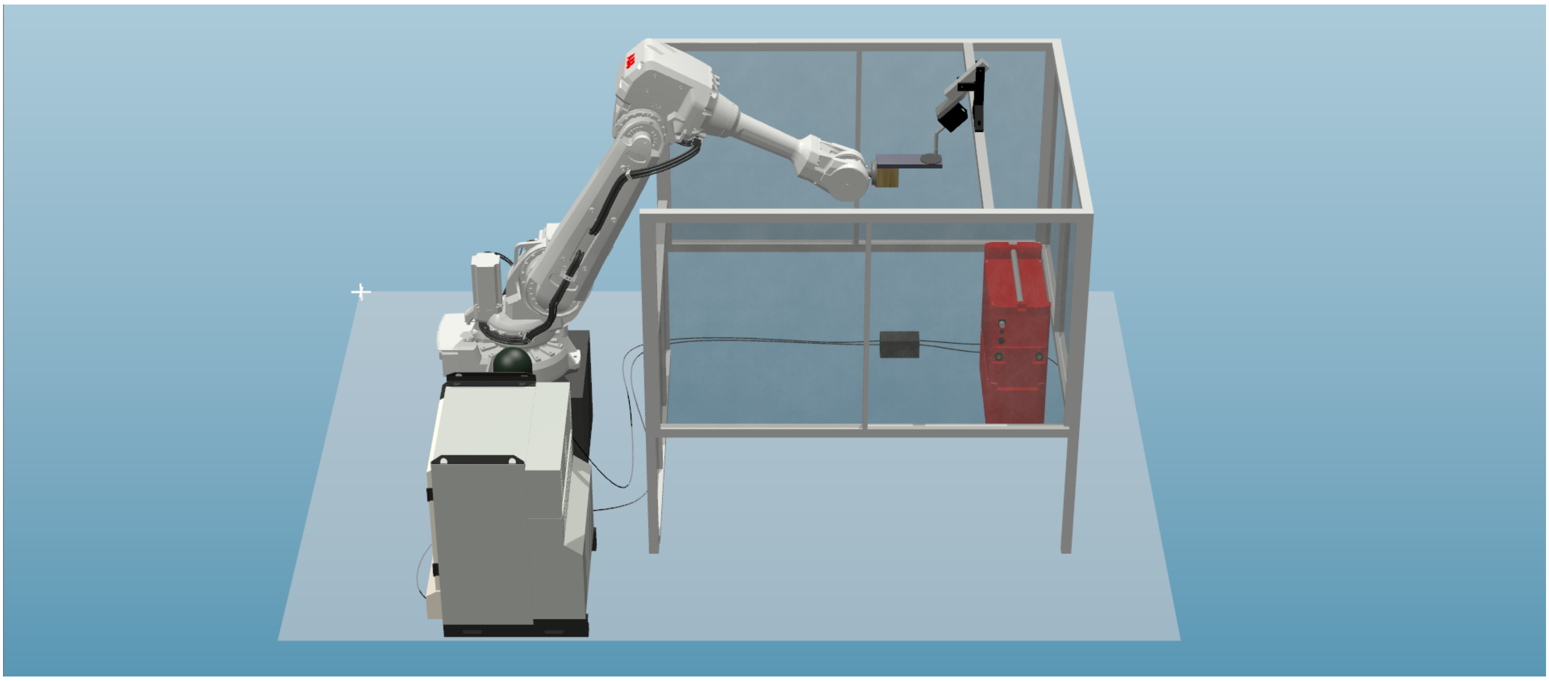
Second Year Internship at IndexLab, Politecnico di Milano

Supervisor :
M. Anthierens

Internships tutor :
M. Ruttico

Student :
Thibault COURTOIS

Academic year 2023-2024
Internships report



Non-Plagiarism Commitment [1]

I, the undersigned, Thibaut Courtois.

Student ID number: 22205247.

Hereby declare that I have read and understood the examination regulations, particularly the section specific to plagiarism. I am fully aware that copying documents or parts of documents published in any form (books, publications, student reports, internet, etc.) without citation or reference constitutes plagiarism and is a violation of copyright as well as a clear act of fraud.

Therefore, I commit to citing all the sources I have used to produce and write this document.

Done on 18/08/2024

Signature

[1] Document from Chapter 10, Annex 5, originating from the Examination Charter adopted by the Board of Directors on July 11, 2013, following the opinion of the CEVU on June 27, 2013 - Resolution No. 2013-73 - Modified following the CFVU on March 12, 2015.

Summary and Keywords

English

During my internship in IndexLab, a leading laboratory specializing in design, manufacturing, artworks and architectural processes located in Italy, I engaged in a variety of assignments focused on advanced robotized and automatized fabrication processes. I worked autonomously on additive manufacturing tasks using an ABB robotic industrial arms and 3D printers. This internship greatly enhanced my knowledge and skills in industrial robotic technologies field. Indeed, I learned how to use Rhino 3D to design robotic tools, programmed robots with GrassHopper and RobotStudio for several industrial applications such as Pick&Place, Welding and WAAM. Notably, I set up two WAAM process that I optimized with a testing procedure to find correct welding parameters and have written a guide on the utilization of RobotStudio and ABB robots.

French

Lors de mon stage à IndexLab, un laboratoire de premier plan spécialisé dans le design, la fabrication, les œuvres d'art et les processus architecturaux situé en Italie, j'ai été impliqué dans une variété de missions axées sur les processus de fabrication robotisés et automatisés avancés. J'ai travaillé de manière autonome sur des tâches de fabrication additive en utilisant un bras robotique industriel ABB et des imprimantes 3D. Ce stage a considérablement amélioré mes connaissances et compétences dans le domaine des technologies robotiques industrielles. En effet, j'ai appris à utiliser Rhino 3D pour concevoir des outils robotiques, à programmer des robots avec GrassHopper et RobotStudio pour plusieurs applications industrielles telles que le Pick&Place, le soudage et le WAAM. Notamment, j'ai mis en place deux processus WAAM que j'ai optimisés avec un protocole de test pour trouver les paramètres de soudage corrects et j'ai rédigé un guide sur l'utilisation de RobotStudio et des robots ABB.

Keywords

- **Industrial robotic** - Robotique industrielle
- **Robotic simulation** - Simulation robotique
- **Robots programming** - Programmation de robots
- **Tool fabrication** - Fabrication d'outils
- **Calibration** - Calibration
- **Additive manufacturing** - Fabrication additive
- **3D printing** - Impression 3D
- **Arc welding** - Soudage à l'arc électrique
- **WAAM (Wire Arc Additive Manufacturing)** - Fabrication additive par soudage à l'arc électrique de filament métalique
- **Pick&Place** - Préhension et placement
- **Tests protocol** - Protocoles de tests
- **Process optimization** - Optimisation de procédés
- **Quaternion algebra** - Algèbre des quaternions
- **Scientific document writing** - Rédaction de documents scientifiques
- **International and professional experience** - Expérience professionelle internationale

Acknowledgments

IndexLab laboratory has a wonderful team of skilled professionals, but also lovely individuals that always helped me when I asked for a hand during my internship. I want to specifically acknowledge **Mr. Ruttico**, head of IndexLab, for taking me as an intern, trusting me and giving such interesting internship assignments. **Mrs. El Bakkali** for her tireless good mood and assistance, **Mr. Qiulin**, **Mr. Deval** and of course **Mr. Bordoni** for their help and advice all along my internship. I greatly appreciated my too short membership in IndexLab team.

I also want to thank the people I work with, in the name of **Mrs. Khakpour**, **Mr. Pini** and especially **Mr. Rajech Kulkani** who now shares with me an enthusiastic interest for WAAM technology and will, I hope, push further IndexLab's WAAM robotized process.

Finally, I want to acknowledge **Mr. Viscardi**, Politecnico di Milano professor, who helped me for the connection between the ABB robot and the Fronius power unit and the testing of the little elephant robotic cobot and **Mr. Beltracchi** for his expertise on robot integration inside GrassHopper environment.

Table

Introduction

This internship was a mandatory internship as an assistant engineer for my engineering cursus at Seatech. I decided this year to do an international work experience in Italy, specifically in a laboratory of Politecnico di Milano : IndexLab. These seventeen weeks internship were a great opportunity for me to develop my knowledge and skills in the robotic and mechatronic field by working in a laboratory specialised in robotized and automatized manufacturing processes such as WAAM.

I came to Lecco in April with an unflagging motivation to learn and progress in my field in an international environment where I could as well improve my English communication skills in real and meaningful situations. Mr. Ruttico the professor and director of IndexLab was interested in my robotic and mechatronic background and gave me the mission to work on robotic creation projects. It is a vast domain covering manufacturing, architecture, construction and artworks applications that can be treated with several processes : machining, cutting, assembling, polishing ... I interested myself in additive manufacturing processes, especially WAAM (Wire Arc Additive Manufacturing).

Indeed, additive manufacturing is a promising technology much faster and less expensive than current machining processes. In the run for mechanically optimized components for aircraft, transport and industry in general, topologically optimized pieces are being developed. Additive manufacturing is key for such pieces, as they are very expensive to sculpt from a block of material with machining processes and really specific to each of their application and consequently mass product with molding techniques. Arc welding or laser welding additive manufacturing might be the future of fabrication industry and is a growing research sector leaded by the company WAAM3D that printed and deployed an urban bridge in Amsterdam in 2021.

In this report, I will detail my path towards the comprehension and the set-up of a functional robotized WAAM process. First, I will describe IndexLab's work environment and present you my various internship's assignments, then I will dedicate a part to ABB IRB 4600-205-45 robotic arm I worked with and learned to control. After that, I will focus on the new CAD and robot programming softwares I learned to design my own robot tools, to simulate and program the robot. Furthermore, I will briefly speak about quaternion algebra in robotic, as I decided to learn some quaternion's concept to enlarge my understanding of invert kinematic algorithm. Finally, I will describe the two processes of WAAM I worked on to produce some real steel pieces.

1 IndexLab work environment

1.1 IndexLab's issues and objectives

IndexLab is a laboratory of Politecnico di Milano and is located in the Lecco campus. IndexLab's work embrace art, engineering, architecture and product design and offers research, consultancy, education and artworks services. Indeed, IndexLab uses it's cutting edge expertise to launch innovative construction systems to explore the future of construction, advanced design and manufacturing projects involving computational design, digital and robotic fabrication, interaction and media design. Consequently, Indexlab's goal is to turn ambitious ideas into proven systems and technologies that make a real impact in the world by handling the early-stage research to the deployment of the project.

IndexLab is evolving in an academic environment and has for the last decade educated engineers and architects students on architecture and practice trend, design and manufacturing. The team believes that continuing education provides a common ground for growth and for this reason share knowledge through academic courses and professional training workshops as well as trade fairs and specialists conferences. Indeed, during my internship, students coached by IndexLab team were using IndexLab manufacturing machine and equipment to create and develop an innovative facade project. The head of IndexLab, M. Ruttico and some members of the team were also supervising master thesis students I had the pleasure to help for some of them during my internship.

Finally, artworks is deeply rooted in IndexLab's as it is a fertile cross-disciplinary playground to push the boundaries of design and manufacturing. Consequently, IndexLab is involved in several special events or piece of art museum where the team displays interactive experiences and innovative designs that reflect the culture of the world around us.

1.2 Functioning and organization

IndexLab is composed of a little team gathering research fellows from engineering and architecture background. Each research fellow is involved in several projects sponsored by companies specialized in construction, manufacturing, design and several other fields. For example, during my internship, four research fellows were working on different projects such as : artificial intelligence based wood defect detection, manufacturing robotic cells dimensioning, robotized construction process and human-robot collaboration. Mr. Ruttico takes the role of project manager and is giving counseling, advising and directing the different projects.

Finding sponsors is also a very difficult job in itself as it requires from IndexLab to organize several meeting, showcase and of course to do all the paperwork necessary to link three organizations: Polimi, the companies and IndexLab. Of course, during the projects they are involed, member of the team meets their client and communicate their progress when necessary.

Each member of the team is involved in education missions such as attempting to students project presentations, helping students, giving class ... They are doing everything they can to push students toward greatness and accomplishment.

IndexLab team has a role to play in sponsorship finding, but every financial and administrative issues are solved by a constant dialogue between Mr. Ruttico and Polimi council. For example, IndexLab has a limited number of research fellowship seats and the distribution of posts inside the laboratory is debated each year in a council. Thus, IndexLab way of functioning is closed to a start-up for advertisement, client attraction and work atmosphere without the financial part that is managed by Polimi and Mr. Ruttico. IndexLab is following the rules of Polimi and in not independent.

1.3 Competition

IndexLab clients are mostly Italian companies, but they are also working for international companies and city projects. Even if IndexLab is progressing in an academic environment, there are still competition issues as IndexLab is not the only laboratory to work on computational design and robotic fabrication on the international scale (in the other hand locally in Italy there are really few to do it and IndexLab is linked to the most recognized university in Italy).

IndexLab targets are companies that are not big enough to have a specified and dedicated research and development department but are still motivated by innovation and new technologies. They are seeing in IndexLab a profitable environment able to quickly adapt to meet their needs. IndexLab benefits of the Politecnico di Milano structure to recruit talented students and individuals, to be at the cutting edge of architecture and various science fields, and to guarantee its economic stability and longevity to outperform the competition they can undergo from startups and other academic laboratory.

IndexLab has found its way of functioning by negotiating sponsorship with other companies. Other laboratory are more focused on scientific publication and patent that are the other possible way to create value as an academic laboratory.

1.4 Summary

IndexLab, part of Politecnico di Milano, specializes in art, engineering, and design, offering research and consultancy with a focus on advanced construction and robotic fabrication. It aims to turn innovative concepts into impactful technologies and provides education through academic courses and professional training. The lab operates with a small team on diverse, sponsored projects and faces international competition. It stands out by quickly adapting to industry needs, supported by its university affiliation.

2 Internship assignments

During this internship, I had various assignments specific to additive manufacturing with robotic industrial arms and 3D printers. Here is the detail of my missions, ordered chronologically. The tasks that were given to me were conducted in total autonomy except for the manual work such as the assembly of the final WAAM robotic cell. The first Gcode WAAM process was also a collaborative work between a master thesis student and another Politecnico di Milano student doing a master in civil engineering.

- I first learned how to use Rhino 8, and it's parametric CAD tool : GrassHopper to be able to design and print the pieces I needed for my other assignments and create some basic programs and simulation for the ABB industrial robotic arm I was working with in IndexLab.
- During the internship period, in parallel with my other work, I was implicated in the maintenance of 3D printers (BambuLab X1 carbon and Prusa MK3 model) especially to help students during their innovative facade project period.
- I participated in an interactive class for high school students (3 sessions in 2 weeks). The objective was to show industrial robotic arm technology to the students and to teach them the basic of industrial robot programming.
- Then I mastered the use of the ABB robot, learned to use RobotStudio and how to program in RAPID (ABB robot programming language). I have written a guide "GrassHopper to RobotStudio" for IndexLab, that explains how to integrate a GrassHopper robotic simulation into RobotStudio as well as the jogging system of the robot, some RAPID language basics and the calibration processes employed by ABB.
- When I was writing my guide, I was also learning quaternion algebra to enhance my comprehension of inverse kinematics algorithm in robotic, such as the SLERP method.
- I designed several tools for the robot, especially for Pick&Place and Welding applications. I also helped two master thesis students for the calibration of an incremental sheet deforming process.
- I helped another master thesis student to print a metal piece with a WAAM process using a delta printer, a Fronius TPS (welding power inverter) and a welding torch. I programmed for the occasion a first version of my Gcode filtering algorithm.
- I briefly handled the testing of a new robot : Mycobot 280 model from Elephant Robotic and participated in its integration inside Rhino and GrassHopper environment using XML file geometry editing.
- Finally, I worked on a robotized WAAM process, handled the communication between the robot and the Fronius TPS, programmed the Gcode translator final algorithm, made the necessary tests and calibration to find suitable parameters and printed some steel pieces.

In the next parts I'm going to detail my work on these various assignments, explain the knowledge and methods I learned by working on them and describe the solutions I deployed to solve the issues of each work.

3 Apprenticeship of the ABB-IRB-4600-205-45 6 DOF robotic arm

In this section, I will briefly explain the architecture of the industrial robot arm I used during my internship. I will also describe my learning journey towards the mastery of this robotic system and will specifically detail some simple calibration and signal configuration processes.

3.1 The ABB IRB 4600-205-45 industrial robotic arm

Mr. Ruttico gave me the chance to use an ABB robot to work on robotized manufacturing projects. This ABB robot is an ABB IRB 4600-205-45, meaning that it is a robot from the 4600 model series with a two meter and five centimeter action radius range, and a forty-five kilogram maximum payload (end effector mass). This robot is a six degree of freedom robot with a spherical wrist geometry (typical kinematic geometry used for industrial robots where joints 4, 5 and 6 revolute axes are concurrent) allowing an easier kinematic inversion as the first three joints are responsible for the position and the other wrist's joints are used for the orientation. Thus, position and orientation are decoupled between joints 1 to 3 and 4 to 6. For more technical information on the action range, axis range and global appearance of the robot, I put some official ABB scheme and tables of the robot in the annexes section **[7.4]** figures **[1], [2] and [3]**.

When we are referring to a position and an orientation in industrial robotic, we are speaking about the position and orientation of the TCP (tool center point of the robot) which is initially the center of the robot tool flange. I designed two tools during my internship for the ABB robot model I worked with. For each robot, you need to use pieces that match the flange geometry of your robot (generally M10 holes arranged in a circle) and enable you to fix your actuator. Here is the technical drawing of ABB IRB 4600 series tool flange **[5]**.

Every robot (ABB, FANUC, Universal Robotics ...) is linked to a robot controller. In my case, I used an IRC5 single armory controller (you can have dual controller to control two robots and other compact models). A robot controller is basically the brain of the robot and includes a computer with a processor and a breadboard, an axes card, a motor drive unit and a security module. The main breadboard/processor unit is solving invert kinematics algorithm calculus and is sending necessary information to the axes card linked to the motor drive. The axes cart is controlling the motor drive, which has one power output for each motor joints. TRI-phased transformer is converting 380 volts alternative into 24 volts continuous current to supply every robot controller's component with electricity. The security module is using the proprioceptive sensors (position encoders, speed encoders, joint torque sensors, motor current sensors, integrated accelerometers, integrated gyroscopes ...) of each robot joints to detect anomalies such as collision and end range position.

The robot and their controller are bounded together and can't be interchanged. Thus, the hardware and software version of the controller can't be updated in the ABB case at least. With the controller comes a remote controller called Flex Pendant for ABB robots. From the Flex Pendant, the operator can do every actions relative to robot/controller configuration, jogging and programming. The other possibility is to use the RobotStudio software environment, I will present it in the next section of this report.

Finally, the controller armory had an input/output unit as well as various connection ports. This is necessary to create your own signals, whether to collect information on the robotic device or to send signals to a tool or an external axis depending on the robot position, orientation or state. The robot is also designed to have connections port on the third axis to avoid cable management problems that can become a nightmare when an application needs a lot of rotations or significant movement.

You can find some additional scheme of the IRC5 controller in the annexes section **[7.4]** figures **[6]**, **[7]**.

3.2 Robot mastery path during my internship

During my work period at IndexLab I had several missions that granted me little by little necessary knowledge and skills to accurately control the ABB robot.

First, in my first internship month, I had the pleasure of animating a high school class aiming the discovery of industrial robotic technology directed by PhD student Mr. Pini. For three sessions I tested Mr. Pini programs (I was at that time working on my own GrassHopper robot simulation but was not directly programming the robot), calibrated the environment of the robot but not the robot in itself (I discovered later the proper calibration processes) and designed necessary tools for the applications we were showing to the students. This first experience enabled me to master the jogging part of the robot, push me to learn proper calibrating methods, and piqued my curiosity for RobotStudio. Indeed, Mr. Pini was teaching students how to program the ABB robot with RobotStudio and was himself programming the industrial applications with we were showing with it.

We presented a Pick&Place simulation where the robot was placing wood blocks on a table. We decided to make the displacement of the blocks in a cycle (start and end position of the blocks on the table at the same position) to decrease the number of calibration we needed to do. After that, we did a labyrinth where the robot needed to go through with the gripper tool we used for the Pick&Place application. Finally, the last application consisted in the placing of wood blocks and a welding mimic action to join the blocks.

After this assignment I finished the GrassHopper robotic simulation I was working with but found some limitations of the GrassHopper robot plugins. Indeed, the RAPID program output was not optimized and difficult to understand. Moreover, it was not a perfect digital twin of the real system, and I couldn't simulate every singularity I could face during the tool path. Thus, I decided to learn RobotStudio to integrate my GrassHopper simulations in it and combine both softwares advantages. Like this, I learned a consequent amount of knowledge on industrial robotics and the ABB robot I was working with. On Mr. Ruttico demand, I wrote a guide on GrassHopper to RobotStudio simulation as well as robot jogging and calibration. At this point I knew how to properly calibrate the robot (TCP and coordinate systems), how to create robot targets point relative to multiple coordinate systems, how to code basic RAPID programs, how to configure signals and how to solve numerous little issues you can face when controlling and programming an ABB robot.

At the end of my guide writing, some master thesis student working on a robotized incremental sheet deforming process needed some help to calibrate the ABB robot and make their program work. It was an opportunity for me to jump from theoretical knowledge to practice. I completed the tool calibration process that was in fact not really necessary as the tool they were using had a well known geometry, taught them how to do a work object coordinate system calibration and advise them to switch from joint movement to linear movement and to add some zone data to avoid some errors. You can find some images of this process in the annexes section **[7.11]** figures **[49]**, **[50]**.

For the first metal sheet, they were unfortunately facing collisions detection problem that could be avoided by deactivating it. However, we didn't know if it could damage the system and Mr. Ruttico decided to change the metal sheet properties. They are now working with thinner aluminum sheet that might be easier to deform.

I finally finished my learning journey of the ABB industrial robotic arm with my work on a robotized WAAM process. For this occasion I did dozens of calibrations, simulations, real tests and metal printing. I learned specific industrial robot programming tricks to optimize my process, I will explain them in the section dedicated to the WAAM process.

3.3 Calibration processes

ABB calibration processes are easy to use and very straightforward. To have an accurate and usable robot, you need to pay attention to the joint motors calibration and synchronization, the TCP position and orientation, coordinate system position and orientation and tool mass and inertia (only known never used).

When you are switching on the robot and the controller, you need to update the joint rotation counters before launching any program. Each joint is using a sophisticated gear system with microscopic play to adapt the motor speed and respect a TCP position maximum accuracy of 0.01 millimeter. You need to put the robot in synchronization position (corresponding to 0° on every axis) by using the joint marks that are at the precision of one motor rotation (see the annexes section **[7.4]** figure **[4]**). If the operator is updating the joint's rotation counter outside the robot synchronization position, he will need to manually released the joint's brakes and manually rotate robot parts. A good practice is to jog the robot back to home/synchronization position before switching off the system.

