this, it was possible to configure the position, orientation, tool force, axis or tool speed... but it was not possible to simulate the interaction of all these functions with the appearance of a person in the safe zone. In addition, it was also possible to define a range of safety or degrees in which the robot axes could move. However, this configuration did not work with the space limitations of the Mini-Factory, resulting in the robots constantly stopping or causing failures in the flow of movements in the laboratory.

The next attempt focused on the implementation of the Virtual Human library [33]. It was then discovered that ABB incorporated this package into their software and called it Collaborative Speed Control a few weeks ago. In general terms, the new guides found about this library explained that the controllers had to be reintroduced in order to configure them with the packages required to implement the safety simulation. In other words, it was required to delete the existing robots and start the project practically from the beginning, since the necessary modifications in the configuration of the controllers were not allowed.

Finally, it was concluded that the best alternative would be to develop a simple homemade simulation, in which a box would perform the movement of crossing the safety zones as if it were an operator. To this end, the `init()` function was programmed, responsible for the initial configuration of all

the safety variables and the declaration of the initial speed of the robots. All the interruptions were also programmed to change the value of the robot speed as required (0% speed, 30% speed or 100% speed). The operation is as follows: if the box crosses the first safety zone, the robot reduces its speed to 30%. On the other hand, if the box manages to reach the danger zone, the robot stops completely to avoid harming the human operator.

In addition, *Figure 10* shows both buttons, "iMoverCaja" and "iResetCaja", to activate the movement of the box or to reset its position.

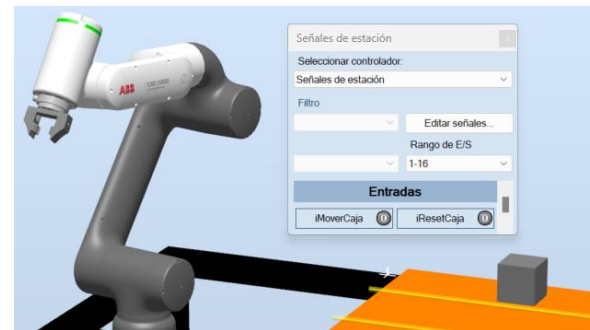


FIGURE 10: "STATION SIGNALS" WINDOW FOR SAFETY PROTOCOL

VI. BUDGET

As in any project of these characteristics, it has been considered convenient to attach a chapter related to the estimated budget that would be needed to implement the development of this Master's Thesis in the industry.

On the one hand, the largest investment should be made in human resources. This project has a crucial component for the development of this digital twin: the working hours of an engineer. In general terms, the practical part of this project was developed in approximately eight months, so it is estimated that this would be the time interval necessary to obtain the finished work. It is worth mentioning that the engineer must be trained in the industrial robotics sector, and must have the necessary qualities to be able to program in Rapid and make good use of all the RobotStudio tools, as well as their corresponding logic. Finally, this professional must also know a design software, such as Solid Edge, to be able to model all the 3D prototypes needed to build the structure of the complete laboratory plant.

On the other hand, all licenses used throughout the project must also be taken into account. Both, the software focused on robotic applications, RobotStudio, and the computer-aided design program, Solid Edge, cannot be downloaded free of charge.

Finally, to get the maximum benefit from this simulation, it is desirable to have a physical laboratory with facilities similar to the ICAI Mini-factory. In addition, it should be equipped with Cobots, such as the GoFa CRB 15000 and the SWIFTI CRB 1100.

VII. RESULTS ANALYSIS

Comparing the simulation carried out in RobotStudio with the reality of the ICAI Mini-factory, the balance is extremely positive. The movements and the workflow of the real laboratory have been practically perfectly traced. When the project is executed, it is capable of processing the four controllers at the same time in a precise manner, in which it can be seen how the assembly and disassembly stations work simultaneously to achieve a result typical of a real industry.

It is worth mentioning that two significant bottlenecks have been identified when simulating this project, although at all times they have been solved by making good use of the tools provided by RobotStudio. On the one hand, the accuracy of the gripper to pick up or put down the pieces was a problem of high precision for this action to be realistic. On the other hand, the interaction of the robot with the loader required programming a kinematic configuration, which meant many weeks of work to deal with importing files into RobotStudio, creating meshes around the geometries and building the precise transformations that would make the robot interact with the loader in an optimal way with respect to reality.

VIII. CONCLUSIONS AND FUTURE DEVELOPMENTS

In conclusion, the main objectives of this project have been covered with great results. The main idea was to achieve a completely virtual model to make the ICAI Mini-Factory accessible to the student remotely. In addition, this platform has an almost complete resemblance to the reality of the laboratory. Therefore, as imposed in the project specifications, the student, thanks to the simulation developed, is able to design and program a completely modular factory from the point of view of industrial robotics.

It is worth mentioning that, throughout the development of this work, priority has been given at all times to obtain a modular tool. It is true that it is always necessary to make a tedious initial investment to build the laboratory infrastructure, but it has been tried to design the mechanisms, define the joints and, in general, program all the elements that make up the simulation in a systematic way to achieve the greatest possible scalability to other configurations of the Cobots or even to other projects.

Finally, this report will end with a reference to the future work of this application. This is a simulation model that provides very accurately what will happen in reality in the laboratory, so it would be desirable to introduce new elements as the ICAI Mini-Factory is updated. For example, it might be desirable to add a warehouse or incorporate a mobile robot mounted on a platform. In addition, another breakthrough would be linked to achieving a complete digital twin of the laboratory.

Feedback from the simulation to the real model can provide many precision, control and safety features. Lastly, as the RobotStudio robotics tool is enhanced and new updates are introduced, safety protocols, such as Virtual SafeMove, can be implemented in a more robust manner.

APPENDIX A

Rapid Code for Station 1:

```

MODULE Module1
    CONST robtarg Home=[[-60.983,-
313.902,732.067],[0.043834723,-0.000002953,-
0.955679747,0.291126669],[-2,-1,-
2,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
    CONST robtarg
Target_10=[413.769,195.88,20],[0,0,1,0],[0,0,0,0],[9E+09,9E+
09,9E+09,9E+09,9E+09,9E+09]];
    CONST robtarg Target_20=[[-64.268,-
405.437],[0,0.707106781,0.707106781,0],[-2,0,-
1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
    CONST robtarg
Target_30=[396.686094163,179.770793858,75.733962425],[0.0008
22978,-0.706698423,-0.707513923,-
0.000842734],[0,0,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
    CONST robtarg Pale=[[-60.983,-405.492.734],[0,0,1,0],[-
2,0,-2,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
    CONST robtarg
Lego=[396.686,179.771,75.734],[0.000821186,-0.706698924,-
0.707513423,-
0.000844909],[0,0,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
    CONST robtarg
Lego2=[396.686,279.771,75.734],[0.000821186,-0.706698924,-
0.707513423,-
0.000844909],[0,0,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
    CONST robtarg Pale2=[[-60.983,-
405.511.934],[0,0,1,0],[-2,0,-
2,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
    PERS robtarg paleTarget=[[-60.983,-
373.512.734],[0,0,1,0],[-2,0,-
2,0],[9E+9,9E+9,9E+9,9E+9,9E+9,9E+9]];
    CONST robtarg Pale H=[[-60.983,-
405.492.734],[0,0.707106781,0.707106781,0],[-2,0,-
1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
    CONST robtarg Pale rot_4=[[-60.983,-405.492.734],[0,-
0.707106781,0.707106781,0],[0,0,0,0],[9E+09,9E+09,9E+09,9E+09,
9E+09,9E+09]];
    CONST robtarg Pale rot3=[[-60.983,-
405.492.734],[0,1,0,0],[-
2,0,0,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
    CONST robtarg
Lego3=[380.686010624,279.75257064,75.733441498],[0.000821186
,-0.706698924,-0.707513423,-
0.000844909],[0,0,1,0],[9E+09,9E+09,9E+09,9E+09,9E+09,9E+09]];
    VAR bool setupDone:=FALSE;
    VAR bool Montar:=FALSE;
    VAR intnum irSpeedSlow;
    VAR intnum irSpeedFast;
    VAR intnum irSpeedNull;

! Programa principal de la estación
PROC main()
    MoveL Home,v2000,fine,PinzaNueva_1\WObj:=wobj0;
    SetDO O_ROBOT1_P1,0;
    IF setupDone THEN
        esperaPale;
        IF Montar=TRUE THEN
            montaFigura;
            bajaPale;
        ENDIF
    ELSE
        init;
        setup;
        setupDone:=TRUE;
    ENDIF
ENDPROC

! Configuración inicial de variables de seguridad
PROC init()
    IDelete irSpeedSlow;
    CONNECT irSpeedSlow WITH tSlowSpeed;