For TCP calibration, you need to register the TCP position by saving 4 different robot orientation reaching the same point. For this, you need to use a calibration stake and jog the robot in order to make the end of the stake and the desired TCP touch. Using the reorient jogging option will change tool orientation and approximately keep the desired TCP at the same position (you will still need to move it). An ABB algorithm is then calculating the TCP coordinates relative to the default tool TCP (center of tool flange) by using the four given orientations (for this it only needs to take into account joints 4, 5 and 6 given the spherical wrist geometry of the robot).

For the coordinate system calibration, you need to already have an accurately calibrated TCP. This straightforward method consists in jogging the TCP to the origin of the frame, another point on the X axis and finally a point of the Y axis. Only three points are necessary to determine an orthogonal 3D frame as X and Y axis are perpendicular and crossing at the origin point, Z axis is determined with X and Y axis.

Another calibration process I never used can be interesting for applications needing maximum accuracy. To determine the exact end effector mass as well as the inertia matrix, the robot can do specific movement with an unknown tool, save the joints motors current and torques values and calculate what is needed.

For more details about the operating and technical actions required to jog the robot and calibrate, feel free to read my guide "GrassHopper to RobotStudio" linked at the end of the annexes section

[7.13].

3.4 Signal configuration

In order to activate the robot actuator at the right moment during a program, you need to set up a digital output signal and cable your actuator to the I/O unit of the robot controller. Signals must be 24 volts to respect the limits of the controller. The cabling is facilitated in the design of the I/O unit and doesn't need any soldering. When any physical connection is done, the operator needs to map a signal with the used port and configure it to the wanted type (digital/analogic, input/output).

When everything is configured, the operator can test the signal by manually switching the signal state with the remote controller of the robot and see if the actuator or external axis for example is reacting as expected.

4 Learning IndexLab's software tools and RobotStudio

IndexLab is working close to exclusively with Rhino 3D, and its GrassHopper parametric extension to design pieces, artworks, buildings and to program robots. During the first two months of my internships, I learned how to design pieces with Rhino 3D and to create complete robotic simulations with GrasHopper. I enlarged my horizons towards RobotStudio to combine the advantages of a close to perfect reproduction of reality given by RobotStudio and a smooth and fluid tool path programming enabled by parametric GrassHopper algorithms.

4.1 Plastic 3D printing technology

Performing Pick&Place, welding operations and creating a WAAM robotic cell required some 3D printed personalized pieces. 3D printing is a cheap but accurate method that will quickly provide plastic made mechanically strong pieces that can be used to do whatever you decided. IndexLab had two 3D printers I worked with : a Bambulab X1 carbon printer with a .4 millimeter extruding nozzle and an old Prusa MK3 printer I calibrated and repaired for some students. They are both plastic extruding printers, using three rails and three motors to move the tool head in any space position (other printer are using delta geometries and resin processes). Little by little I learned how to repair both of them and understood some tricks to enhance the quality of my prints. For all my pieces, I used filament spools of PLA material, which is the most commonly used material and require an approximate temperature of 170°C to melt.

Designing a piece for 3D printing is a bit more tricky than designing for simulation purpose, for example. You need to pay attention to the tolerance of the machine for screw holes and joints, you also want to think about the printing direction that will improve the print accuracy but also mechanical properties of your final piece. Generally, the slower your print is given the speed and acceleration the printer is using and the nozzle diameter, the more precise you are. Slicing software given by your 3D printer brand are here to discretize a 3D geometry for additive manufacturing purpose and will output a Gcode (usual manufacturing machine language) to print it. A slicing software comes normally with all the necessary options to configure the piece material density (it is not mandatory to have a completely filled piece), create supports to strengthen the bed adhesion of the piece and to make tricky pieces printable ...

Compact 3D printer available on the market have their own limitations in terms of model size you can print in one run, and of course material strength. Moreover, the bed can move on the Z axis but tool head orientation and bed orientation is not used. Slicing algorithm are putting support to print tricky models, which is possible given the fact we are using plastic material (it's easy to remove after the print) but not feasible for metal additive processes.

4.2 Designed and printed pieces

I won't detail every piece designing process but will only explain some interesting issues I faced for some of them. Every piece isometric, top and side views I printed and used during my internships can be seen in the annexes section **[7.5]** figures **[9] to 16**.

4.2.1 Gripping claws for Pick&Place

For robotic application purpose, I designed gripper claws that were fixed on a pneumatic actuator. I made two versions that were both used during the high school industrial robotic class. Each version has a little 1° slop on the gripping sides to anticipate the deformation due to gripping action. Gripping surface is also covered by diamond shaped pikes to firmly grip pieces. Both model were printed with 100% material density. Surprisingly, the second model broke on my first gripping test. I made an error with screw position. Screws head position were the same ones as the first design, way below the gripping point that was generating a moment. Print layers were horizontal and couldn't handle the force. For the final version, we bought long screws that would go through the hole claw. A metal cylinder was now preventing the claw destruction.

4.2.2 Torch collar

For robotic welding purpose, I needed to create a collar that would hold the Fronius welding torch still and could easily be fixed on the robot. Thus, I created two pieces that matches the form of the torch at the handle and trigger position to avoid high temperatures (PLA melting point is light years away from metal melting point). Another piece was meant to be screwed on the robot tool mount and would serve as a snap joint. At first, I wanted to grip directly the handle, but the shape was too difficult to properly mold. I decided to remove the ergonomic handle and accurately measured the easier shapes of the torch to create the final collar. A little compartment inside the collar could receive the trigger, and its cable, that were not any more fixed to the torch core. Initially, this piece was used for a double tool-mount setup, but I used it until the very end for the robotized WAAM process.

4.2.3 Inclining 45° inclining device and simulation purpose models

Finally, on Mr. Ruttico demand, I created an inverted robotic cell configuration where the welding torch was fixed, and the robot was moving the piece bed. Thus, I needed an object to incline by 45° the torch to be perpendicular to the ground. I used a simple geometry that required to print two parts that would be assembled together with a little joint system and four screws.

For simulation purpose, I also modeled the two robotic cells devices I worked with, the tools I created, the pieces I wanted to print ...

4.3 GrassHopper and RobotStudio for industrial robot programming

I used two plugins in GrassHopper to simulate an ABB IRB 4600-45-205 robotic arm (IndexLab ABB robot). The first one is the *Robot* plugin by Vicose, and the second one is the *Robot Components* plugin, which works similarly to the *Robot* plugin but includes additional tools specific to ABB robots.

The main advantage of using GrassHopper is the ability to quickly create robot targets. By creating an algorithm with GrassHopper blocks, you can simultaneously generate hundreds of targets based on a curve, surface, or brep you create in Rhino or GrassHopper. This is a significant advantage compared to RobotStudio which is less intuitive and fast. Moreover, you can directly integrate Rhino 3D models to create your tools, work objects, and robotic cells for robotic simulation. Consequently, I think that using GrassHopper as a first simulation approach is better compared to a strict utilization of RobotStudio.

However, many tutorials advise using RobotStudio after your GrassHopper simulation to utilize the exact numeric twin of your robot and the ABB inverse kinematic algorithm solver. This solver simulates more singularities and is more accurate because it is the same one used on the real robot. In addition, GrassHopper will generate a RAPID program (the programming language used by ABB robots), which will include one line of code for each target (thus will create RAPID programs that have hundreds of lines). This approach is not optimized, readable, or reusable for other students or people who have not read your GrassHopper algorithm but will, ultimately, use the FlexPendant (the remote controller for your ABB robotic arm) to run the RAPID program. So it is for me essential to learn the basics of RAPID programming to restructure your code and provide a comprehensive version inside the real robot.

Finally, it is difficult to simulate tool animations (such as a gripper closing) and work object displacement (such as visualizing the displacement of a block picked and placed) in GrassHopper. Managing work object data and creating targets relative to different orientation frames (we will see this in the tutorial) is also challenging in GrassHopper, especially with the *Robot* plugins (although it can be done more easily with the *Robot Components* plugins). These tasks are often necessary in RobotStudio.

RobotStudio is far slower than GrassHopper to program robot targets and path. Indeed, I will not describe this process here, but you need to create frames manually for every target you want (unless you wants to use advanced features that are similar to GrassHopper but less intuitive). Tool integration is also slower and more detailed. The software globally needs more rigor. However, this rigor is interesting as working with RobotStudio particularly the manual mode will fasten your mastery of the real robot. So it won't be a lost of time at the end.

Knowing all of this, I thought that learning both GrassHopper robotic plugins and RobotStudio would be an interesting combination, and I created a guide for the laboratory and Mr.Ruttico students I linked in the annexes section **[7.13]**.

4.4 Simulation examples

I did two simulations on GrassHopper I then integrated inside RobotStudio for Pick&Place and Pick&Place and welding. Of course, I also did a lot of RobotStudio simulations for WAAM processes. Again, I won't lose you into uninteresting technical things, the detail of GrassHopper algorithms, an example of the raw output code of GrassHopper simulations, some picture of RobotStudio simulations and an example of a polished robot code established form a GrassHopper raw code can be found in the annexes section **[7.7]**, figures **[32] to [42]**.

For each simulation, I programmed the robot in a secured way. Indeed for each repeating pattern, the robot is coming back to known position, speed is being reduced when the robot is approaching a wood piece, pauses has been implemented to wait the actuators to grip or to toggle the electric arc and targets orientations are thought to reduce as much as possible useless rotations. I couldn't find a way to create various coordinate systems for one of the robotic command plugin I used in GrassHopper. Thus, I switched to the robot component plugin in order to create targets relative to meaningful coordinate systems (first block position reference frame for initial block gripping position and table reference frame for table block's releasing positions, for example). For the PickPlaceWeld simulation, I employed a specific tool path for the block soldering part, which is closer to the technical recommendations of welding professionals.

More details about tool integration, calibration and even a detail tutorial of my RobotStudio Pick&Place simulation can be found in my guide **[7.13]**. I will explain my WAAM simulation inside the dedicated WAAM processes part.

5 Quaternions algebra basics

During my second month of internship, I had some material issues : I couldn't work with RobotStudio as my trial license was finished, and laboratory machine were taken by innovative facade project students. Thus, I decided to learn the basics of quaternions algebra from the demonstration of quaternion existence through standard rotation matrix to spherical quaternion interpolation. In this brief part, I'm not going to give a class on quaternion algebra, but I'm going to explain their role in current industrial robotic technologies.

5.1 Quaternion in industrial robotic

- First, quaternions are of course used in the forward and inverse kinematics algorithm of industrial robotic arms. In SYSMER classes, we already worked on inverse kinematics calculus and had to implement one for a PUMA robot in Matlab.
- The orientation precision of the robot TCP is also calculated by using quaternions. Indeed, standard TCP orientation of a robot is given by the unit quaternion [1, 0, 0, 0] and the rotation distance is calculated by a quaternion product between unit quaternion (initial orientation) and second quaternion orientation.
- Moreover, the SLERP method (spherical linear interpolation) is used by the majority of industrial robots to reorient the end effector when following a rectilinear trajectory. The calculated rotation trajectory correspond to the minimal torque trajectory.
- Furthermore, in my case, I was using an ABB robot and an ABB robot operator needs by default to set the orientation of reference frame with quaternion formalism and not in Euler angle formalism
- Finally, for aerial devices and maybe even internal industrial sensors such as gyroscope, can't handle some rotation sequences of specific rotations that can remove captor degrees of freedom and generates bugs that can have severe consequences if Euler angles are being used (this phenomenon is called gimbals lock problem). Quaternion algebra is less intuitive for a human being to use, but far more powerful than Euler angles in this matter. Quaternion based algorithm are easier to compute, simple and stable. Rotations sequences are coded in a four dimension space and are not suffering from rotation axes alignment problems.

Given these facts, I was motivated to learn in detail quaternion algebra when I couldn't efficiently do my other assignments because of material issues.

6 MyCobot 280, a collaborative robot

IndexLab bought a cheap and really tiny robot from Elephant Robotics to give some projects to students. Some of them were programming it for clay printing for their thesis related to a Mars exploration scenario where an industrial robotic arm mounted on a rover would 3D print clay shelter for astronauts or spatial settlers. Mycobot 280 is a collaborative robot (cobot) not based on a spherical wrist geometry. This type of robot geometry, is standard for cobot as it is reducing the number of singularities and thus securing the human working in collaboration with it (you can find some detail of its dimensions and geometry in the annexes section **[7.12]**, figures **[52] and [53]**).

6.1 XML geometry files using Denavit-Hartenberg formalism

I was given the assignment to test the basic features of this little robot and also to think on a way to integrate it inside Rhino and GrassHopper environment. GrassHopper robot plugin is using XML files to code the kinematic geometry of a robot using the Denavit-Hartenberg kinematic formalism. I created my own XML file for this robot and used its datasheet to correctly place each joint. The integration was working to a certain scale. Indeed, the robot plugin used well known robot manufacturers such as ABB, Fanuc, Universal robot ... Elephant robotics was not registered as a manufacturer, and I couldn't directly create a GrassHopper algorithm with the correct MyCobot280 robotic language output (it required language translation). The thesis student created their own method to translate Universal Robot code and programmed the cobot with GrassHopper.

Consequently, I learned about Denavit-Hartenberg kinematic formalism, which is using D-H parameters : d, a, θ and α. Each reference frame position and orientation is coded relatively to the previous one.

- d is the offset along previous joint frame z-axis.
- a is the offset along previous joint frame x-axis.
- θ is the rotation angle about the previous z-axis between the previous x-axis and the new one.
- α is the rotation angle about the previous x-axis between the previous z-axis and the new one.

Robot plugins are extracting D-H parameters from geometry XML file to do the necessary invert kinematic calculus in order to move the robot according to the constructed GrassHopper algorithm.

7 WAAM process and robotic cell creation

I experimented WAAM (Wire Arc Additive Manufacturing) with a first already existing and functional IndexLab's set up constituted of a WASP delta 3D printer, transformed to take the welding torch as an extruding head and to send arc on signal to the Fronius TPS which is the power unit of the welding torch. I then created my own robotized WAAM set up with the ABB industrial robotic arm I described in a previous section.

7.1 Welding basics

They are dozens of welding processes for different types of metal and applications, but they all have some common points. Indeed, welding consists in the deposition of melted metal drops on a based metal that is also reaching fusion temperature during the weld. Every welding types except new laser welding technologies are using electrodes or electric arc to cause metal fusion. The metal rod used for welding is covered by a coating flow that is enhancing the metal deposition, helps to clean the deposit and prevent oxidation reaction. Each method is using another shielding flux that is usually a gas but can be liquid to protect the melted deposit from the atmosphere gas components to ensure mechanical properties of the deposition. In my case, I was using argon to shield my welding.

The welding post, Iused was the Fronius TPS 320 i, and is a power electronics unit that is delivering high intensity and voltage. When the current increase, the welding temperature is increasing and thus the deposition flow as well. Consequently, it is possible to adjust every welding parameters (mostly current, tension and filament feeding rate) to vary the deposition properties. Technologies have progressed in the welding sector, and new welding posts such as the one I used are now digitized and microprocessor controlled. During the welding, data are being collected and measured continuously to respond immediately to any changes. The rod filament position, for example, is regulated at the same distance from the torch end at any time during the process.

The metal I used during my internship was steel, as well as a really standard welding process called MIG (metal inert gas) to perform WAAM.

7.2 WASP WAAM process

A master thesis student needed to print a specific topologically optimized piece in steel that enable to screw two wood planks together. I was interested in helping her as it would be a first opportunity to find good parameters for WAAM and widen my understanding of the welding post. You can find some pictures of the piece and the process in the annexes section**[18]**, **[19]**, **[20]**.

7.2.1 Slicing strategy

The WASP printer was using Gcode as for every other 3D printer. Consequently, I decided to post process the code outputs of an existing and open source slicing software used for Prusa 3D printers for different reason :

- I had a close deadline for this assignment as the master thesis deliberation was close.

- Slicing software were unknown to me and creating a custom one would have required a lot of time, I preferred to spend mine on the robotic part of my internship.
- Prusa slicer is open source and complete. It comes with hundreds of options I could change to match my WAAM application.
- I previously did a lot of 3D printing and exactly knew how was structured the output Gcode file. Indeed, for each layer and sequence of Gcode, Pruse slicer was putting comments, my post-processing algorithm could be simplified with this characteristic.

7.2.2 First version of my post-processing algorithm

My first functional post-processing made in Python code was reading a selected user Gcode file and filtering comments and G1 commands (I only needed linear movement command). I was then putting G1 sequences in Gcode sequence list based on the commentaries given by Prusa slicing software : layer change, infill, external perimeter, internal perimeter ... Finally, the algorithm was extracting and changing G1 command parameters (X, Y, Z for position, F for speed, E for feed rate) as I wanted. My code was automatically opening and saving input and output files. It was necessary to have a quick process in order to do the maximum amount of welding parameters test possible. You can find my python post-processing code in my GitHub deposit **[7.13]**.

7.2.3 Testing procedure and final result

WAAM process has five parameters to play with : current, feed rate, tool path, tool speed and material. In the Fronius case, metal feed rate, current and tension were regulated and linked. Moreover, the material was already chosen, I could play with current, tool path and tool speed.

I could easily control the tool path with the Prusa slicer software and adjust it to my needs. Tool speed could be easily changed with my post-processing algorithm, and current could be adjusted with the welding post. I conducted batteries of test to optimize each parameter. I was fixing to of them to find the other. When I found a nice deposition current, I jumped to the tool path parameter and then to the tool speed.

The difficulty of this piece was to find good infill parameters. Indeed, the piece was only three millimeter thick, and I was not risking defect propagation compared to my robotized WAAM objectives.

The final result could still be optimized and had some defects, but dimensions were respected. To give you an idea, this print took thirty minutes of continuous welding and two layers of one point five millimeter height to print. You can find pictures of it in the **[7.6.1]**, figure **[20]**. Compared to the robotized process, I had no simulation environment to visualize and control in advance my printing results, but the calibration was much easier.

7.3 Robotized WAAM process

Before my work, nothing was made in the laboratory for robotized WAAM (the ABB robot had no WAAM option) and I found everything by myself. I needed to find a way to connect the welding post and the I/O unit of the robot to at least send the arc toggle signal at the right moment and position.

7.3.1 Welding post and robot controller connection

After some data sheet reading, I found how the AI/IO V2 (the I/O unit of the welding post) component could be linked to the robot controller and set up from the welding post local server. I could change the type of information exchanged for each input and output. In my case, I could change four Fronius input (thus four ABB robot digital output signals could be created). I adjusted the signals so that IN1 was the arc toggle signal and IN2 to 4 job number bits state signal. Indeed, with such signals, I could change a bit state of the welding job binary number of the welding post to change welding job parameters preset during a WAAM process. This was the key for variable layer height metal printing, for example. With this interesting options, I could save eight job parameter presets and switch between all of them during a print. After configuring the IRC5 controller, the necessary connection between the Fronius welding post and the robot controller was done.

7.3.2 Test bench creation to final robotic cell assembly

IndexLab had some (but not enough) Bosch aluminum profiles to work with. I manage to assemble a first test bench, with the pieces I could find, you can see it in the annexes section **[7.6.2]** figure **[21]**. I knew that the with a square tool mount, I could have a suitable robot configuration to reach the torch end of filament position. The bench was functional, and I even started some test with it, but it was absolutely not secure, and we decided to create something more professional to work with.

Thanks to the help of Mr.Bordoni we moved and fixed the robot to a new position, assembled and adapted a protective glass cell used for a previous IndeLab project. With this product we could fix the torch in a higher and more comfortable position, bring the air suction system above the torch and put some anti lightning radiation curtain. The final robotic cell was done.

7.3.3 Robot's WAAM tool

For the robotized process, I used little steel disks of fifty and thirty millimeter radius as a welding bed. The tool consisted of a square aluminum piece where I could screw the disk on and a wood plate which was isolating the robot from high temperature and welding current. A more professional tool could have been done if we had a specific polymer material in the lab. Of course, connections used for the square piece and screws for the tool flange were separated to prevent any electric conduction. Overall, the device was rustic but correctly leveled and easy to use, I had some good result with it. Sometimes you don't require incredibly sophisticated materials to do the job.

7.3.4 Final version of the post-processing algorithm

As for the first WAAM process, I kept my strategy of regular 3D printer slicer post-processing. This time I had to translate the Gcode file output into a RAPID program module. This was not really difficult, as I only needed to convert G1 command into MoveL RAPID instructions. The biggest issue I faced was the length of Gcode file. Indeed, Gcode is giving, one command for each discretized point of a sliced geometry and I often had Gcode files with more than fifty thousand lines. However, RobotStudio and the IRC5 controller couldn't handle that many lines in one module (robot program). I did some functions to compress easy repetitive shapes such as cylinder and cones by using only the first layer point and some loops, but for complex shapes I couldn't find an intelligent way to reduce the output program size. Thus, I decided to split the output program into several RAPID programs of twenty thousands line of code. Given the fact that RobotStudio handled a limited amount of module, I was copy and pasting my programs in my printing module each time I finished one.

In the annexes section **[7.8]**, figure **[45]**, you can find an example of a simulation for a complex shape that required seven programs of twenty thousands lines of code. Each color correspond two one file. It is the result of the TCP trace of the robot and is giving the printed piece upside down. I was using RobotStudio for this incredibly useful visualization tool. I was also switching the TCP trace depending on the state of the arc toggle signal to make a difference between printing movements and repositioning movements.

You can find my post-processing python algorithm inside my GitHub page I linked in the annexes section **[7.13]**.

7.3.5 Robotized inverted configuration WAAM process functioning

The process had a particularity, the welding torch was fixed, and the robot was moving the piece bed and the piece during the metal printing. This had some serious advantages compared to the usual robotic WAAM process. Indeed, IndexLab had no external axes such has a rotating bed to give more degree of freedom. This method is enabling the robot to angle the piece during the print and thus to print really complex shapes impossible to print without support even for a plastic 3D printer. My post-processing strategy was a limitation for dynamic piece orientation during a print, but I could still do some static reorientation, as you will see at the end of this section.

Here are some technical details about my process : the robot was executing the 3D plastic printer tool head movement (given by the Prusa slicing software) in the torch coordinate system. However, it is the torch that actually deposits melted steel on the bed. Thus, the deposition is offset from the distance between the tool head and the origin of the printer, which in our case is the torch work object plane. Consequently, the TCP trace of the robot in RobotStudio is the offset and upside-down version of the slicing. As you can see in the **[7.8]**, figure **[44]**.

To calibrate the robot TCP, I used the torch endpoint as a fixed calibrating stake and used the orient and linear jogging mode of the robot to perform the ABB calibration process. After calibrating the TCP, I could calibrate the torch reference frame. For this, I placed the center of the WAAM disk at the endpoint of the torch. Then, I would redefine the torch reference frame values by jogging the robot in linear mode near the torch endpoint in the previously defined torch coordinate system to check if everything was fine (zero of Z axis, extremity of the disk welding bed at fifty millimeters from the origin of the coordinate system for a fifty millimeter radius disk bed ...). With this protocol, I produced centered pieces and printed pieces with a one millimeter tolerance to prevent any welding outside the bed area.