```

```

ISignalDI I_OBJ_CLOSE,high,irSpeedSlow;
IDelete irSpeedFast;
CONNECT irSpeedFast WITH tFastSpeed;
ISignalDI I_OBJ_CLOSE,low,irSpeedFast;
IDelete irSpeedNull;
CONNECT irSpeedNull WITH tNullSpeed;
ISignalDI I_OBJ_DANGER,high,irSpeedNull;
IF TestDI (I_OBJ_CLOSE) THEN
    SpeedRefresh 50;
ELSE
    SpeedRefresh 100;
ENDIF
ENDPROC

TRAP tSlowSpeed
    SpeedRefresh 30;
ENDTRAP

TRAP tFastSpeed
    SpeedRefresh 100;
ENDTRAP

TRAP tNullSpeed
    SpeedRefresh 0;
ENDTRAP

! Configuración inicial de los actuadores
PROC setup()
    SetDO O_GSTOP_OFF,1;
    SetDO O_GSTOP_ON,0;
    SetDO O_S1_PSTOP_ON,1;
    SetDO O_S1_PSTOP_OFF,0;
    SetDO O_S1_STOP_ON,1;
    SetDO O_S1_STOP_OFF,0;
    SetDO O_S1_LIFT_DOWN,1;
    SetDO O_S1_LIFT_UP,0;

    SetDO O_D12_ARM_SH,0;
    SetDO O_D12_ARM_ST,1;
    SetDO O_D12_STOP_ON,0;
    SetDO O_D12_STOP_OFF,1;
ENDPROC

! Esperar palé
PROC esperaPale()
    SetDO O_S1_PSTOP_ON,1;
    SetDO O_S1_PSTOP_OFF,0;
    WaitDI I_S1_PSTOP,1;
    SetDO O_S1_STOP_ON,1;
    SetDO O_S1_STOP_OFF,0;
    WaitTime 1;

    IF I_S1_Lego=1 THEN
        SetDO O_S1_PSTOP_ON,0;
        SetDO O_S1_PSTOP_OFF,1;
        SetDO O_S1_STOP_ON,0;
        SetDO O_S1_STOP_OFF,1;
        WaitTime 1;
        SetDO O_S1_PSTOP_ON,1;
        SetDO O_S1_PSTOP_OFF,0;
        Montar:=FALSE;
    ELSE
        Montar:=TRUE;
        SetDO O_S1_PSTOP_OFF,1;
        SetDO O_S1_PSTOP_ON,0;
        WaitTime 0.3;
        SetDO O_S1_PSTOP_OFF,0;
        SetDO O_S1_PSTOP_ON,1;
        WaitDI I_S1_STOP,1;
        WaitTime 1;
        SetDO O_S1_LIFT_DOWN,0;
        SetDO O_S1_LIFT_UP,1;
        WaitDI I_S1_LIFT_UP,1;
        WaitTime 1;
        SetDO O_S1_STOP_ON,0;
        SetDO O_S1_STOP_OFF,1;
        SetDO O_S1_PSTOP_ON,0;
        SetDO O_S1_PSTOP_OFF,1;
    ENDIF
ENDPROC

! Llevar piezas a sus posiciones de montaje
PROC montaFigura()
    colocaLego 0,-2,0,1,FALSE,TRUE;
    colocaLego 0,2,0,1,FALSE,TRUE;
    colocaLego -4,2,0,1,FALSE,TRUE;
    colocaLego -4,-2,0,1,FALSE,TRUE;

    colocaLego 0,-1,1,1,FALSE,TRUE;
    colocaLego -1,2,1,1,FALSE,TRUE;
    colocaLego -4,1,1,1,FALSE,TRUE;
    colocaLego -3,-2,1,1,FALSE,TRUE;

    colocaLego 0,1,2,1,TRUE,TRUE;
    colocaLego -1,4,2,2,TRUE,TRUE;
    colocaLego -4,-1,2,1,TRUE,TRUE;
    colocaLego 0,-4,2,4,TRUE,FALSE;
ENDPROC

! Bajar palé
PROC bajaPale()
    SetDO O_S1_PSTOP_ON,1;
    SetDO O_S1_PSTOP_OFF,0;
    WaitTime 2;
    SetDO O_S1_LIFT_DOWN,1;
    SetDO O_S1_LIFT_UP,0;
    WaitDI I_S1_LIFT_DOWN,1;
    WaitTime 2;
    SetDO O_S1_STOP_ON,1;
    SetDO O_S1_STOP_OFF,0;
ENDPROC

! Instrucciones montar una pieza
PROC colocaLego(num x,num y,num z,num rot,bool largo,bool
centro)
    paleTarget.trans.x:=Pale.trans.x+32/2*x;
    paleTarget.trans.y:=Pale.trans.y+32/2*y;
    paleTarget.trans.z:=Pale.trans.z+19.2*z;
    IF rot=1 THEN
        paleTarget.rot.q1:=Pale.rot.q1;
        paleTarget.rot.q2:=Pale.rot.q2;
        paleTarget.rot.q3:=Pale.rot.q3;
        paleTarget.rot.q4:=Pale.rot.q4;
    ENDIF
    IF rot=2 THEN
        paleTarget.rot.q1:=Pale_H.rot.q1;
        paleTarget.rot.q2:=Pale_H.rot.q2;
        paleTarget.rot.q3:=Pale_H.rot.q3;
        paleTarget.rot.q4:=Pale_H.rot.q4;
    ENDIF
    IF rot=3 THEN
        paleTarget.rot.q1:=Pale_rot3.rot.q1;
        paleTarget.rot.q2:=Pale_rot3.rot.q2;
        paleTarget.rot.q3:=Pale_rot3.rot.q3;
        paleTarget.rot.q4:=Pale_rot3.rot.q4;
    ENDIF
    IF rot=4 THEN
        paleTarget.rot.q1:=Pale_rot_4.rot.q1;
        paleTarget.rot.q2:=Pale_rot_4.rot.q2;
        paleTarget.rot.q3:=Pale_rot_4.rot.q3;
        paleTarget.rot.q4:=Pale_rot_4.rot.q4;
    ENDIF

    IF largo THEN
        IF centro THEN
            MoveL
            Lego3,v2000, fine, PinzaNueva_1\Wobj:=wobj0;
        ELSE
            MoveL
            Lego2,v2000, fine, PinzaNueva_1\Wobj:=wobj0;
        ENDIF
    ELSE
        MoveL Lego,v2000, fine, PinzaNueva_1\Wobj:=wobj0;
    ENDIF

    SetDO O_ROBOT1_P1,1;
    MoveL Home,v2000,z100,PinzaNueva_1\Wobj:=wobj0;
    paleTarget.trans.z:=paleTarget.trans.z+20;
    MoveL paleTarget,v2000, fine, PinzaNueva_1\Wobj:=wobj0;
    paleTarget.trans.z:=paleTarget.trans.z-20;
    MoveL paleTarget,v2000, fine, PinzaNueva_1\Wobj:=wobj0;
    SetDO O_ROBOT1_P1,0;
    MoveL Home,v2000,z100,PinzaNueva_1\Wobj:=wobj0;
ENDPROC
ENDMODULE