7.3.6 Test procedure and optimizations

As for the previous set-up, I did dozens of test to optimize welding parameters to be able to print reel 3D steel shapes. I reused the speed parameter I found from the WASP WAAM process but changed the current parameter as well as the tool path because I was not using infill anymore.

I wanted to print 3D simple shapes such as tubes and cone. The difficult part was to find good start and end of welding parameters. Indeed, arc toggle is generating the majority of the defects. Moreover, defects are propagating layer by layer and a tiny default can quickly produce massive print failures. Thus, I played with different tricks : first, I used a Prusa slicer option to randomize the position of each welding start for each layer. This was preventing default propagation by spreading them in every possible places. Then, I started the first arc outside the shape to create a defect outside the print. Finally, I played with welding post options to decrease the current at the end of the deposition with a slop coefficient parameter.

I was also using two different welding parameters presets between the first layer (directly on the bed) and the other layers. Indeed, the first layer need a slightly higher current for a correct deposition.

I encountered another issue. Indeed, to prevent robot vibration and guarantee the desired speed, I needed to use a zone data parameter on my movement instruction (I used the smallest predefined RobotStudio zone data of 0.3 millimeter radius) this was causing troubles as the controller is anticipating three instructions in advance when using zone parameters to create smoother tool path. Consequently, the robot was switching digital output states such as the arc toggle before reaching the desire position. Thus, I was joggling between zone data movement and maximum accuracy movements before each signal update. I also used some specific type of pauses to enhance my process.

7.3.7 Final results

Tests and final print were done in the last two weeks of the internship. Moreover, I had problem of argon gas shortage and used approximately five days to print everything. If you are curious, you can see a welding test I realized without shielding gas **[7.6.2]**, figure **[25]**.

I printed four pieces, chronologically : the big cylinder, the little cylinder with fliers, the stone stand and the cone. I found the finalized my parameter research after the big cylinder print.

The big cylinder print faced defect propagations issues. Each of my pieces except the stand were five millimeter thick. Each perimeter was constituted of two welded concentric circles with a two point five millimeter gap between them.

My most interesting print was the little cylinder with four fliers on it. Indeed, it is the demonstration of the inverted configuration utility. I first printed the little cylinder and then rotated it by 90° to print fliers shapes on it. This piece is impossible to print without rotations and impossible to print without support on a plastic 3D printer. Moreover, it had practically zero defects. I am very proud of this piece.

The stand and cone shape were experimentation. For the stand, I was trying to push the accuracy limit of my process by printing something really tiny. The final product fills its function, but it is not respecting the initial 3D model. The cone shape was meant to fail as I was running out of gas. I wanted to know if I could print an object with slopes without dynamic piece reorientation. It is working for a tiny slope, but it requires repeating the experience to be completely fixed on this matter.

Each of these pieces required some polishing to clean and smooth the surface. The longest print was the big cylinder and lasted forty-five minutes. Of course, I have taken pictures of these pieces that can be found in the annexes section **[7.6.2]**, figures **[21]** to **[31]**.

Conclusion

During this internship at IndexLab I learned various knowledge, methods and software tools for general robotic, industrial robotic, additive and welding manufacturing processes. I am now proficient with industrial robotic arms, especially the ABB ones, and can use them in various manufacturing applications such as Pick&Place, incremental sheet deforming and mostly WAAM and welding.

This work period at IndexLab was thrilling, and I'm proud of the functional WAAM process I manage to build. The parameter optimization, slicing strategy reflection and calibration processes were really instructive for me. Overall, this internship was a great experience for me, and I am more competent in my studying field.

If I had more time at IndexLab, I would have improved the robotized WAAM process by creating a functional custom GrassHopper slicing algorithm to make dynamic reorientation of the piece during its printing. It would have greatly widened the print possibilities and open the door to complex shapes and angled pieces. Variable layer height design is also a feature to develop to enhance the quality of printed shapes and their complexity. With more gas, some more ambitious and crazy project could have been realized even with the current state of the process. In addition, a combined WAAM technique and robotized polishing post-process could be a consequent step towards a fully automated manufacturing metal printing method.

To sum it up, a lot of improvements are yet to come for this functioning but really basic and primitive WAAM robotic cell I created in IndexLab. WAAM is an incredible technology with great potential that is increasingly important in the manufacturing industry. Moreover, laser welding technologies will further enhance metal additive methods that will deliver astonishing pieces in the future and might enable the mass production of topologically optimized metal products.

Annexes

7.4 ABB IRB-4600-205-45

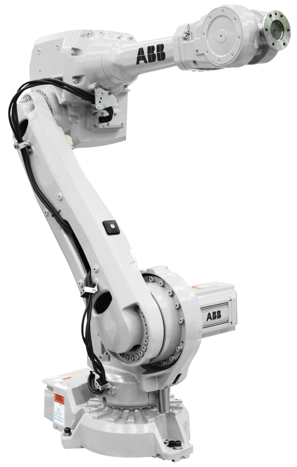
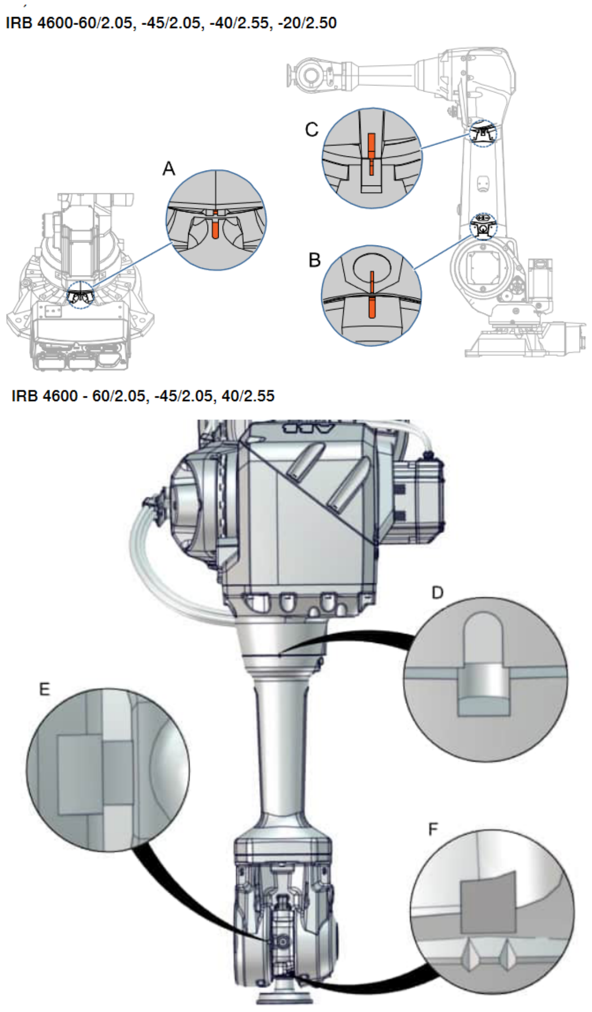
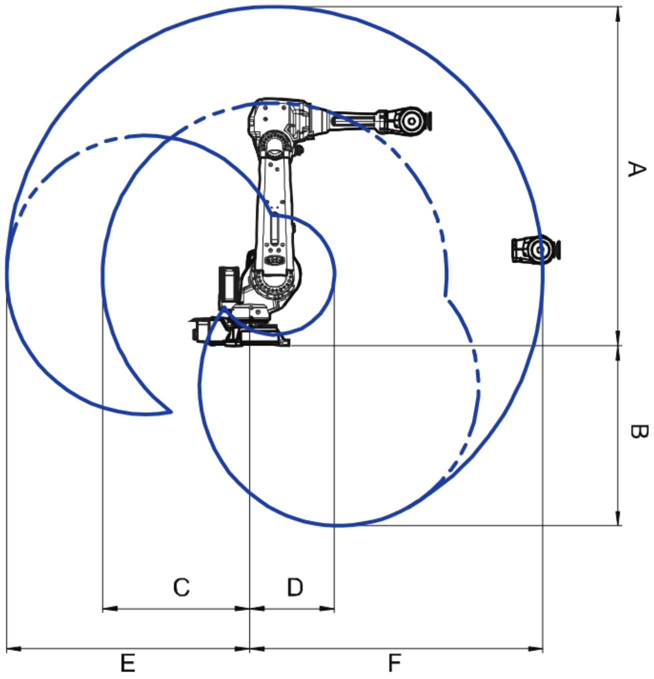


Fig. 1 – ABB IRB-4600-205-45


IRB 4600-60/2.05, -45/2.05, -40/2.55, -20/2.50

IRB 4600 - 60/2.05, -45/2.05, 40/2.55

Fig. 4 – ABB IRB-4600-205-45 synchronization marks



Variant	Pos. A	Pos. B	Pos. C	Pos. D	Pos. E	Pos. F
IRB 4600 - 60/2.05	2371 mm	1260 mm	1028 mm	593 mm	1701 mm	2051 mm
IRB 4600 - 45/2.05	2371 mm	1260 mm	1028 mm	593 mm	1701 mm	2051 mm
IRB 4600 - 40/2.55	2872 mm	1735 mm	1393 mm	680 mm	2202 mm	2552 mm
IRB 4600 - 20/2.50	2833 mm	1696 mm	1361 mm	665 mm	2163 mm	2513 mm

Fig. 2 – ABB IRB-4600-205-45 range

Location of motion	Type of motion	Range of movement
Axis 1	Rotation motion	±180° ±165° (Clean Room)
Axis 2	Arm motion	+150° / -90°
Axis 3	Arm motion	+75° / -180°
Axis 4	Wrist motion	±400°
Axis 5	Bend motion	+120° / -125°
Axis 6	Turn motion	±400°

Fig. 3 – ABB IRB-4600-205-45 axes range

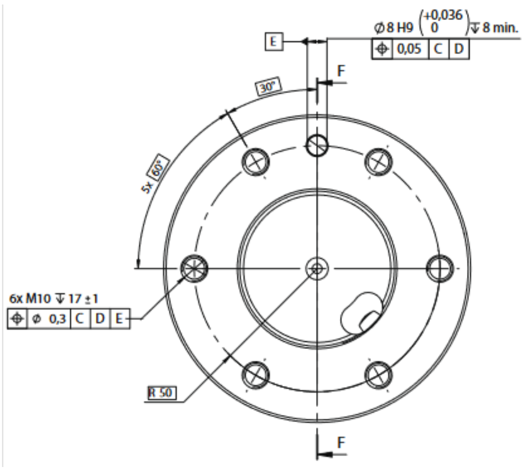
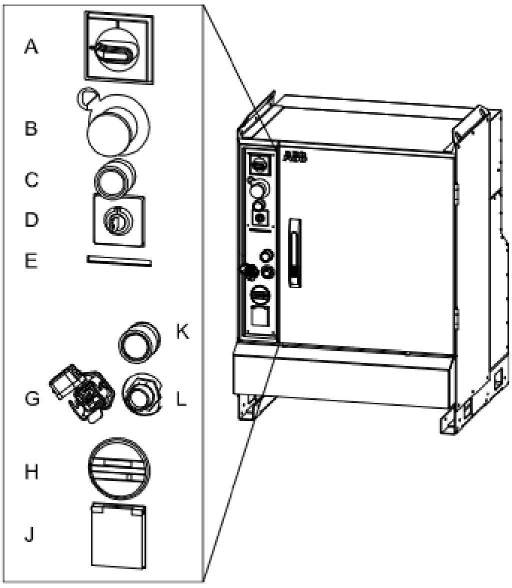


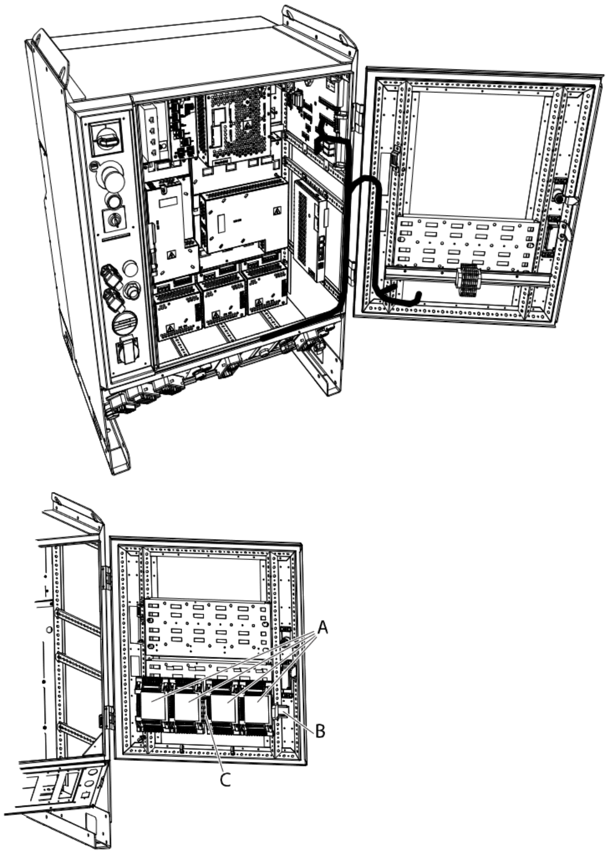
Fig. 5 – ABB IRB-4600-205-45 tool flange



xx0600002782

A	Main switch
B	Emergency stop
C	Motors on
D	Mode switch
E	Safety chain LEDs (option)
G	Service port for PC (option)
H	Duty time counter (option)
J	Service outlet 115/230 V, 200 W (option)
K	Hot plug button (option)
L	Connector for FlexPendant or T10

Fig. 6 – IRC5 controller



xx0500001859

A	I/O or encoder units
B	Mounting rail
C	Connection terminal XT31

Fig. 7 – IO unit and inside view of the controller

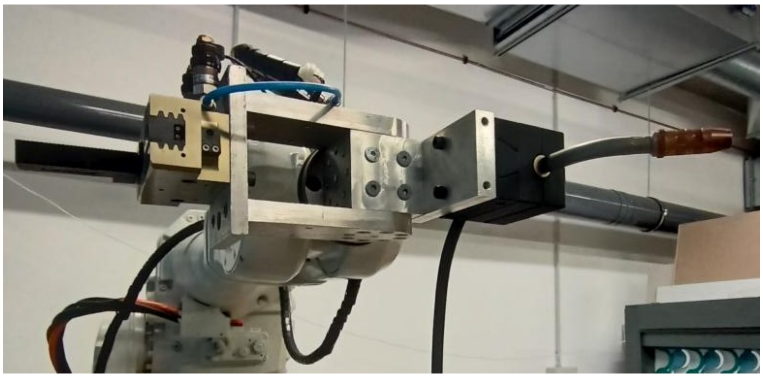


Fig. 8 – Double tool mount installed

7.5 3D models

7.5.1 Tools

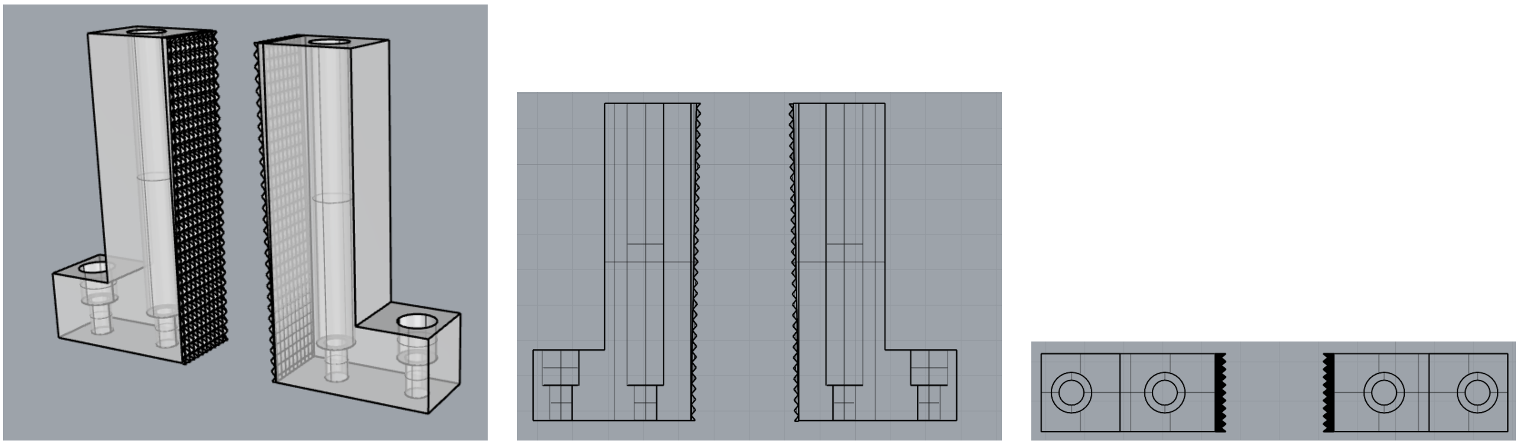


Fig. 9 – Gripper claws first version

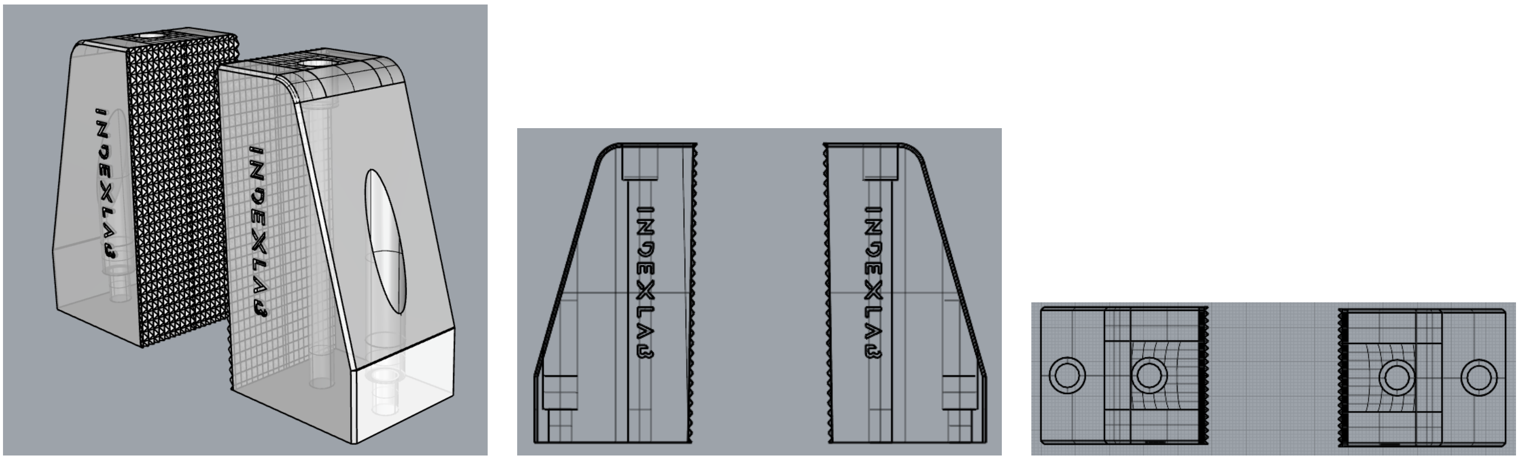


Fig. 10 – Gripper claws final version

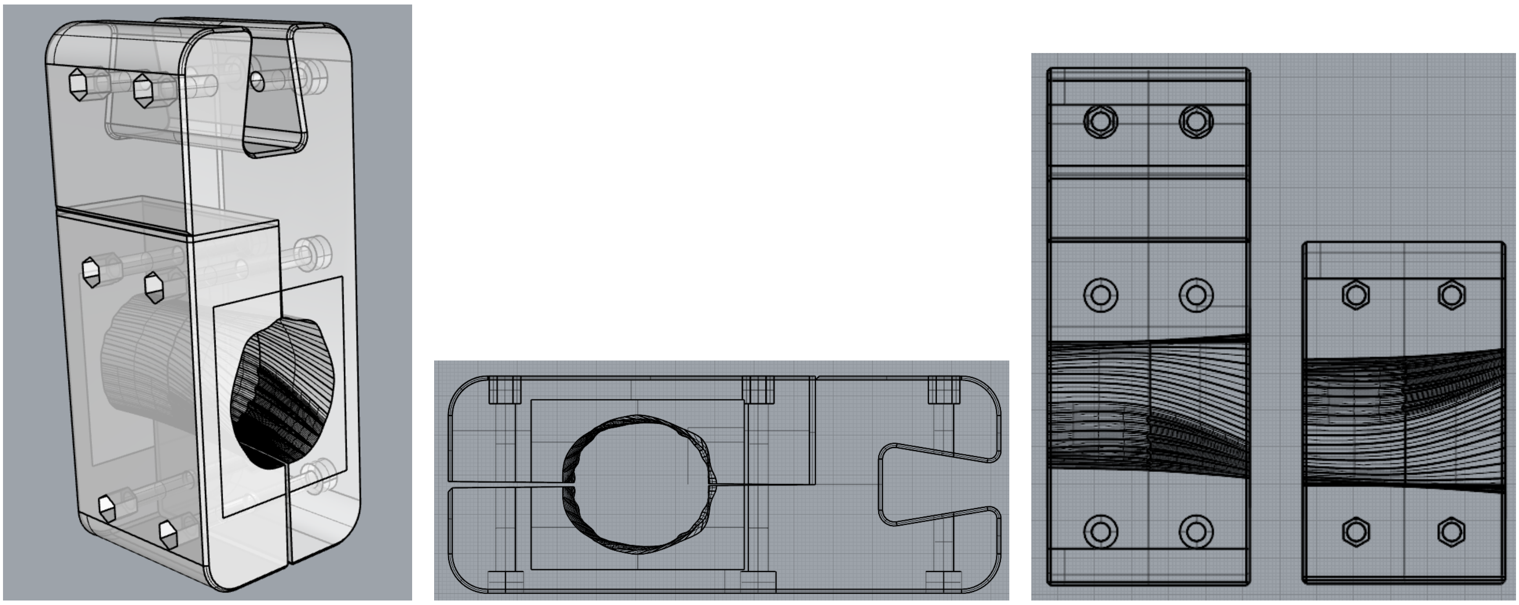


Fig. 11 – Torch collar first concept

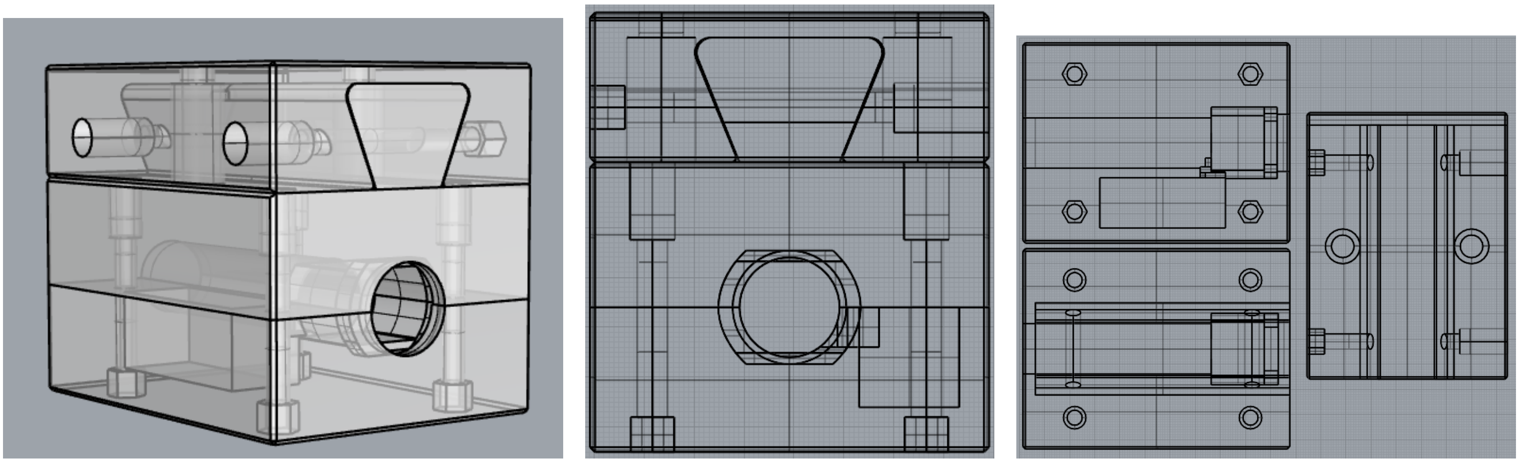


Fig. 12 – Torch collar final version

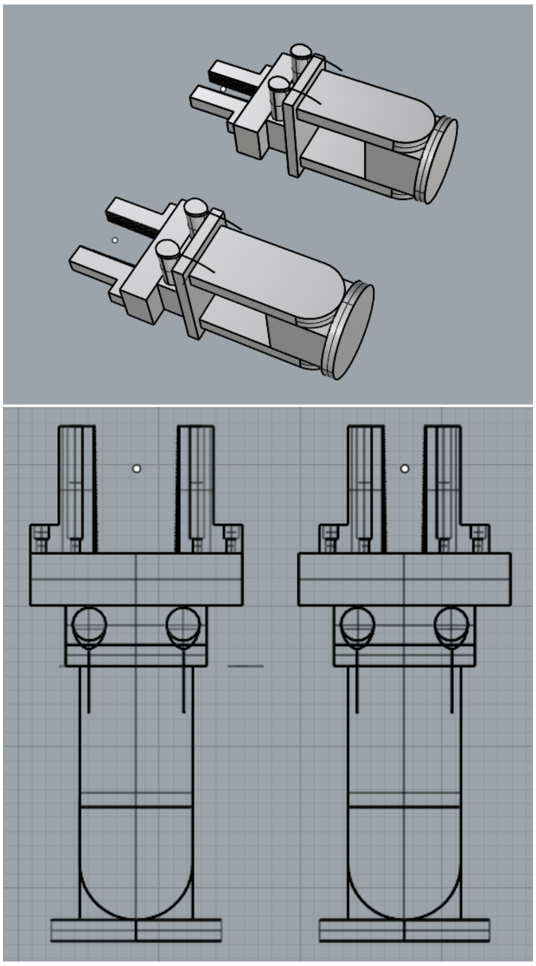


Fig. 13 – Gripper tool

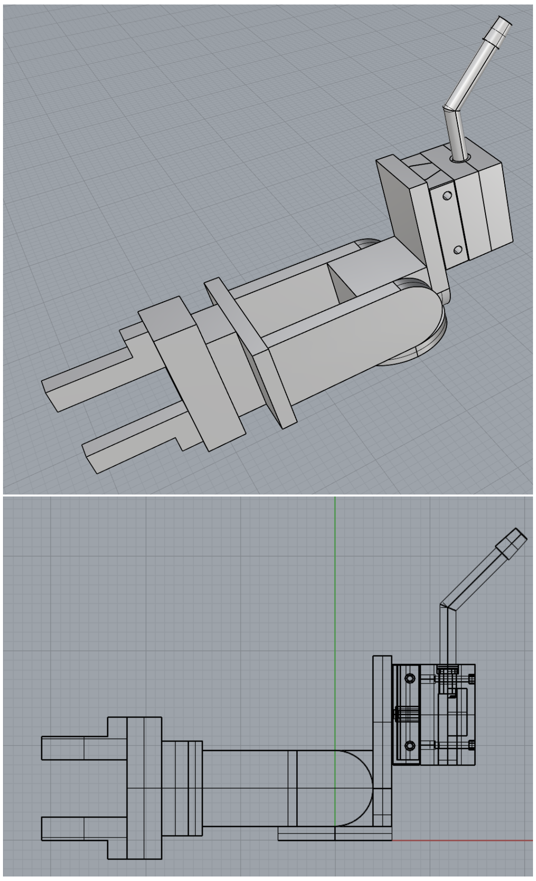


Fig. 14 – Double tool mount

7.5.2 WAAM

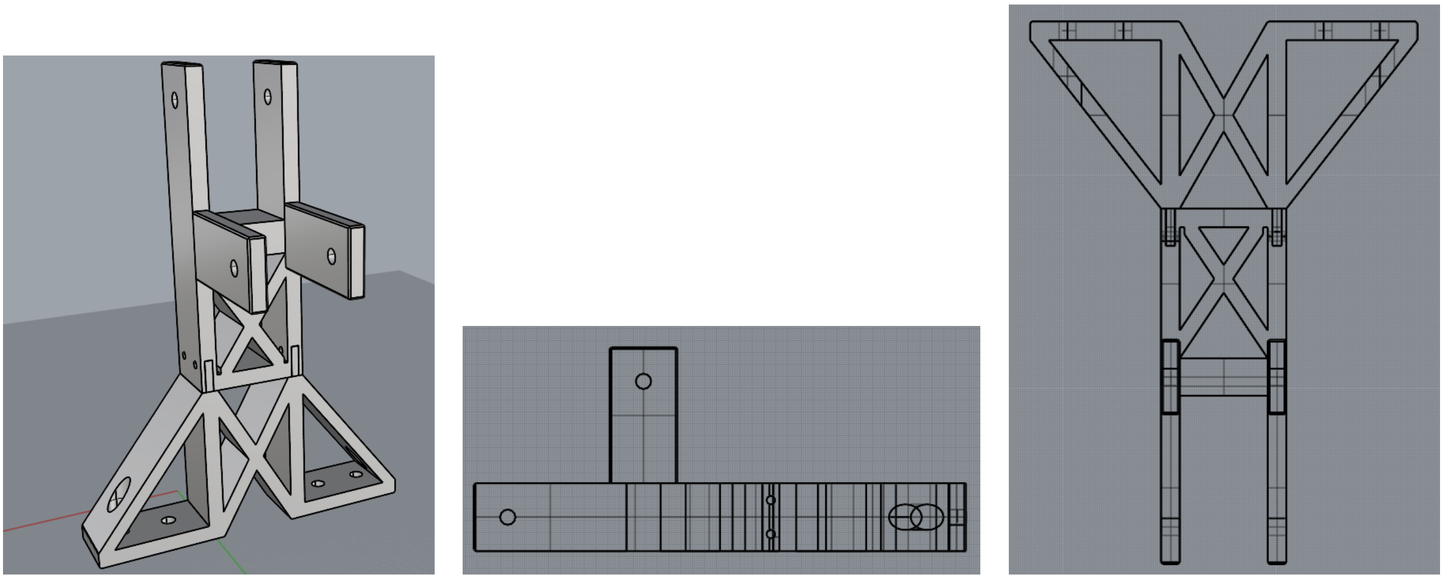


Fig. 15 – Torch inclining dispositiv

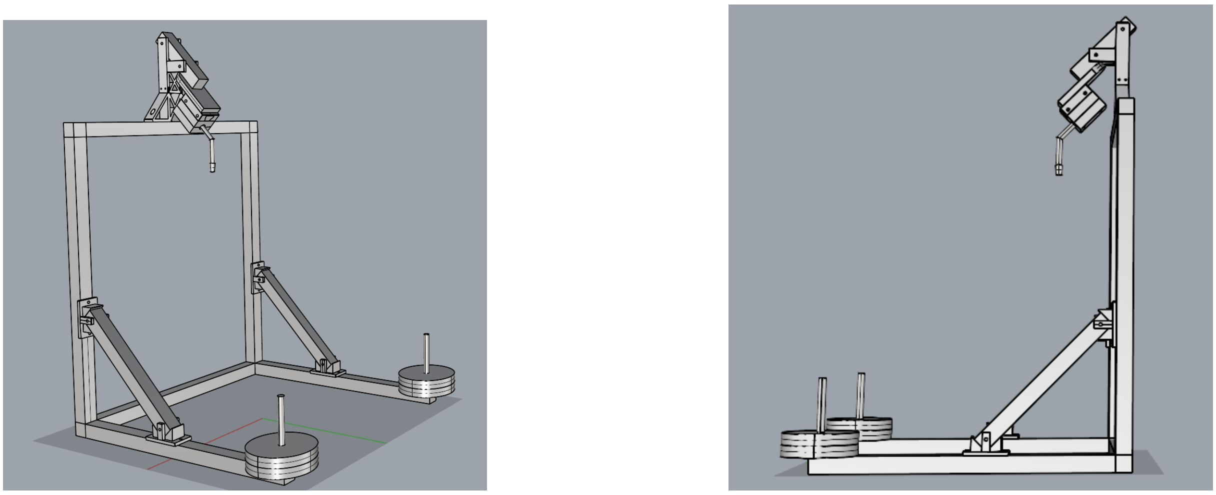


Fig. 17 – WAAM dispositiv concept

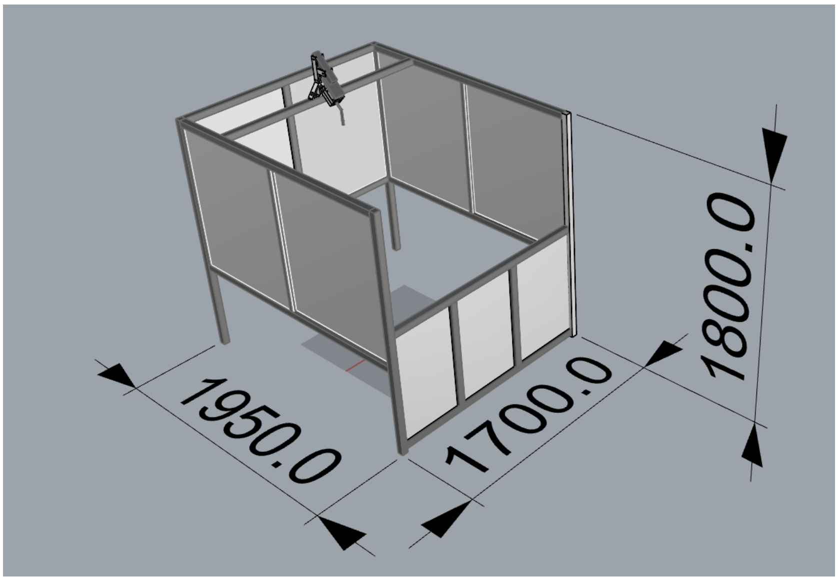


Fig. 16 – Final WAAM dispositiv

7.6 WAAM process

7.6.1 WASP WAAM process

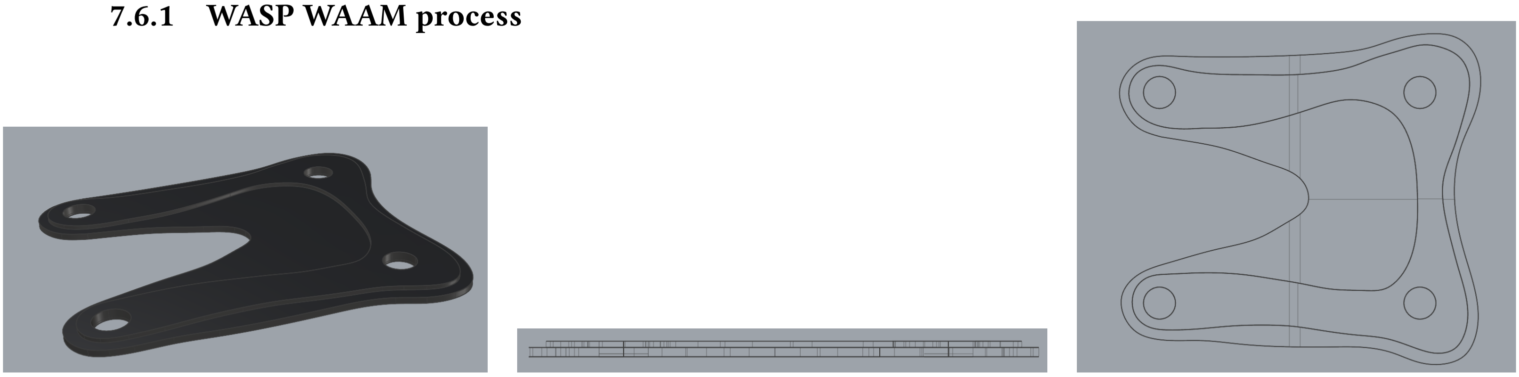


Fig. 18 – Thesis piece 3D model

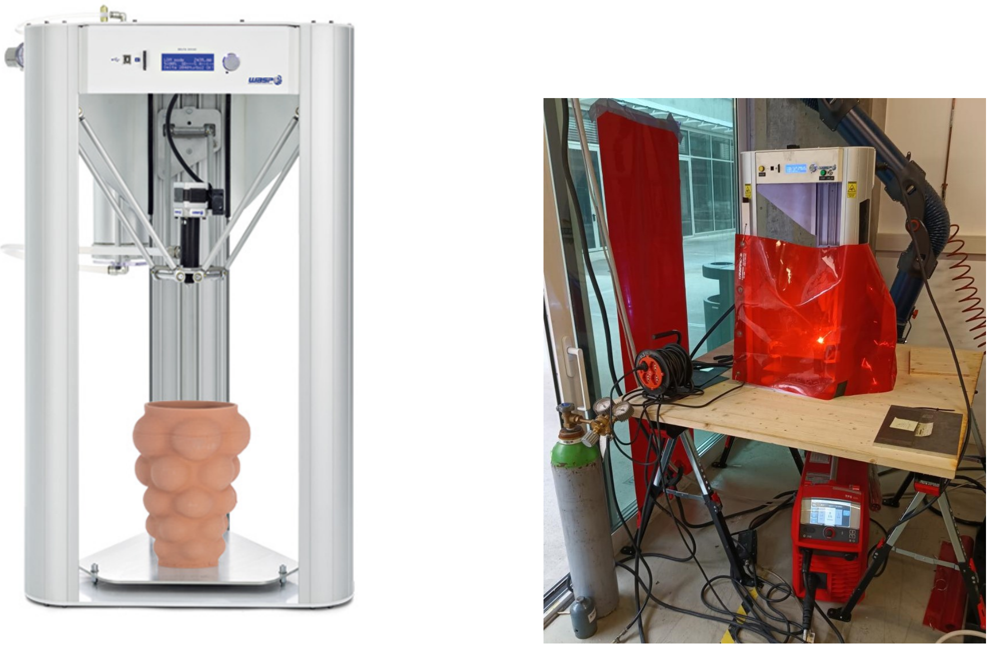


Fig. 19 – WAAM setup with a WASP delta printer

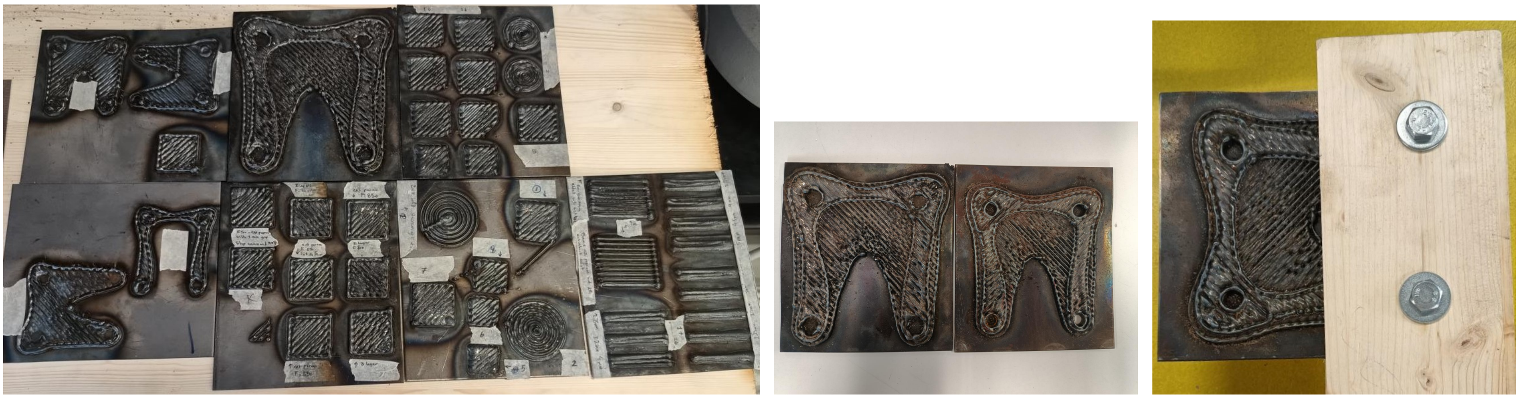


Fig. 20 – WAAM WASP dispositive printed pieces

7.6.2 WASP robotized process

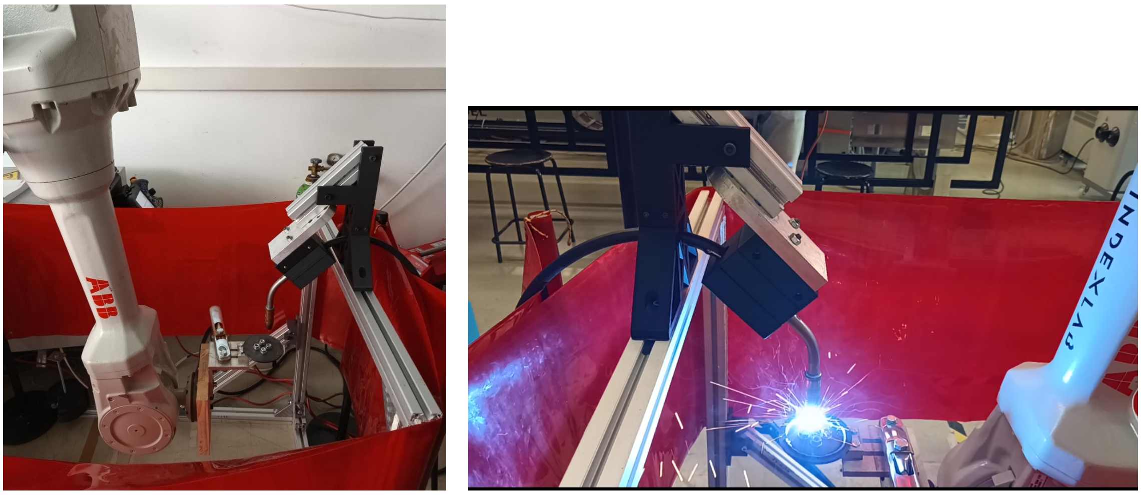


Fig. 21 – First setup of robotized WAAM

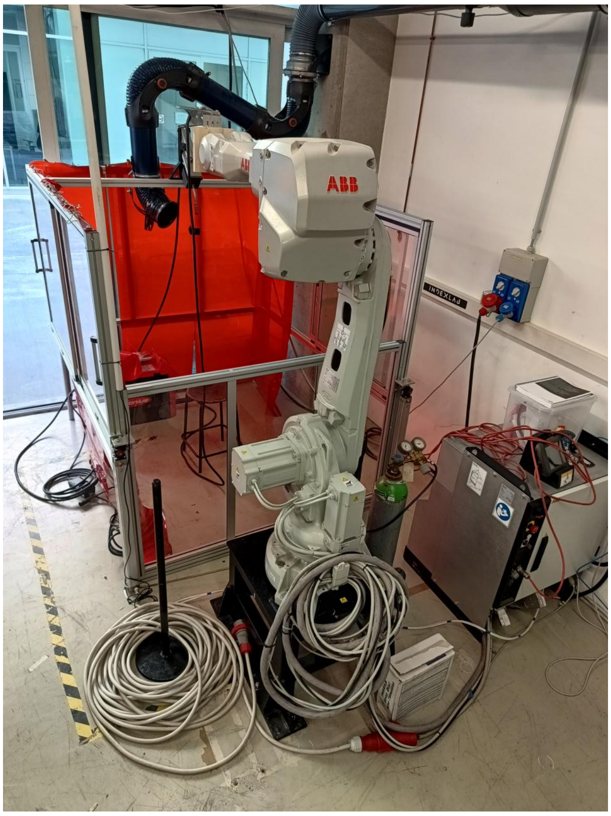
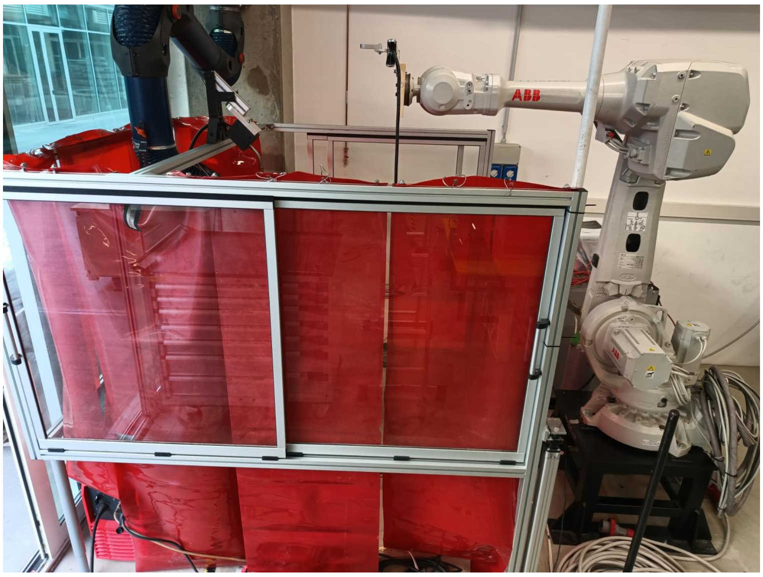