```


APPENDIX B

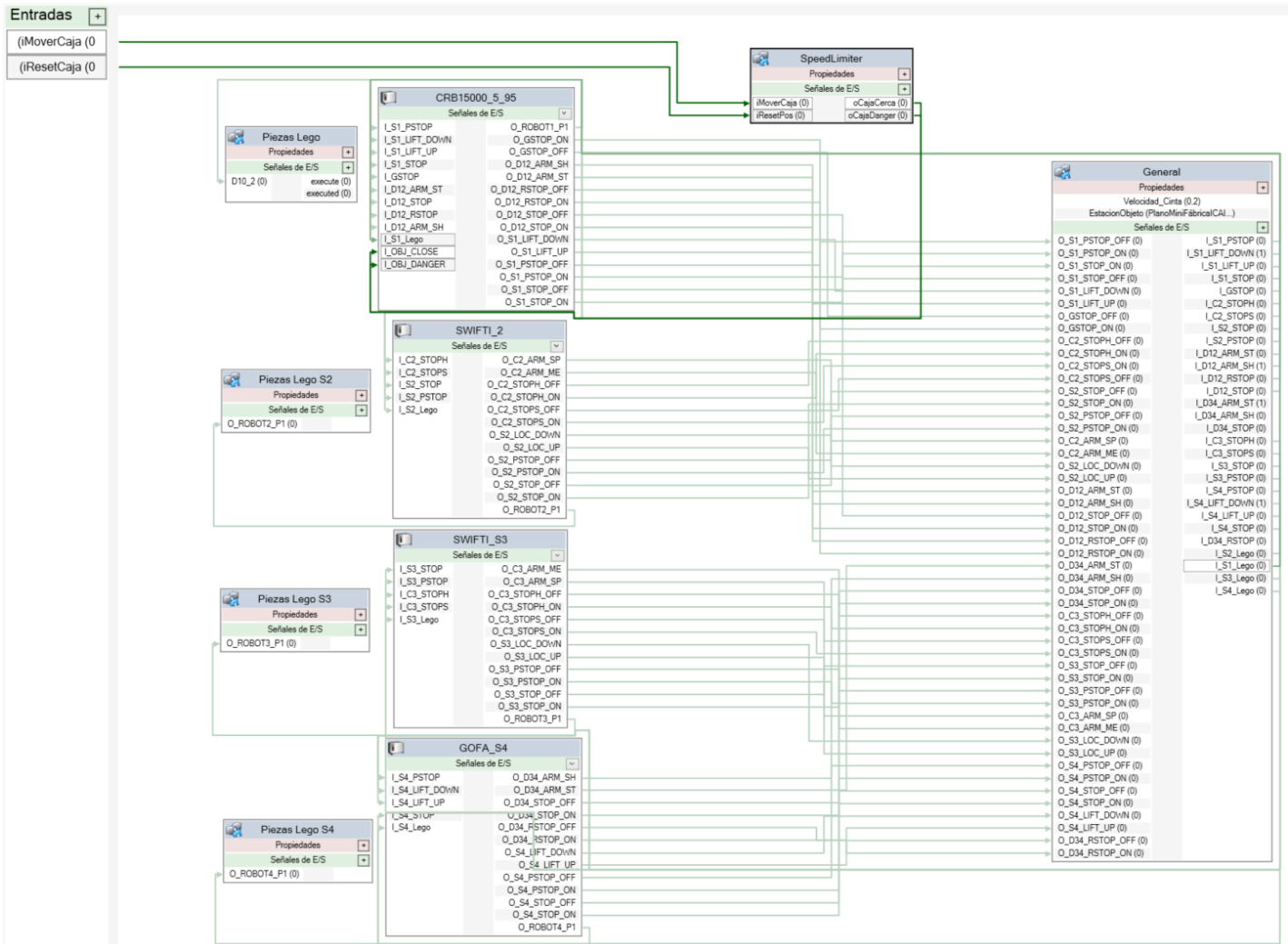


FIGURE 11: LOGIC OF GENERAL STATION

REFERENCES

- [1] ABB, «RobotStudio Suite.» [En línea]. Available: <https://new.abb.com/products/robotics/es/robotstudio>. [Último acceso: 2023].
- [2] ABB, «GoFa CRB 15000.» [En línea]. Available: <https://new.abb.com/products/robotics/robots/collaborative-robots/crb-15000>. [Último acceso: 2023].
- [3] ABB, «SWIFTI.» [En línea]. Available: <https://new.abb.com/products/robotics/robots/collaborative-robots/swifti>. [Último acceso: 2023].
- [4] ABB, «Collaborative Robots.» [En línea]. Available: <https://new.abb.com/products/robotics/robots/collaborative-robots>. [Último acceso: 2023].
- [5] Automate, «UNIMATE The First Industrial Robot.» [En línea]. Available: <https://www.automate.org/a3-content/joseph-engelberger-unimate>. [Último acceso: 2023].
- [6] H. d. empresas, «El protagonismo decisivo de los ‘cobots’ en la industria 4.0.» [En línea]. Available: <https://hablemosdeempresas.com/grandes-empresas/queson-los-cobots/>. [Último acceso: 2023].
- [7] U. ROBOTS, «LA VENTA DEL PRIMER COBOT DE UNIVERSAL ROBOTS.» [En línea]. Available: <https://www.universal-robots.com/es/acerca-de-universal-robots/noticias/historia-de-los-cobots/>. [Último acceso: 2023].
- [8] movicontrol, «Robot colaborativo: marcas y ejemplos.» [En línea]. Available: <https://movicontrol.es/robots-colaborativos/>. [Último acceso: 2023].
- [9] weir, «Our Linatex Range of Rubber.» [En línea]. Available: <https://www.global.weir/brands/linatex/?lang=es>. [Último acceso: 2023].
- [10] SIEMENS, «Solid Edge.» [En línea]. Available: <https://solidedge.siemens.com/en/>. [Último acceso: 2023].
- [11] UltiMaker, «UltiMaker Cura.» [En línea]. Available: <https://ultimaker.com/software/ultimaker-cura/>. [Último acceso: 2023].
- [12] ABB, «RobotStudio®.» [En línea]. Available: <https://new.abb.com/products/robotics/es/robotstudio/robotstudio#:~:text=Introducci%C3%B3n%20de%20RobotStudio&text=Permite%20simulaci%C3%B3n%20en%20tiempo,sin%20afectar%20la%20producci%C3%B3n..> [Último acceso: 2023].



MARINA BEATRIZ LETTIERI PALENCIA is a highly motivated and organized individual, known for her ability to inspire and lead teams effectively. She thrives in collaborative settings and eagerly embraces challenges to enhance her professional growth. With a strong passion for social progress and technological advancements, Marina approaches each day as an opportunity to confront diverse challenges and contribute to the improvement of society.

During her professional journey, Marina has gained valuable experience as a Business Analyst Intern at Amazon in Luxembourg. Here, she conducted in-depth profitability research and impact analysis to create reports for internal and external teams. Her focus on optimizing resources and building trust with customers led her to develop standard automation paths and implement models in the AWS public cloud.