Fig. 22 – WAAM final robotic cell



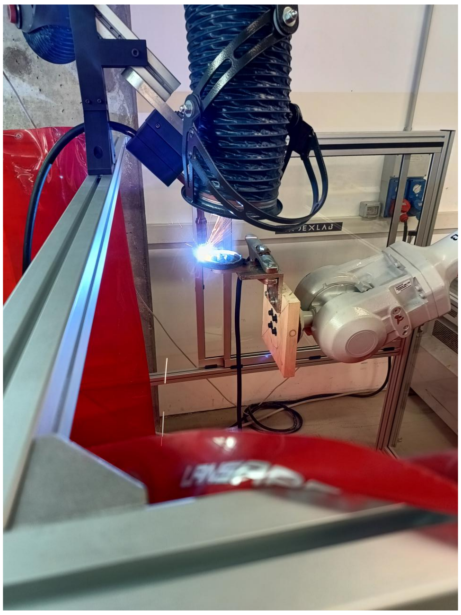


Fig. 23 – Robotized WAAM process in action

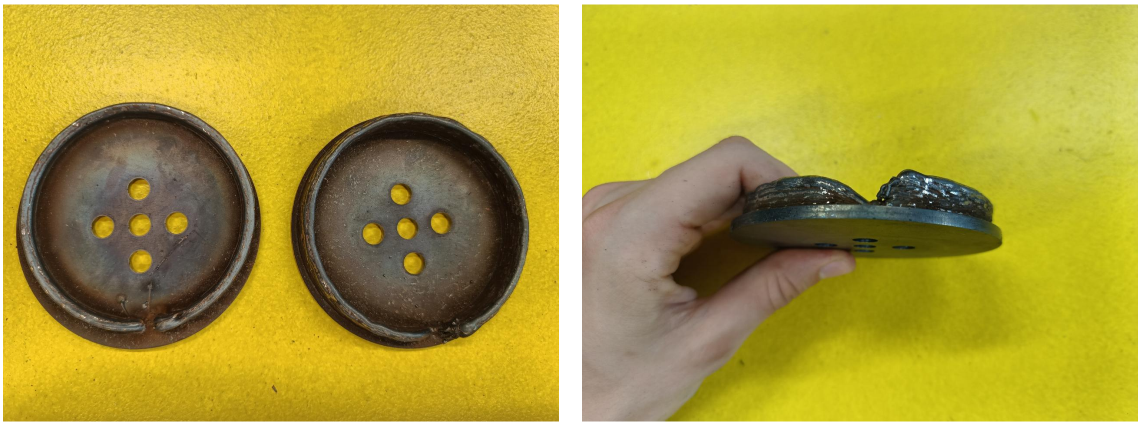


Fig. 24 – Tests and defect propagation

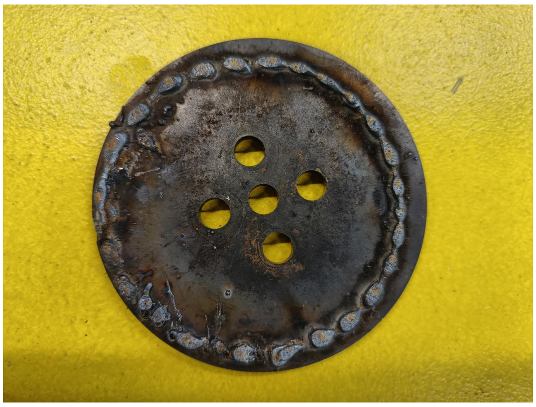


Fig. 25 – No gas test

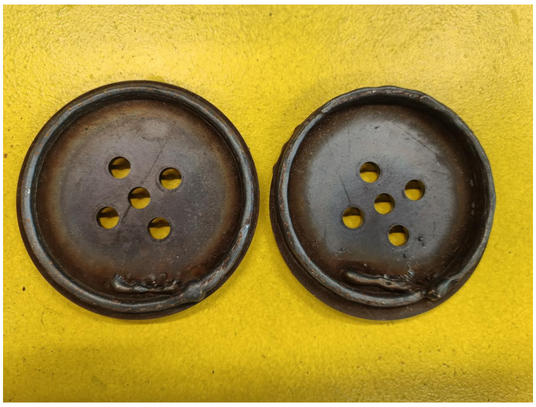


Fig. 26 – Optimization strategy

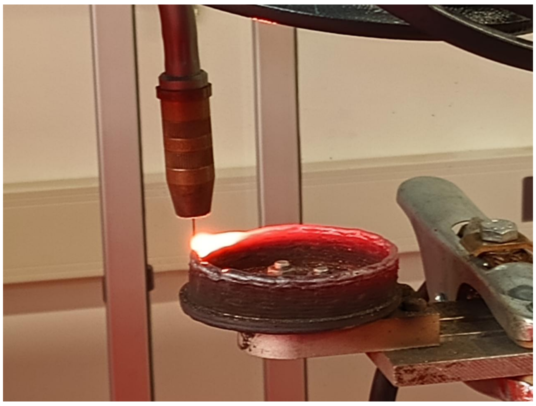


Fig. 27 – Big cylinder print

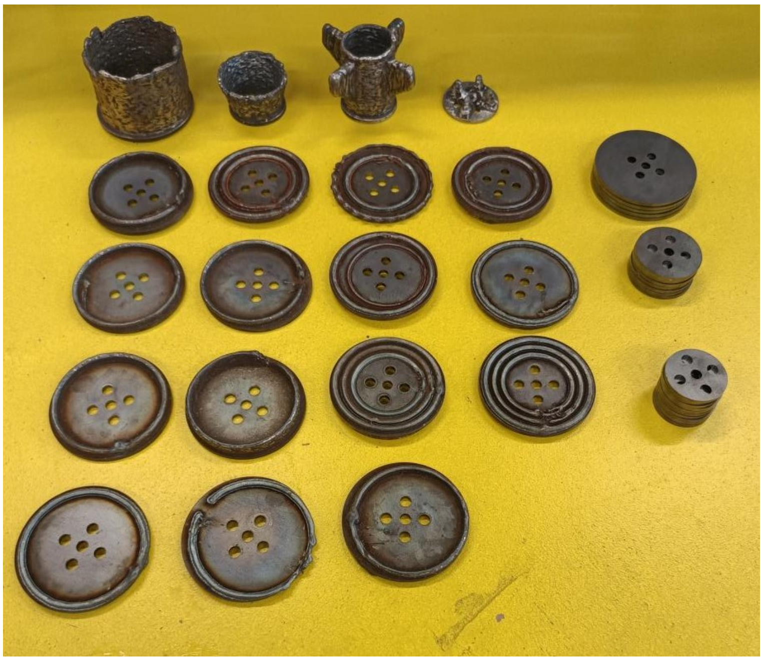


Fig. 28 – Every robotized print

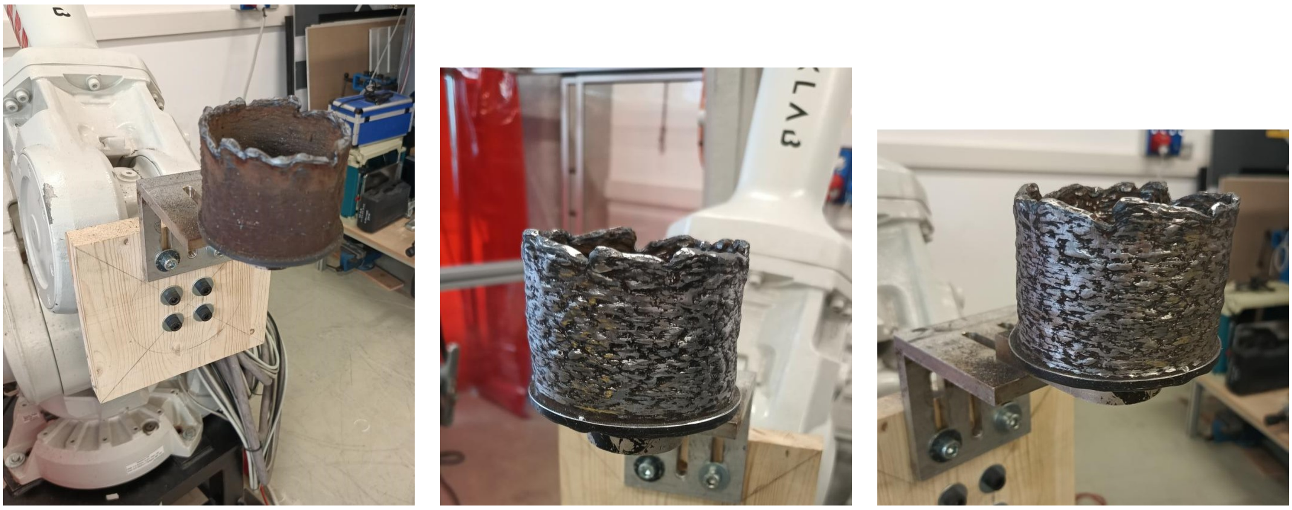


Fig. 29 – First printed piece

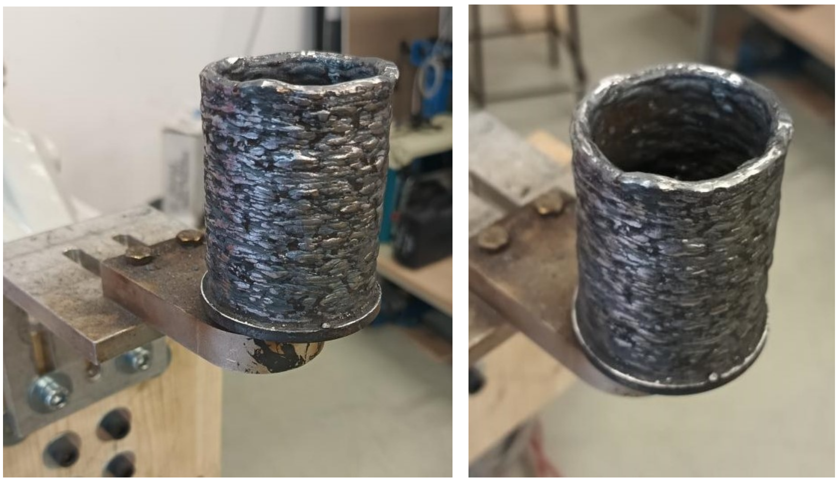


Fig. 30 – Second piece

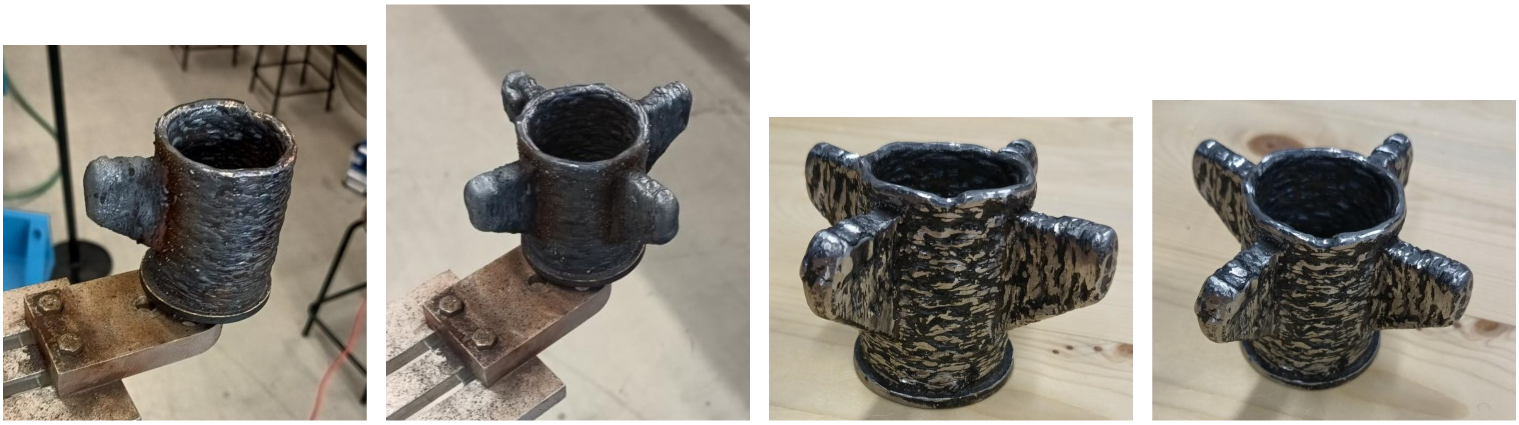


Fig. 31 – Modified second piece

7.7 GrassHopper robotics command plugins

7.7.1 Pick and Place

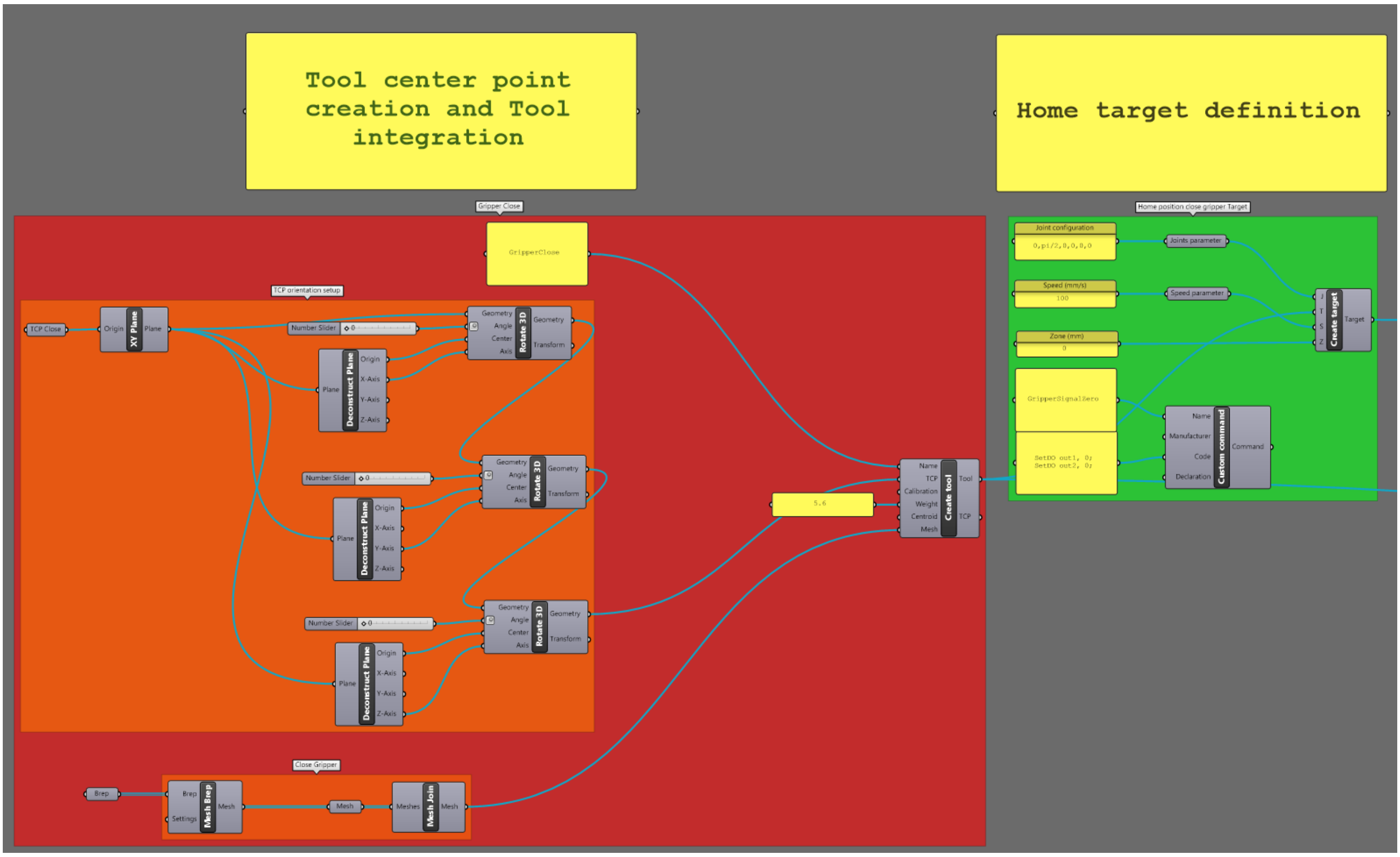


Fig. 32 – Robot plugin - GrassHopper - tool creation algorithm

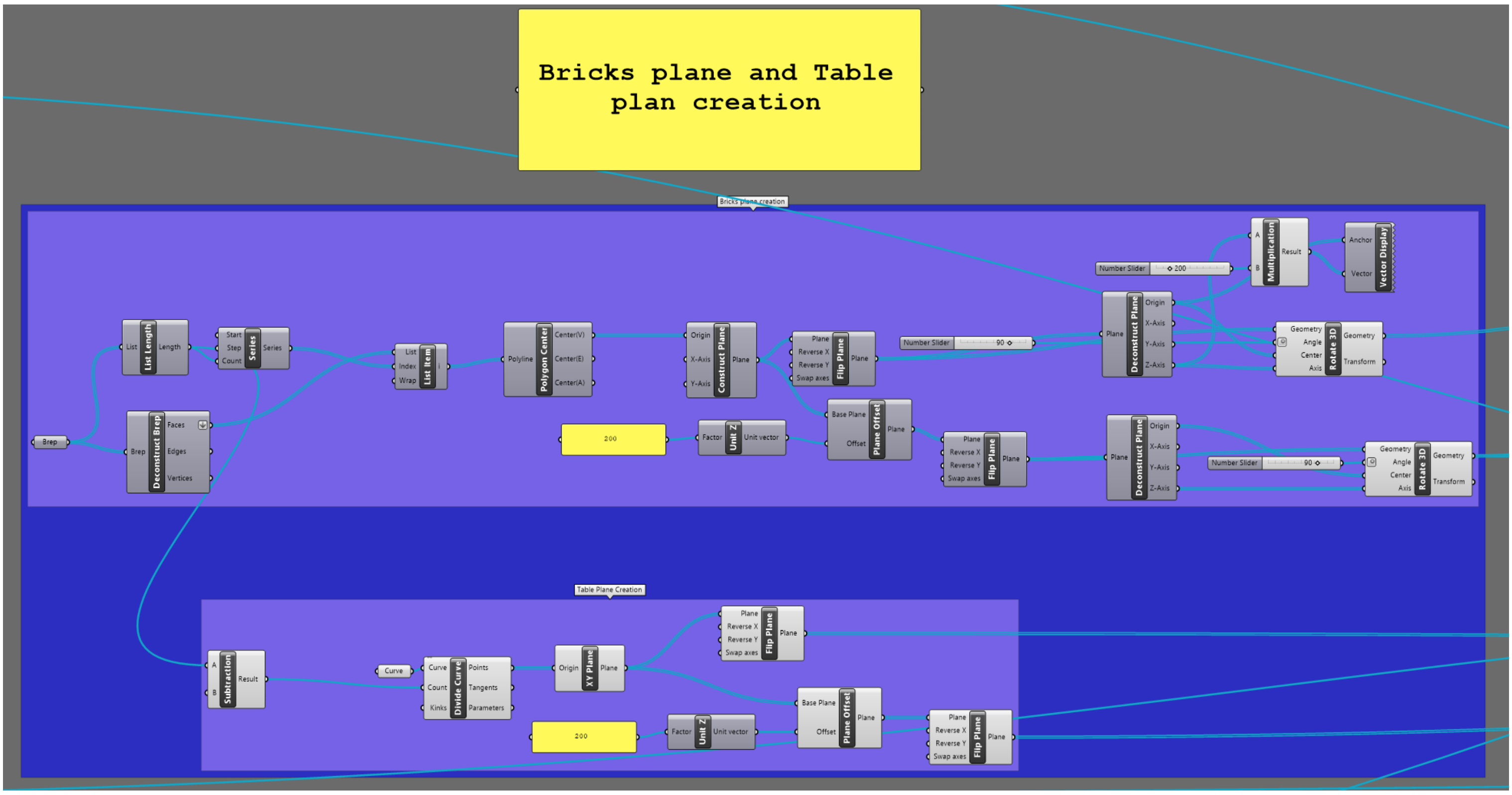


Fig. 33 – Robot plugin - GrassHopper - target frame creation algorithm

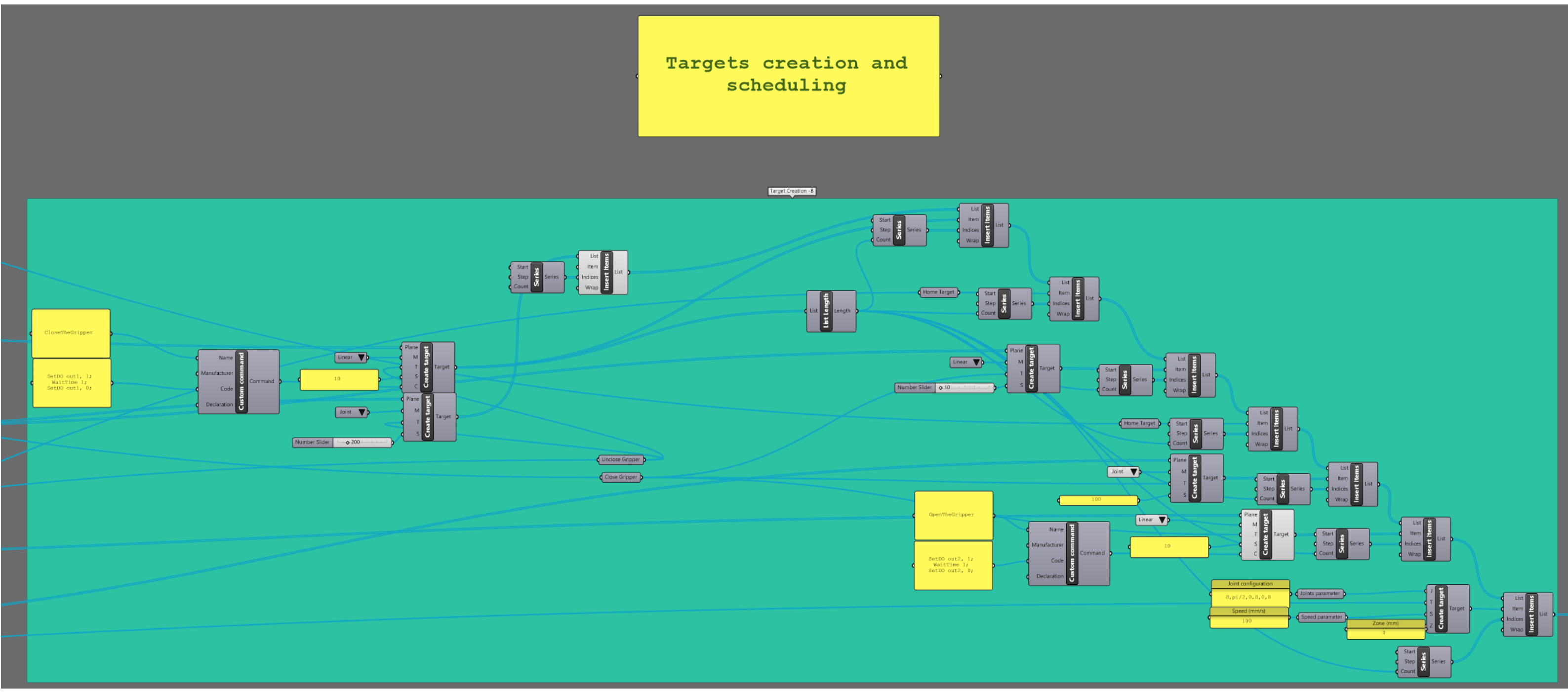
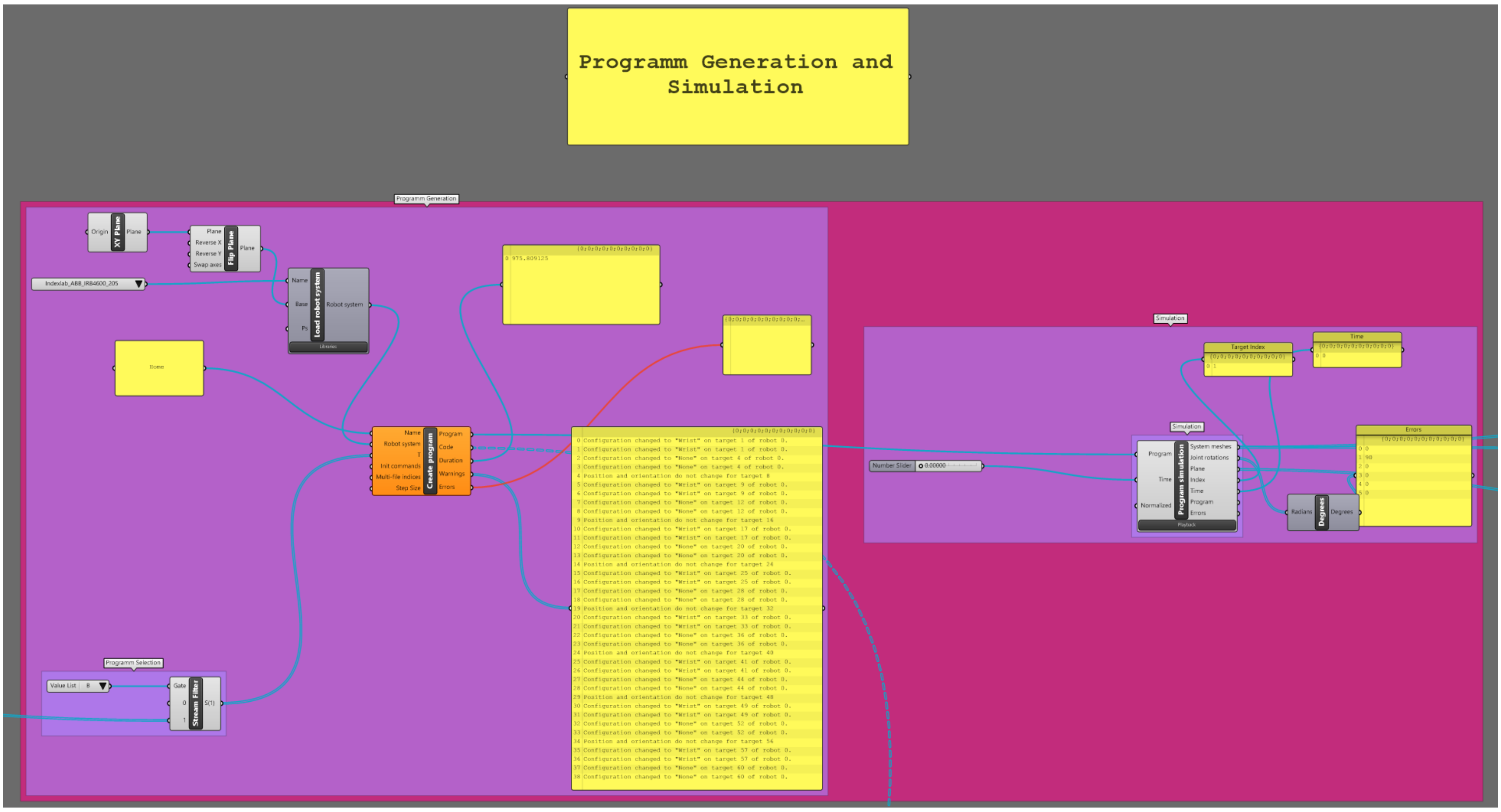