Additionally, Marina has served as an Industry 4.0 Business Intelligence Engineer Intern at Gestamp. Within the Industry 4.0 adoption team, she conducted data analysis and research tasks to evaluate and interpret data, collaborating closely with Gestamp industrial plants. Her proficiency in Power BI, SQL, R, and company-specific development environments facilitated process improvements and technology implementation.

Furthermore, Marina contributed to the advancement of technology at the Technological Research Institute, ICAI, in Madrid. As an Electronics, Automation, and Communications Intern, she played a crucial role in bringing the automation laboratory to Industry 4.0. Her responsibilities included programming in RAPID and C# languages, as well as designing prototypes of 3D robotic components using SolidWorks and SolidEdge software.

In terms of education, Marina holds a Double Master's Degree in Industrial Engineering and Smart Industry from Universidad Pontificia Comillas, ICAI, where she specialized in Machine Learning, IoT, Big Data, Cybersecurity, Deep Learning, and 4.0 Industry. She also earned a Bachelor's Degree in Industrial Engineering and obtained Diplomas in Personal, Communication, and Professional Skills, as well as Communication Skills and Studies in a Foreign Language.

Marina is multilingual, being a native Spanish speaker and possessing an advanced level of proficiency in both English and French. Her abilities encompass a wide range of technical skills, including Microsoft Office, operating systems, development in virtual environments and containers, simulation and design with various software tools, database management, digital twins execution in Unity and RobotStudio, and programming in multiple languages.

Beyond her academic pursuits, Marina actively engages in extracurricular activities, demonstrating her commitment to personal growth and community involvement. Notably, she served as the Head of the Marketing and Communication Department in ISC Racing Team at ICAI and held significant roles as a School Subdelegate, member of the board and cloister at Universidad Pontificia Comillas, ICAI. Furthermore, Marina has dedicated her time as a Volunteer Collaborator in the Charcoal production process in Abidjan, Côte d'Ivoire, and has contributed as a Social Work Unit Intern at ICAI.

Marina's exceptional soft skills, including teamwork, stress management, leadership, public speaking, innovation, project management, and creativity, round out her well-rounded profile. Her dedication and adaptability make her an invaluable asset in any professional setting, while her commitment to community service showcases her altruistic nature and passion for making a positive impact on society.