Fig. 34 – Robot plugin - GrassHopper - target creation and scheduling algorithm



```
{0;0;0;0;0;0;0;0;0;0;0}
 0 Configuration changed to "Wrist" on target 1 of robot 0.
 1 Configuration changed to "Wrist" on target 1 of robot 0.
 2 Configuration changed to "None" on target 4 of robot 0.
 3 Configuration changed to "None" on target 4 of robot 0.
 4 Position and orientation do not change for target 8
 5 Configuration changed to "Wrist" on target 9 of robot 0.
 6 Configuration changed to "Wrist" on target 9 of robot 0.
 7 Configuration changed to "None" on target 12 of robot 0.
 8 Configuration changed to "None" on target 12 of robot 0.
 9 Position and orientation do not change for target 16
10 Configuration changed to "Wrist" on target 17 of robot 0.
11 Configuration changed to "Wrist" on target 17 of robot 0.
12 Configuration changed to "None" on target 20 of robot 0.
13 Configuration changed to "None" on target 20 of robot 0.
14 Position and orientation do not change for target 24
15 Configuration changed to "Wrist" on target 25 of robot 0.
16 Configuration changed to "Wrist" on target 25 of robot 0.
17 Configuration changed to "None" on target 28 of robot 0.
18 Configuration changed to "None" on target 28 of robot 0.
19 Position and orientation do not change for target 32
20 Configuration changed to "Wrist" on target 33 of robot 0.
21 Configuration changed to "Wrist" on target 33 of robot 0.
22 Configuration changed to "None" on target 36 of robot 0.
23 Configuration changed to "None" on target 36 of robot 0.
24 Position and orientation do not change for target 40
25 Configuration changed to "Wrist" on target 41 of robot 0.
26 Configuration changed to "Wrist" on target 41 of robot 0.
27 Configuration changed to "None" on target 44 of robot 0.
28 Configuration changed to "None" on target 44 of robot 0.
29 Position and orientation do not change for target 48
30 Configuration changed to "Wrist" on target 49 of robot 0.
31 Configuration changed to "Wrist" on target 49 of robot 0.
32 Configuration changed to "None" on target 52 of robot 0.
33 Configuration changed to "None" on target 52 of robot 0.
34 Position and orientation do not change for target 56
35 Configuration changed to "Wrist" on target 57 of robot 0.
36 Configuration changed to "Wrist" on target 57 of robot 0.
37 Configuration changed to "None" on target 60 of robot 0.
38 Configuration changed to "None" on target 60 of robot 0.
```

Fig. 35 – Robot plugin - GrassHopper - program generatinon and simulation algorithm

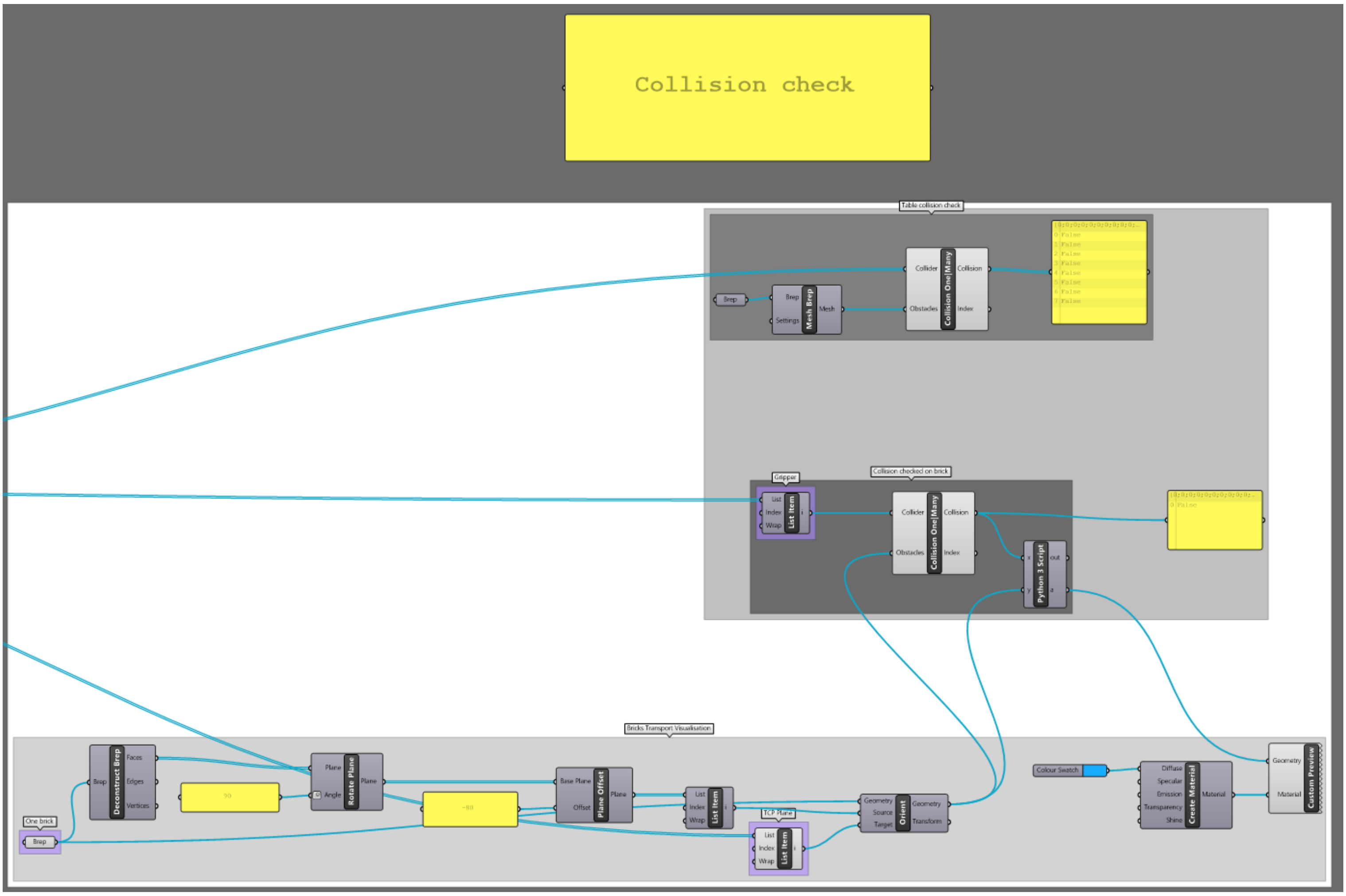


Fig. 36 – Robot plugin - GrassHopper - collision check algorithm

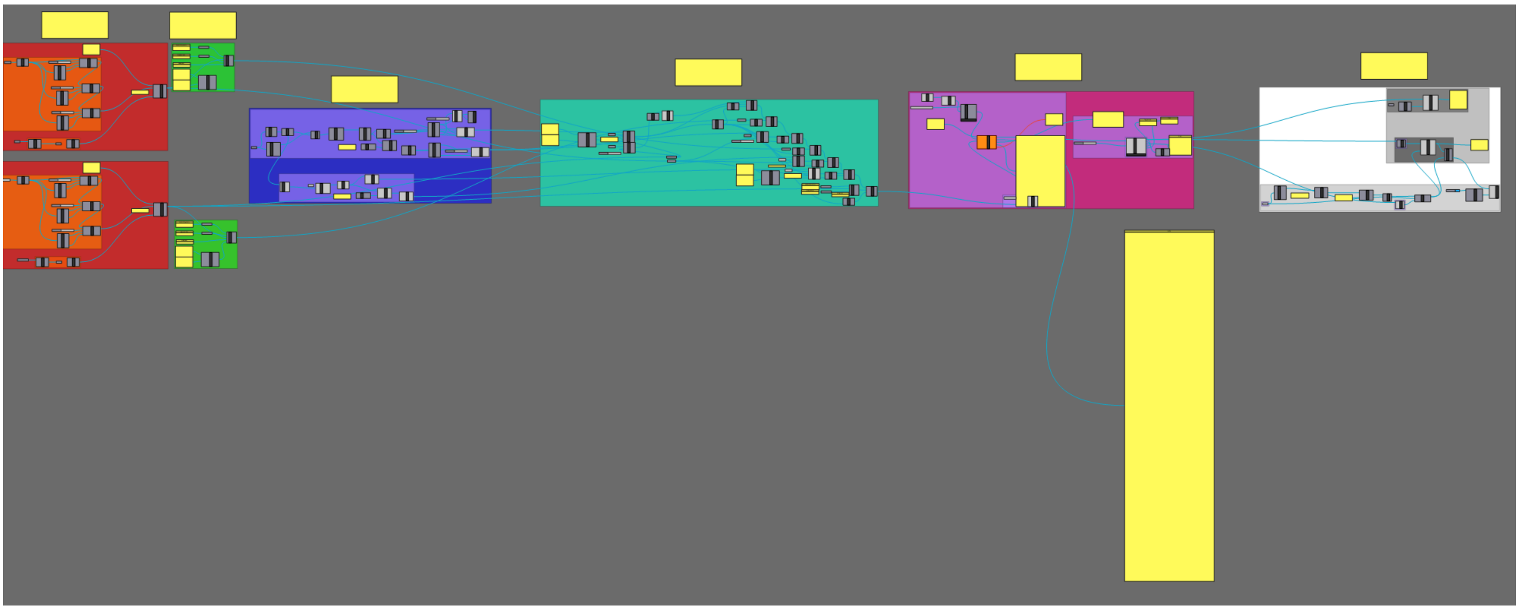


Fig. 37 – Robot plugin - GrassHopper - total algorithm

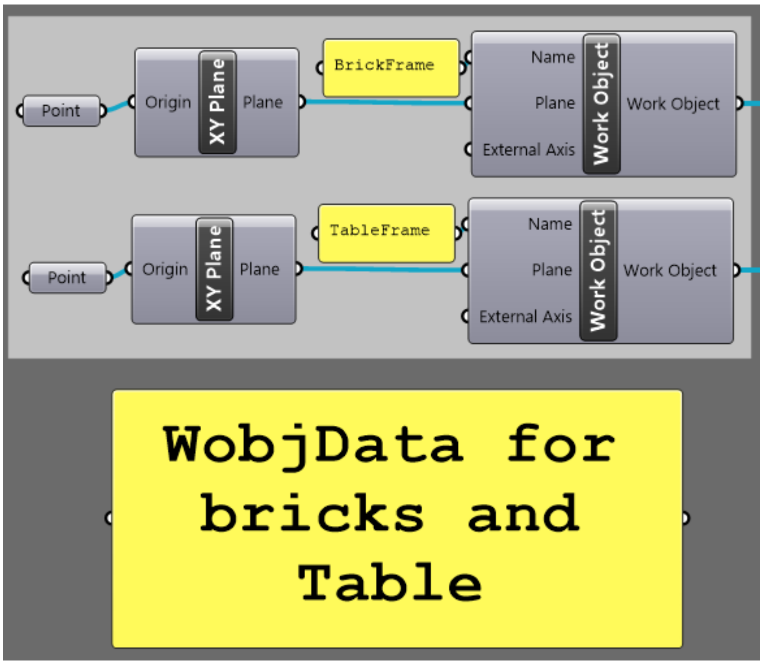


Fig. 38 – Robot component plugin - GrassHopper - main difference

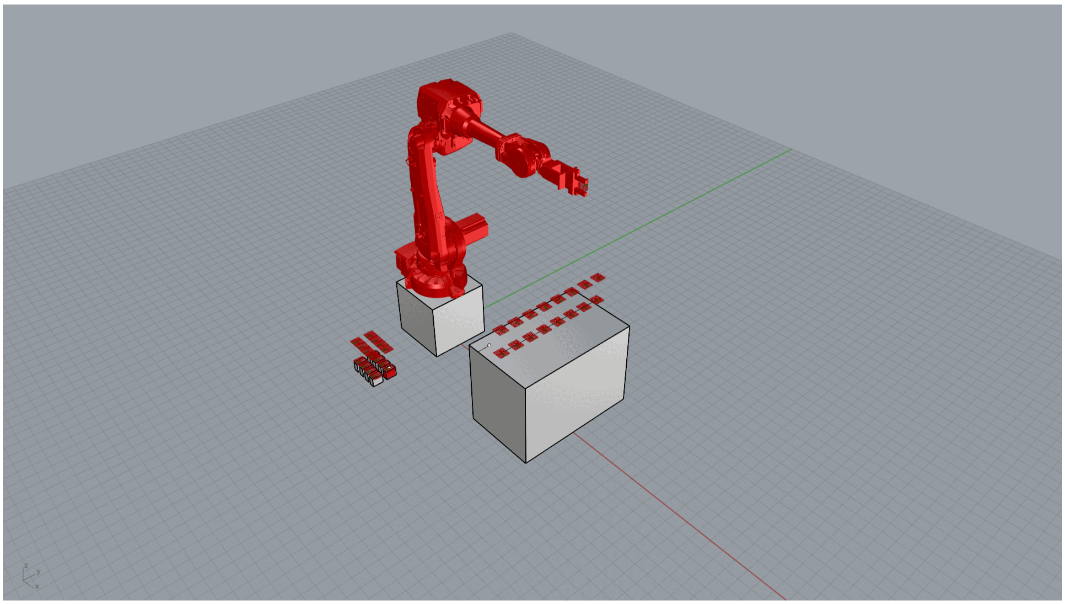


Fig. 39 – GrassHopper final Pick&Place simulation

7.7.2 Pick, Place and Weld

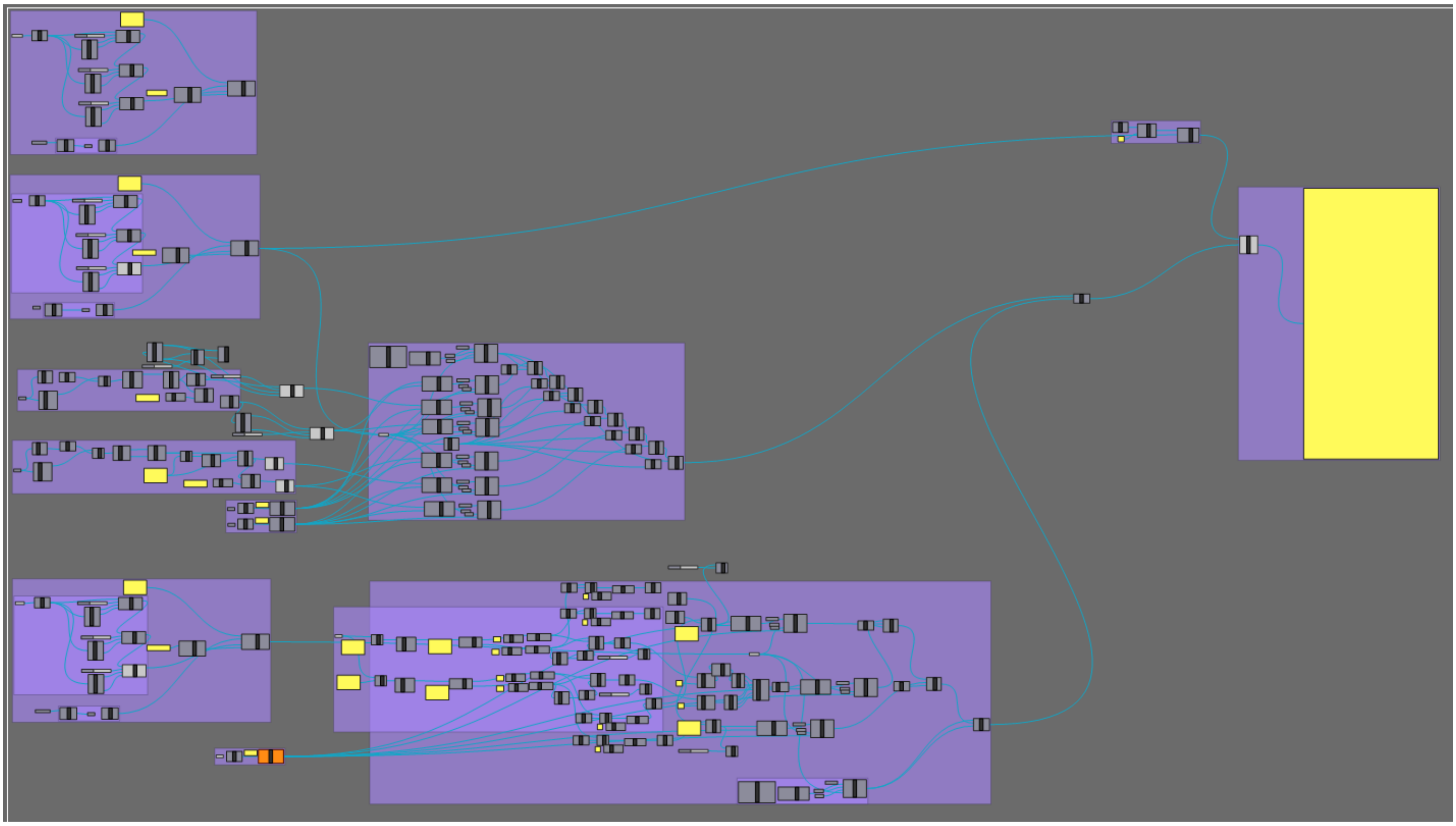
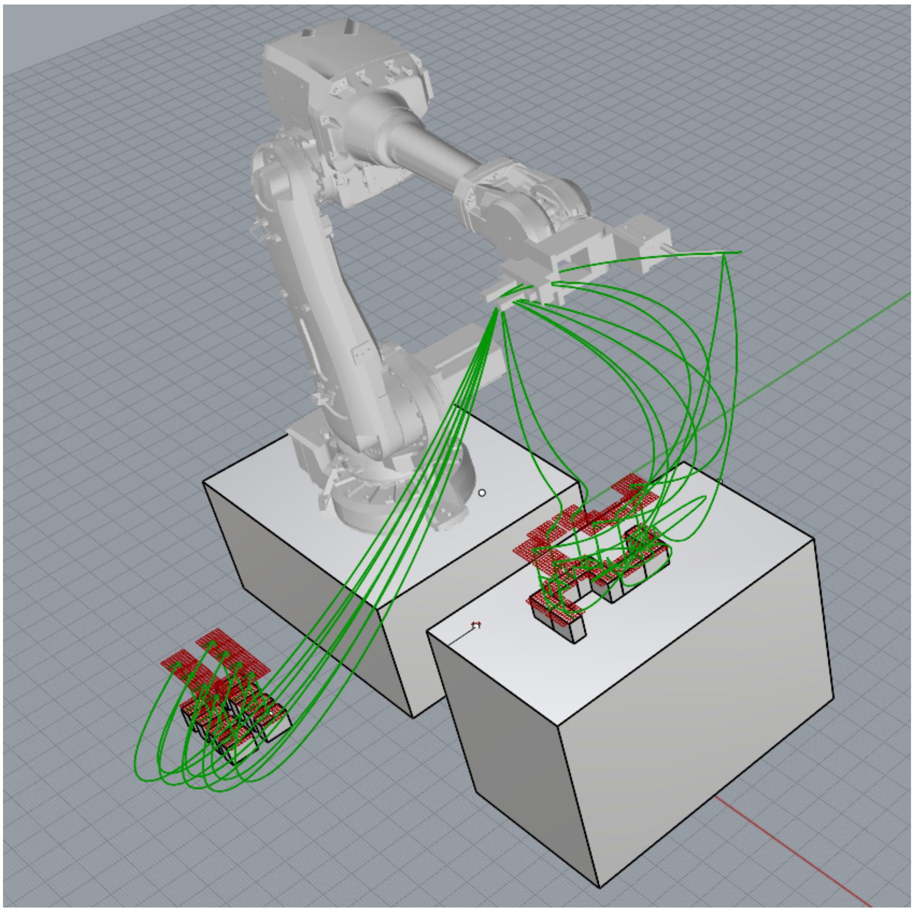


Fig. 40 – Robot component plugin - GrassHopper - Pick Place Weld total algorithm



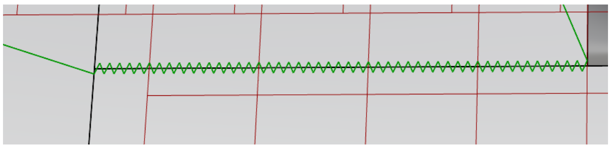


Fig. 42 – Robot component plugin - GrassHopper - Welding path close up view

Fig. 41 – Robot component plugin - GrassHopper - Simulation rendered with tool path trace

Raw output code of the Pick&Place GrassHopper simulation :

```
MODULE Home_T_ROB1
VAR extjoint extj := [9E9,9E9,9E9,9E9,9E9,9E9];
VAR confdata conf := [0,0,0,0];
PERS tooldata GripperUnclose := [TRUE,[[0.069,-0.022,335],[1,0,0,0]],[5.6,[0.069,-0.022,335],[1,0,0,0],0,0,0]];
PERS tooldata GripperClose := [TRUE,[[0.069,-0.022,335],[1,0,0,0]],[5.6,[0.069,-0.022,335],[1,0,0,0],0,0,0]];
TASK PERS wobjdata DefaultFrame := [FALSE,TRUE,"",[[0,0,0],[1,0,0,0]],[[0,0,0],[1,0,0,0]]];
TASK PERS speeddata Speed000 := [100,180,5000,1080];
TASK PERS speeddata Speed001 := [200,180,5000,1080];
TASK PERS speeddata Speed002 := [10,180,5000,1080];
PROC Main()
  ConfL \Off;
  MoveAbsJ [[0,-0,-0,0,-0,0],extj],Speed000,fine,GripperUnclose;
  SetDO out1, 0;
  SetDO out2, 0;

  MoveJ [[172.5,-655,262.5],[0,1,0,0],[-2,0,1,1],extj],Speed001,fine,GripperUnclose \WObj:=DefaultFrame;
  MoveL [[172.5,-655,62.5],[0,1,0,0],conf,extj],Speed002,fine,GripperUnclose \WObj:=DefaultFrame;
  SetDO out1, 1;
  WaitTime 1;
  SetDO out1, 0;
  MoveL [[172.5,-655,262.5],[0,1,0,0],conf,extj],Speed002,fine,GripperClose \WObj:=DefaultFrame;
  MoveAbsJ [[0,-0,-0,0,-0,0],extj],Speed000,fine,GripperClose;
  MoveJ [[1303,-415,1010],[0,0.70711,0.70711,0],[-2,0,0,0],extj],Speed000,fine,GripperClose \WObj:=DefaultFrame;
  MoveL [[1303,-415,810],[0,0.70711,0.70711,0],conf,extj],Speed002,fine,GripperClose \WObj:=DefaultFrame;
  SetDO out2, 1;
  WaitTime 1;
  SetDO out2, 0;
  MoveAbsJ [[0,-0,-0,0,-0,0],extj],Speed000,fine,GripperUnclose;
  MoveAbsJ [[0,-0,-0,0,-0,0],extj],Speed000,fine,GripperUnclose;
  SetDO out1, 0;
  SetDO out2, 0;

  MoveJ [[172.5,-780,262.5],[0,1,0,0],[-2,0,1,1],extj],Speed001,fine,GripperUnclose \WObj:=DefaultFrame;
  MoveL [[172.5,-780,62.5],[0,1,0,0],conf,extj],Speed002,fine,GripperUnclose \WObj:=DefaultFrame;
  SetDO out1, 1;
  WaitTime 1;
  SetDO out1, 0;
  MoveL [[172.5,-780,262.5],[0,1,0,0],conf,extj],Speed002,fine,GripperClose \WObj:=DefaultFrame;
  MoveAbsJ [[0,-0,-0,0,-0,0],extj],Speed000,fine,GripperClose;
  MoveJ [[1303,-285.571,1010],[0,0.70711,0.70711,0],[-2,0,0,0],extj],Speed000,fine,GripperClose \WObj:=DefaultFrame;
  MoveL [[1303,-285.571,810],[0,0.70711,0.70711,0],conf,extj],Speed002,fine,GripperClose \WObj:=DefaultFrame;
  SetDO out2, 1;
  WaitTime 1;
  SetDO out2, 0;
  MoveAbsJ [[0,-0,-0,0,-0,0],extj],Speed000,fine,GripperUnclose;
  MoveAbsJ [[0,-0,-0,0,-0,0],extj],Speed000,fine,GripperUnclose;
  SetDO out1, 0;
  SetDO out2, 0;
ENDPROC
ENDMODULE
```

7.8 RobotStudio simulations

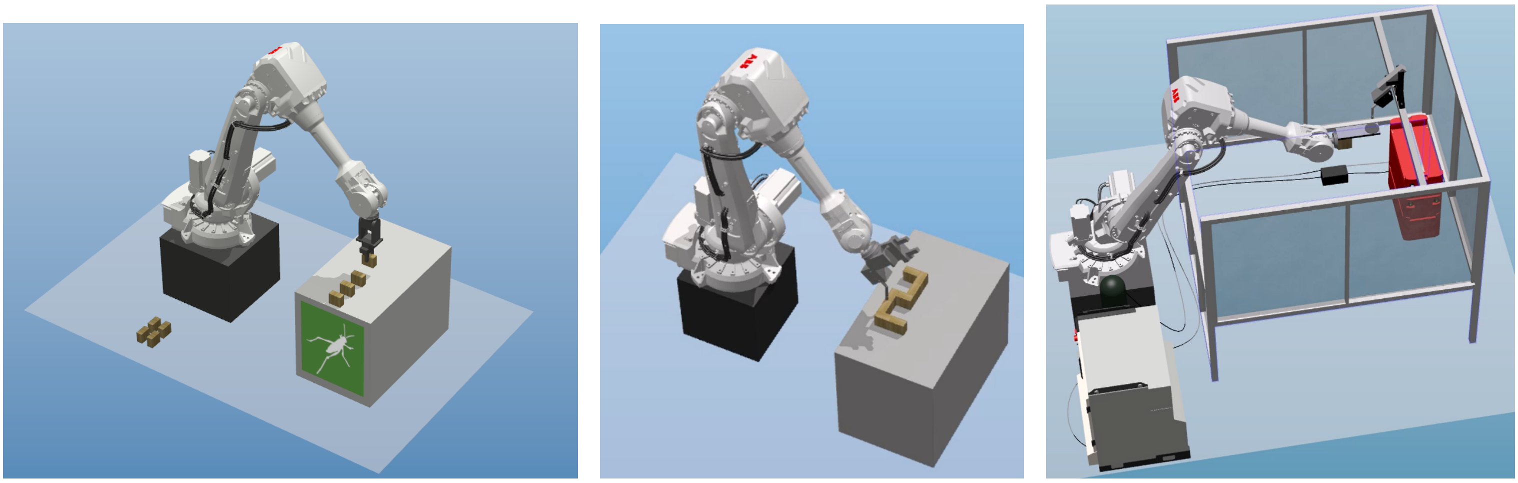


Fig. 43 – Three different RobotStudio simulations

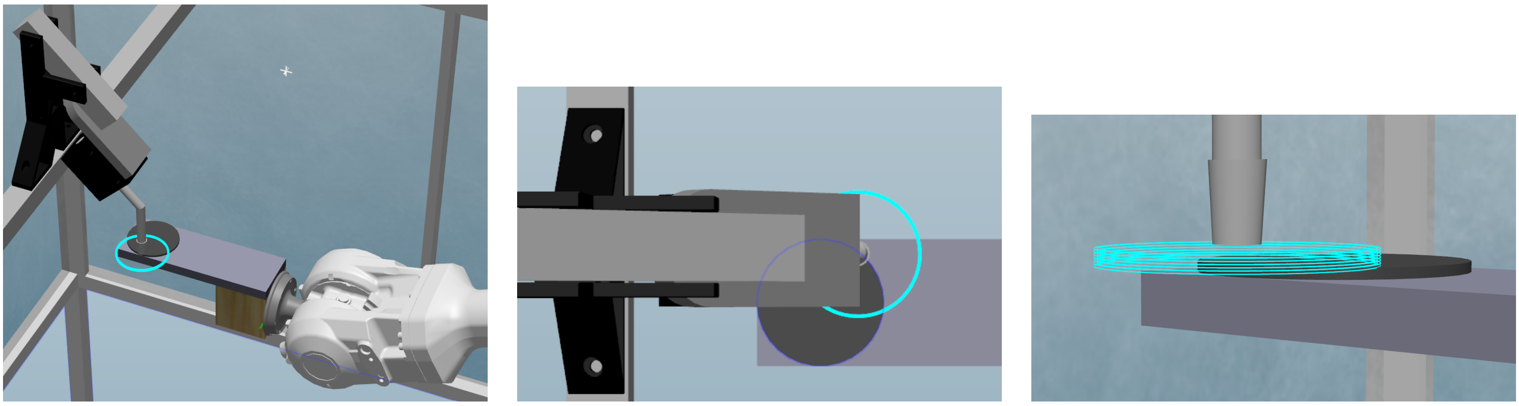


Fig. 44 – WAAM simulation close up view

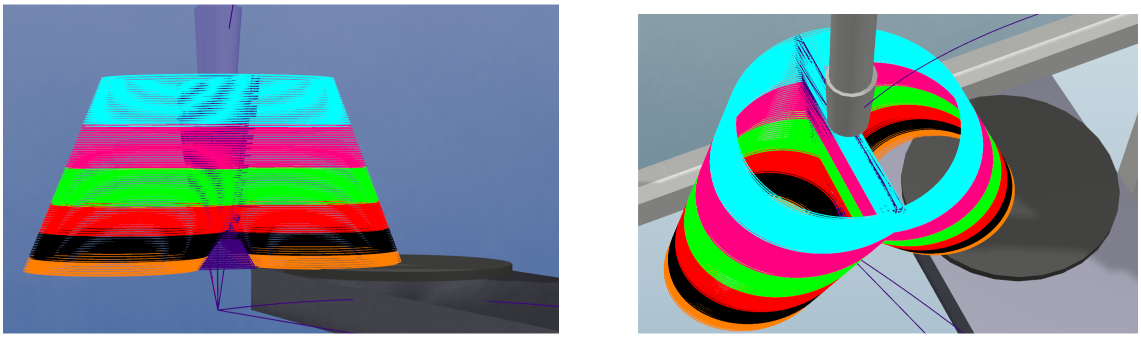


Fig. 45 – WAAM simulation tool path trace for a complex shape

7.9 RobotStudio simulation RAPID program examples

Main module of the Pick&Place RobotStudio simulation

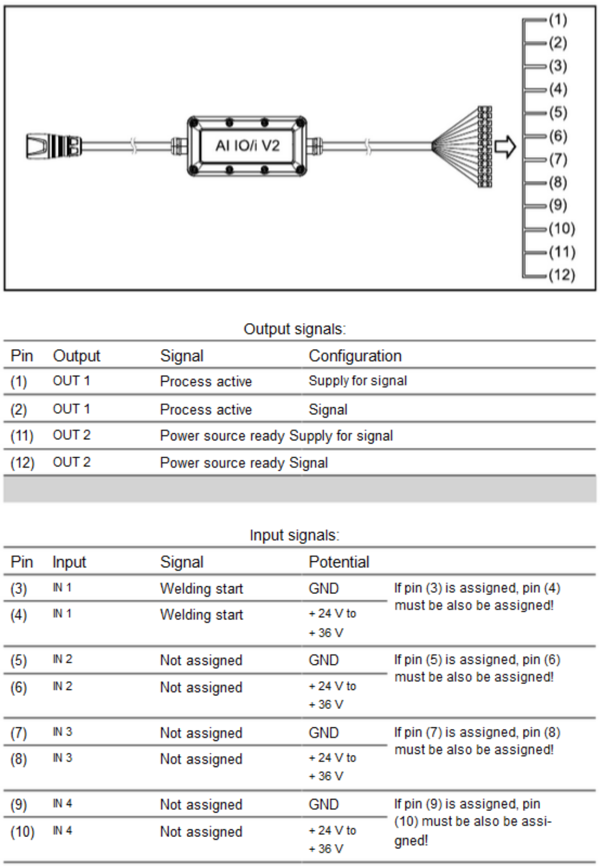
```
MODULE Pick_Place_Module

   CONST extjoint extj := [9E9,9E9,9E9,9E9,9E9,9E9];
   CONST confdata conf := [0,0,0,0];
   PERS wobjdata DefaultFrame := [FALSE, TRUE, "", [[0, 0, 0], [1, 0, 0, 0]], [[0, 0, 0], [1, 0, 0, 0]]];
   TASK PERS speeddata Fast_Speed := [2000,2000,2000,2000];
   TASK PERS speeddata Approach_Speed := [100,100,100,100];

   ! Targets declaration
   !Home
   CONST jointtarget Home := [[0,-0,-0,0,-0,0],extj];

   !Brick 1
   CONST num brickCounter1 := 1;
   CONST robtarget Brick1_Upper := [[0, 0, 200], [0, 1, 0, 0], [0, 0, 0, 0], extj];
   CONST robtarget Brick1_Lower := [[0, 0, 0], [0, 1, 0, 0], [0, 0, 0, 0], extj];
   CONST robtarget Table1_Upper := [[200, -51, 280], [0, 0.707107, 0.707107, 0], [0, 0, 0, 0], extj];
   CONST robtarget Table1_Lower := [[200, -51, 80], [0, 0.707107, 0.707107, 0], [0, 0, 0, 0], extj];

   !Brick 2
   CONST num brickCounter2 := 2;
   CONST robtarget Brick2_Upper := [[0 ,-125 ,200],[0,1,0,0],[-2,0,1,1],extj];
   CONST robtarget Brick2_Lower := [[0 ,-125 ,0],[0,1,0,0],[-2,0,1,1],extj];
   CONST robtarget Table2_Upper := [[200, 78.43, 280], [0, 0.707107, 0.707107, 0],[-2,0,0,0],extj];
   CONST robtarget Table2_Lower := [[200, 78.43, 80], [0, 0.707107, 0.707107, 0],[-2,0,0,0],extj];

   !Brick 3
   CONST num brickCounter3 := 3;
   CONST robtarget Brick3_Upper := [[-90 ,0 ,200],[0,1,0,0],[-2,0,1,1],extj];
   CONST robtarget Brick3_Lower := [[-90 ,0 ,0],[0,1,0,0],[-2,0,1,1],extj];
   CONST robtarget Table3_Upper := [[200, 207.86, 280], [0, 0.707107, 0.707107, 0] ,[-2,0,0,0],extj];
   CONST robtarget Table3_Lower := [[200, 207.86, 80], [0, 0.707107, 0.707107, 0] ,[-2,0,0,0],extj];

   !Brick 4
   CONST num brickCounter4 := 4;
   CONST robtarget Brick4_Upper := [[-90 ,-125 ,200],[0,1,0,0],[-2,0,1,1],extj];
   CONST robtarget Brick4_Lower := [[-90 ,-125 ,0],[0,1,0,0],[-2,0,1,1],extj];
   CONST robtarget Table4_Upper := [[200, 337.29, 280], [0, 0.707107, 0.707107, 0],[-2,0,0,0],extj];
   CONST robtarget Table4_Lower := [[200, 337.29, 80], [0, 0.707107, 0.707107, 0],[-2,0,0,0],extj];

   !Brick 5
   CONST num brickCounter5 := 5;
   CONST robtarget Brick5_Upper := [[-180 ,0 ,200],[0,1,0,0],[-2,0,1,1],extj];
   CONST robtarget Brick5_Lower := [[-180 ,0 ,0],[0,1,0,0],[-2,0,1,1],extj];
   CONST robtarget Table5_Upper := [[200, 466.71, 280], [0, 0.707107, 0.707107, 0],[0,0,1,0],extj];
   CONST robtarget Table5_Lower := [[200, 466.71, 80], [0, 0.707107, 0.707107, 0],[0,0,1,0],extj];

   !Brick 6
   CONST num brickCounter6 := 6;
   CONST robtarget Brick6_Upper := [[-180 ,-125 ,200],[0,1,0,0],[-2,0,1,1],extj];
   CONST robtarget Brick6_Lower := [[-180 ,-125 ,0],[0,1,0,0],[-2,0,1,1],extj];
   CONST robtarget Table6_Upper := [[200, 596.14, 280], [0, 0.707107, 0.707107, 0],[0,0,1,0],extj];
   CONST robtarget Table6_Lower := [[200, 596.14, 80], [0, 0.707107, 0.707107, 0],[0,0,1,0],extj];

   !Brick 7
```

```
CONST num brickCounter7 := 7;
CONST robtarget Brick7_Upper := [[-270 ,0 ,200],[0,1,0,0],[-3,0,0,1],extj];
CONST robtarget Brick7_Lower := [[-270,0 ,0],[0,1,0,0],[-3,0,0,1],extj];
CONST robtarget Table7_Upper := [[200, 725.57, 280], [0, 0.707107, 0.707107,
    ↪ 0],[0,0,1,0],extj];
CONST robtarget Table7_Lower := [[200, 725.57, 80], [0, 0.707107, 0.707107,
    ↪ 0],[0,0,1,0],extj];

!Brick 8
CONST num brickCounter8 := 8;
CONST robtarget Brick8_Upper := [[-270 ,-125 ,200],[0,1,0,0],[-3,0,0,1],extj];
CONST robtarget Brick8_Lower := [[-270 ,-125 ,0],[0,1,0,0],[-3,0,0,1],extj];
CONST robtarget Table8_Upper := [[200, 855, 280], [0, 0.707107, 0.707107,
    ↪ 0],[0,0,1,0],extj];
CONST robtarget Table8_Lower := [[200, 855, 80], [0, 0.707107, 0.707107,
    ↪ 0],[0,0,1,0],extj];

PROC Pick_Place(robtarget Brick_Upper, robtarget Brick_Lower, robtarget
    ↪ Table_Upper, robtarget Table_Lower, num brickCounter)
   SetDO out1, 0;
   SetDO out2, 0; ! signal initialisation
   MoveJ Brick_Upper, Fast_Speed, fine, GripperTool \WObj:= BrickFrame;
   MoveL Brick_Lower, Approach_Speed, fine, GripperTool \WObj:= BrickFrame;
   SetDO out1, 1; ! gripping the piece
   WaitTime 1; ! waiting for the mechanism to grip correctly
   ! gripped brick flag
   TEST brickCounter
      CASE 1:
         SetDO b1, 1;
      CASE 2:
         SetDO b2, 1;
      CASE 3:
         SetDO b3, 1;
      CASE 4:
         SetDO b4, 1;
      CASE 5:
         SetDO b5, 1;
      CASE 6:
         SetDO b6, 1;
      CASE 7:
         SetDO b7, 1;
      CASE 8:
         SetDO b8, 1;
   ENDTEST
   SetDO out1, 0;
   MoveL Brick_Upper, Approach_Speed, fine, GripperTool \WObj:= BrickFrame;
   MoveAbsJ Home, Fast_Speed, fine, GripperTool \WObj:= DefaultFrame;
   MoveJ Table_Upper, Fast_Speed, fine, GripperTool \WObj:= TableFrame;
   MoveL Table_Lower, Approach_Speed, fine, GripperTool \WObj:= TableFrame;
   SetDO out2, 1; ! releasing the piece
   WaitTime 1;
   ! gripped brick flag
   TEST brickCounter
      CASE 1:
         SetDO b1, 0;
      CASE 2:
         SetDO b2, 0;
      CASE 3:
         SetDO b3, 0;
      CASE 4:
         SetDO b4, 0;
      CASE 5:
         SetDO b5, 0;
      CASE 6:
         SetDO b6, 0;
      CASE 7:
         SetDO b7, 0;
      CASE 8:
         SetDO b8, 0;
   ENDTEST
   SetDO out2, 0; ! waiting the mechanism to release correctly
   MoveL Table_Upper, Approach_Speed, fine, GripperTool \WObj:= TableFrame;
   MoveAbsJ Home, Fast_Speed, fine, GripperTool \WObj:= DefaultFrame;
```

```
    ENDPROC

    PROC Main()
        ConfL \Off;
        MoveAbsJ Home, Fast_Speed, fine, GripperTool \WObj:= DefaultFrame;
        Pick_Place Brick1_Upper, Brick1_Lower, Table1_Upper, Table1_Lower,
            ↪ brickCounter1;
        Pick_Place Brick2_Upper, Brick2_Lower, Table2_Upper, Table2_Lower,
            ↪ brickCounter2;
        Pick_Place Brick3_Upper, Brick3_Lower, Table3_Upper, Table3_Lower,
            ↪ brickCounter3;
        Pick_Place Brick4_Upper, Brick4_Lower, Table4_Upper, Table4_Lower,
            ↪ brickCounter4;
        Pick_Place Brick5_Upper, Brick5_Lower, Table5_Upper, Table5_Lower,
            ↪ brickCounter5;
        Pick_Place Brick6_Upper, Brick6_Lower, Table6_Upper, Table6_Lower,
            ↪ brickCounter6;
        Pick_Place Brick7_Upper, Brick7_Lower, Table7_Upper, Table7_Lower,
            ↪ brickCounter7;
        Pick_Place Brick8_Upper, Brick8_Lower, Table8_Upper, Table8_Lower,
            ↪ brickCounter8;
    ENDPROC
ENDMODULE
```

7.10 Fronius TPS 320i

Fig. 46 – Fronius welding TPS 320 i



Output signals:

Pin	Output	Signal	Configuration
(1)	OUT 1	Process active	Supply for signal
(2)	OUT 1	Process active	Signal
(11)	OUT 2	Power source ready	Supply for signal
(12)	OUT 2	Power source ready	Signal

Input signals:

Pin	Input	Signal	Potential	
(3)	IN 1	Welding start	GND	If pin (3) is assigned, pin (4) must be also be assigned!
(4)	IN 1	Welding start	+ 24 V to + 36 V	
(5)	IN 2	Not assigned	GND	If pin (5) is assigned, pin (6) must be also be assigned!
(6)	IN 2	Not assigned	+ 24 V to + 36 V	
(7)	IN 3	Not assigned	GND	If pin (7) is assigned, pin (8) must be also be assigned!
(8)	IN 3	Not assigned	+ 24 V to + 36 V	
(9)	IN 4	Not assigned	GND	If pin (9) is assigned, pin (10) must be also be assigned!
(10)	IN 4	Not assigned	+ 24 V to + 36 V	

Fig. 47 – AI/IO v2 (signal scheduling unit)

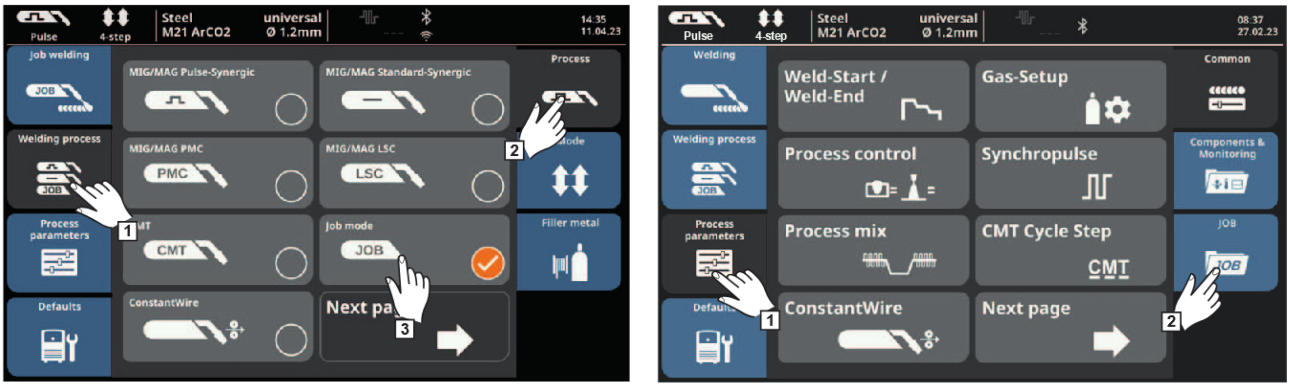


Fig. 48 – Fronius Job welding mode

7.11 Incremental sheet deformation

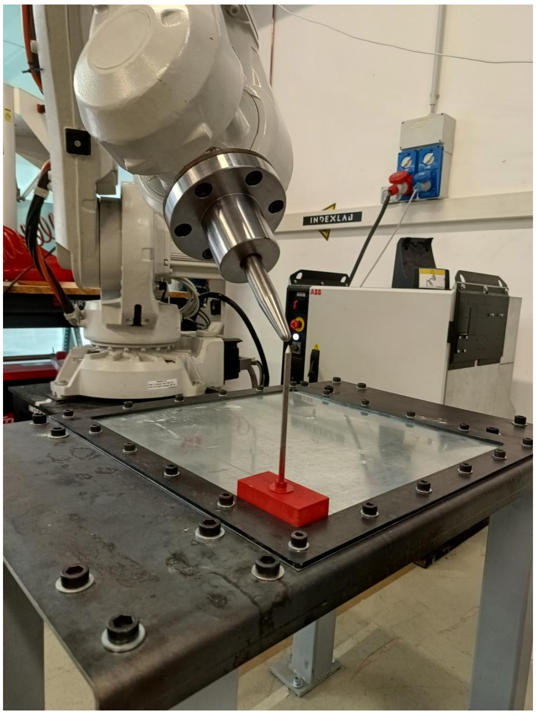


Fig. 49 – Calibration for the incremental sheet forming process

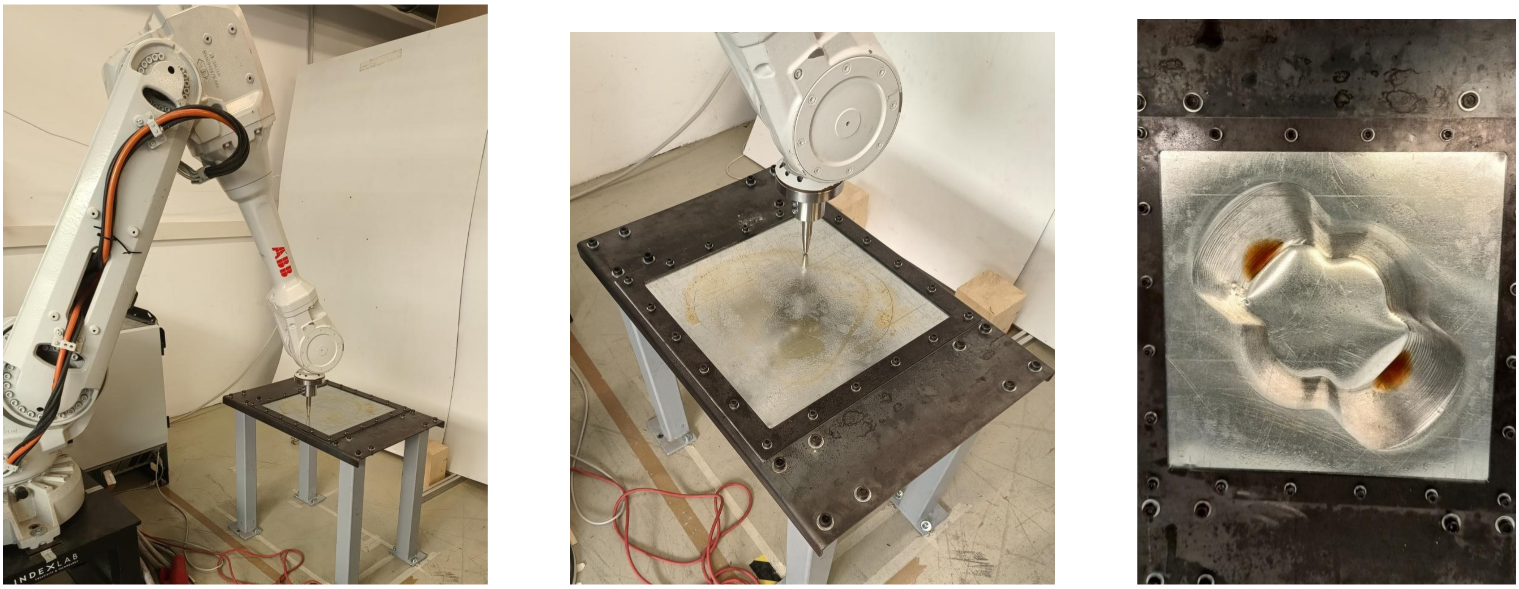


Fig. 50 – Incremental sheet deformation process

7.12 MyCobot280

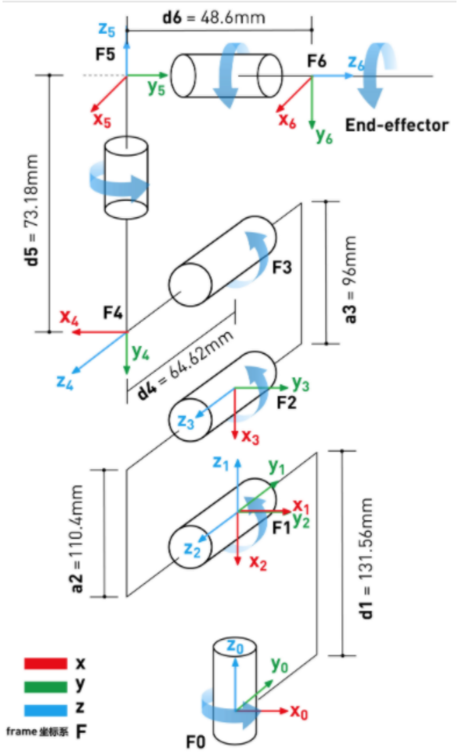
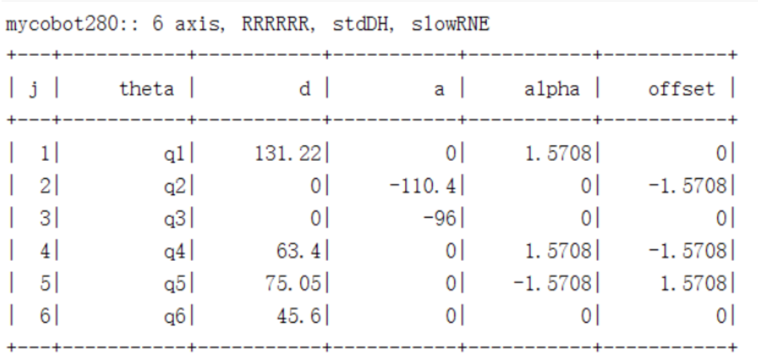


Fig. 51 – Mycobot 280 kinematic scheme



```
mycobot280:: 6 axis, RRRRRR, stdDH, slowRNE
+---+-----------+-----------+-----------+-----------+-----------+
| j |     theta |         d |         a |     alpha |    offset |
+---+-----------+-----------+-----------+-----------+-----------+
|  1|         q1|    131. 22|          0|    1. 5708|          0|
|  2|         q2|          0|    -110. 4|          0|   -1. 5708|
|  3|         q3|          0|        -96|          0|          0|
|  4|         q4|      63. 4|          0|    1. 5708|   -1. 5708|
|  5|         q5|     75. 05|          0|   -1. 5708|    1. 5708|
|  6|         q6|      45. 6|          0|          0|          0|
+---+-----------+-----------+-----------+-----------+-----------+
```

Fig. 52 – Mycobot 280 Denavit-Hartenberg kinematic formalism

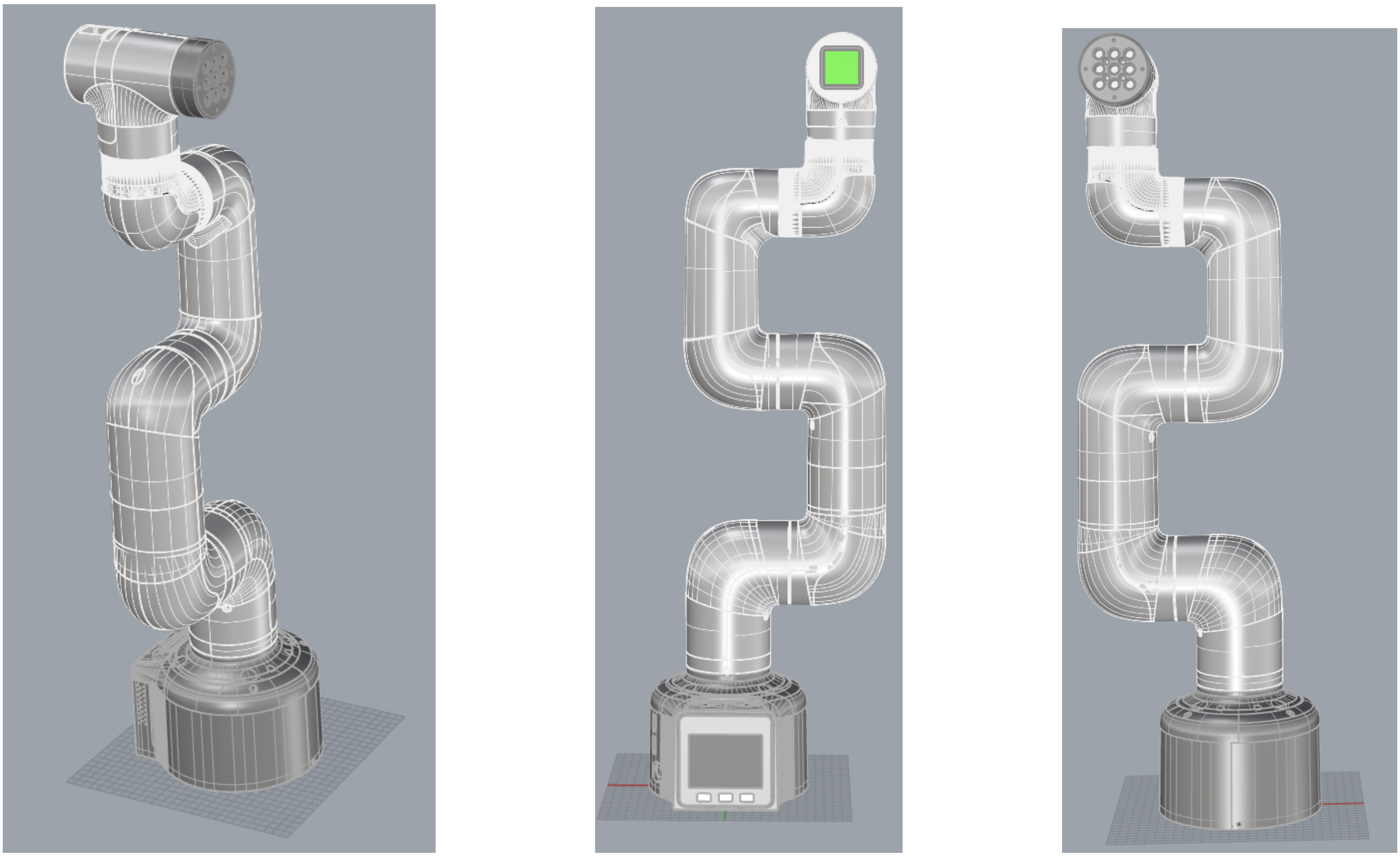


Fig. 53 – MyCobot 280 3D model

7.13 GitHub link

Here is the link for my GitHub internship page, where you can find the GrassHopper to RobotStudio guide I have written during my internship : GitHub link

7.14 Student and company evaluation documents

(Documents begin next page)

This form is to be completed by the company tutor and returned by August 19 to sre@seatech.fr.

Host organization: IndexLab, Politecnico di Milano
Student: Thibault Courtois
Internship subject: Robotic Creation

Assessment: skills development

The assessment grid is consistent with the training's skills framework. The 6 skills are developed over the three years of training. Please evaluate the trainee according to the different criteria.

References for quotation

(A) Excellent	Beyond expectations (beyond students of the same level)
(B) Suitable	Meets expectations (within the norm for students of the same level)
(C) Insufficient	Does not meet expectations (below students of same level)
(N) Not applicable	Criterion not assessable in the context of the course

Competence	Criteria	Eval	Comment
Design engineering solutions	Analyze your needs	A	
	Meeting the need	A	
	Use the appropriate tools	A	
	Document your choices and sources	A	
Implement ing solutions	Analyze and improve an existing solution	A	
	Propose a new solution	A	
	Use the principles of improvement continues	A	
	Writing a document scientific and technical support	A	
Developing an R&D approach	Carrying out a technology watch state of the art	A	
	Formulating assumptions	A	
	Propose an experimental approach, a protocol or an model	A	

NB: For ease of reading, the masculine gender is used without discrimination between male and female.

	Adopting an innovative approach	A	

NB: For ease of reading, the masculine gender is used without discrimination between male and female.

SEATECH TOULON

Company tutor evaluation sheet 2023-24

Managing engineering projects	Take part in or lead one or more stages of a project project	A	
	Take into account the overall management of organizations or the rules of operation, economic or social. legal	N	
	Use the tools of project management and collaborative tools	N	
	Identify or mobilize resources appropriate	A	
Managing a team	Integrating and collaborating	A	
	Working in a team multidisciplinary and/or international	A	
	Ensuring leadership responsibility	A	
	Communicate (orally and in writing) adapted	A	
Acting as a responsible professional	Take into account ethical and societal issues (CSR, SD...), RGPD, ...)	N	
	Take responsibility for your acts and decisions	A	
	Taking a look critical about the meaning of the activity	A	
	Being in a dynamics learning	A	

Overall assessment of the trainee and recommendations for an engineering position ***(it is advisable to discuss this assessment with the trainee)***

If you have detected any personal qualities in the trainee, such as the ability to overcome difficulties, to accept criticism, to express points of view in a well-argued way, or any other aspect, you can note it here for his/her benefit.

Thibault Courtois is someone in whom you can place your trust. He has a strong sense of responsibility and works with a focused approach. He is autonomous and knows how to mange his energy to deliver great results.

Date, company stamp, name and signature of host organization tutor

NB: For ease of reading, the masculine gender is used without discrimination between male and female.

Politecnico di Milano
Polo territoriale di Lecco
09 AGO. 2024

SEATECH TOULON

Fiche d’évaluation étudiant

Ce document est à insérer dans le rapport. Le rapport est à fournir pour le 19 aout dernier délai.

Organisme d’accueil :
Sujet du stage :

Evaluation qualitative du stage	I	S	B	E	N	Commentaires
Qualité de l’entreprise						
L’entreprise offre-t-elle un contexte propice à une carrière d’ingénieur ?			X			
L’entreprise connaissait elle l’école (accueil de stagiaire, embauche, relations autres…) ?	X					
L’entreprise a-t-elle mis à votre dispositions les moyens nécessaires pour réaliser votre mission (documents, éléments d’information, matériels) ?				X		
Qualité de la missions						
Vos missions étaient-elles en rapport avec votre formation ?				X		
Les missions effectuées étaient-elles bien celles définies au départ ?				X		
Qualité de l’encadrement						
Votre tuteur organisme d’accueil a-t-il pris le temps de vous présenter le fonctionnement de la structure et l’équipe ?				X		
Votre tuteur organisme d’accueil vous a-t-il aidé et conseillé quand cela était nécessaire ?				X		
Votre enseignant référent vous a-t-il aidé et conseillé lorsque cela était nécessaire ?				X		

Explication des cotations	
I	(I) Insuffisant
S	(S) Suffisant
B	(B) Bien
E	(E) Excellent
N	(N) Non applicable

NB : Dans le but d’alléger la lecture du document, le genre masculin est utilisé sans discrimination pour le genre masculin et féminin.

SEATECH TOULON

Fiche d'évaluation étudiant

Autoévaluation : développement des compétences et trajectoire professionnelle

En prenant un peu de recul sur votre activité durant le stage pensez-vous avoir travaillé / développé certaines des compétences du référentiel de la formation Seatech ? Lesquelles ? Pourquoi et comment ? D'autres compétences ?

Compétence	Critère	Commentaire
Concevoir des solutions d'ingénierie	Analyser le besoin	Qualité développée en particicipant à différents projets de création robotisés/automatisés
	Répondre au besoin	Idem, les projets analysés et les solutions trouvées ont par la suite été mises en place
	Utiliser les outils appropriés	J'ai appris à utiliser de nouveaux outils comme Rhino3D, GrassHopper et RobotStudio pour réaliser les missions demandées
	Documenter ses choix et ses sources	J'ai rédigé un guide d'utilisation de GrassHopper et RobotStudio pour la commande de robot 6 axes dans lequel j'ai documenté mes sources
Mettre en œuvre des solutions	Analyser et améliorer une solution existante	J'ai analysé la solution de WAAM avec une imprimante 3D WASP pour l'améliorer en une solution de WAAM avec un robot 6 axes.
	Proposer une solution nouvelle	Idem, développement d'un processus WAAM avec un robot ABB 6 axes. Post processing d'un logiciel de slicing d'imprimante 3D plastique pour faire de l'impression métal
	Utiliser les principes de l'amélioration continue	J'ai essayé dans la limite du possible d'apporter le plus d'optimisation à mon travail pour maximiser mes résultats
	Rédiger un document scientifique et technique d'appui	J'ai rédigé un guide d'utilisation de GrassHopper et RobotStudio pour la commande de robot 6 axes dans lequel j'ai documenté mes sources
Développer une démarche R&D	Réaliser une veille technologique / un état de l'art	Non appliqué je me suis cependant mis à jour sur les méthodes de WAAM et les technologies de soudure mais pas pour mon entreprise
	Formuler des hypothèses	Lors des batteries de tests que j'ai effectuées pour optimiser l'impression de métal, j'ai formulé des hypothèses sur les résultats pour apporter des solutions et des optimisations
	Proposer une démarche expérimentale, un protocole ou un modèle	Idem, démarche expérimentale pour la WAAM (wire arc additiv manufacturing)
	Adopter une démarche d'innovation	Le procédé de WAAM a été réalisé avec une configuration inversée : la torche à souder est statique et le robot tient la pièce pendant l'impression

Fiche d'évaluation étudiant

Piloter des projets d'ingénierie	S'insérer dans ou conduire une ou plusieurs étapes d'un projet	J'ai conduit le projet de WAAM avec le bras ABB robotisé de A à Z en autonomie
	Prendre en compte la gestion globale des organisations ou les règles de fonctionnement, économiques ou juridiques	Non appliqué
	Utiliser les outils de gestion de projet et outils collaboratifs	Non appliqué, j'ai travaillé au maximum quitte à dépasser mes horaires de travail. Mais je ne l'ai pas structuré avec un logiciel par exemple.
	Identifier ou mobiliser les ressources appropriées	J'ai essayé au maximum d'identifier les contraintes matérielles pour optimiser mon emploi du temps de travail
Encadrer une équipe	S'insérer et collaborer	J'ai aidé les étudiants en thèse master dans leur travail, j'ai travaillé avec l'équipe internationale d'IndexLab
	Assurer une responsabilité d'animation	J'ai participé à un projet pour faire découvrir la robotique industrielle à des lycéens sur 3 séances de 4 heures en début de stage
	Travailler en équipe pluridisciplinaire et/ou internationale	J'ai aidé des étudiants en thèse master, j'ai travaillé avec l'équipe internationale d'IndexLab
	Communiquer (écrit et oral) de manière adaptée	J'ai essayé d'être le plus clair possible dans mon expression orale en anglais, j'ai aussi rédigé mon guide en anglais
Agir en professionnel responsable	Prendre en compte les enjeux éthiques et sociétaux (RSE, DD, RGPD, …)	Non appliqué
	Assumer la responsabilité de ses actes et décision	Toujours
	Porter un regard critique sur le sens de l'activité conduite	Mon premier banc d'essai WAAM était peu sécurisé par rapport aux dangers (arc électrique, rayonnement lumineux, hautes températures). J'ai fait un pas en arrière et demandé de l'aide à mes collègues pour créer un autre dispositif.
	Être dans une dynamique d'apprentissage	Quand il y avait trop de contraintes matérielles pour que je puisse avancer dans mes missions, j'ai décidé d'apprende l'algèbre des quaternions

Trajectoire professionnelle

A la suite de votre stage, avez-vous confirmé ou affiné votre projet professionnel d'être ingénieur (métier plus précis, secteur, contexte ou type d'entreprise, …) ? Si oui, quelles actions pensez-vous devoir entreprendre pour y arriver (renforcer certaines connaissances, développer certaines compétences, lesquelles)?

J'ai adoré travailler pour IndexLab sur des applications de robotique industrielle et de R&D.
Je chercherai peut être un premier poste dans le secteur industriel pour contrinuer à travailler dans ce secteur.
Il faut que je renforce mes compérences manuelles (outillage éléctroportatif)

Date et signature de l'étudiant :

18/08/2024 Thibault Courtois

Bibliography

- **Wikipedia**. Denavit - Hartenberg parameters.
  Available on : https://en.wikipedia.org/wiki/Denavit%E2%80%93Hartenberg_parameters

- **Mecademic**. How to use quaternions in industrial robotic.
  Available on : https://www.mecademic.com/fr/academic_articles/comment-utiliser-les-quaternions-en-robotique-industrielle/

- **ABB Operating manual**. IRC5 with FlexPendant.
  Available on : https://us.v-cdn.net/5020483/uploads/editor/x7/0kqcav7a5yfx.pdf

- **ABB Product manual**. IRB 4600.
  Available on : https://search.abb.com/library/Download.aspx?DocumentID=3HAC033453-001&LanguageCode=en

- **Fronius operating instructions manual**. TPS 320i C.
  Available on : https://www.rapidwelding.com/files/FRONIUS-TPS-320I-C-PULSE-Operating-Instructions.pdf

- **Andrew J. Hanson**. Visualizing Quaternions. Elsevier, 2006.