Grant agreement nº: 101006798

Call identifier: MG-3-7-2020

Deliverable D2.3
Robot portfolio for collaborative operation in shipbuilding operations

Work Package 2
SEAMLESS AND INTUITIVE HUMAN-ROBOT COLLABORATIVE SOLUTIONS

Document type : Demonstrator Deliverable
Version : 1.0
Date of issue : 12/06/2023
Dissemination level : PUBLIC
Lead beneficiary : GIZELIS

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement nº 101006798.

DOCUMENT HISTORY

Version	Date	Changes	Stage	Distribution
0.1	21/02/2023	First Draft	Draft	Internal
0.2	27/02/2023	Partners contributions added	Draft	AIMEN
0.3	14/03/2023	Missing contributions added	Draft	WP2 partners
0.4	15/04/2023	Document updated with partners changes	Draft	WP2 partners
0.5	23/05/2023	Minor format/content changes and inclusion of links for the demo videos.	Final	All
1.0	12/06/2023	Minor changes	Final	All

EXECUTIVE SUMMARY

The objective of this deliverable is to showcase the robot portfolio designed for collaborative operations in shipbuilding within the scope of WP2. It begins with a comprehensive overview of each tool included in the portfolio. Following that, the lab-scale demonstrators are presented, highlighting the integration of various technologies and tools. Each section dedicated to a demonstrator is complemented by a video showcasing its capabilities. This deliverable provides a detailed exploration of the developed robot portfolio, offering valuable insights into its functionality and practical applications in shipbuilding operations.

1 INTRODUCTION

Human-robot collaboration is a rapidly growing field that has the potential to revolutionize the shipyard manufacturing sector. The advancements in robotics technology have made it possible to integrate robots into human environments, leading to increased efficiency and productivity. D2.3 focuses on reporting the developments implemented in the context of WP2, as in total they consist of a robot portfolio for collaborative operations in the shipbuilding sector. The innovations are collected and dependent on different scenarios derived from the shipyard manufacturing sectors, a combination of technologies is integrated. The innovation to be discussed are the following:

- Provision of collaborative robot-based solutions that allow the improvement of production performance and precision, preserving industry-specific knowledge and worker skills. The deployment of the collaborative robotic solutions will be compliant with the ISO directives, for human safety and ergonomics.
- Development of a unified and adaptable robot perception system for advanced collaboration of mobile and articulated robots. The system comprises a modular 2D/3D perception backbone and machine vision algorithms tailored to meet functional specifications and use case requirements. Interfaces for real-time configuring of the machine vision pipeline will also be created to serve shipyard operators and integrators.
- Development of control modules for monitoring and controlling flexible robotic systems, with a focus on human-centered interfaces. Two levels of orchestration and planning: local orchestration by the Task Manager (ROS-based, translates production tasks to executable structures) and coordination by the Production Manager (web-based, communicates with IoT platforms and production equipment, offers data visualization and supervision).
- Development of intuitive human-machine interfaces for robot programming and control in shipbuilding. Approaches include AR assisted robot path teaching using an AR pen, hand-guiding, and skill-based operation for easy robot control and configuration.

2 HIGH LEVEL MODULES DESCRIPTION

2.1. Process Perception Module (PPM)

The PPM module consists of a set of modular and reconfigurable perception and localization systems, or sub-modules, that can be easily deployed and used in conjunction with robotic systems to increase their flexibility and adaptability. Their design was made in accordance with the shipyard's use case scenarios.

In Table 1 are summarized the sub-modules that were developed. Their detailed description can be found in D2.2.

Table 1. PPM sub-modules developed under WP2.

PPM sub-modules	Brief Description
Object Perception (3OPS)	Provides a reconfigurable pipeline to dynamically construct a 3D-based perception flow. The main algorithms and their inclusion within the perception system's main

	stages, which include down sampling, noise removal, segmentation, clustering, surface normal estimation, key point estimation and description, initial alignment point cloud registration, object symmetry postprocessing and point cloud registration analysis. The perception pipeline can be used with an external object segmentation module or be configured to use the clustering algorithm.
Object Grasping Planner (OGPS)	Provides different tools to generate a set of feasible grasping positions candidates considering a specific geometry. The grasping candidates are generated off-line based on two main tool classes: the GraspIt! and the Mimic Grasping interface. At run-time the best grasp is selected over a set of previously taught grasp poses of an object. The best grasping candidate is chosen according to a cascade of heuristics defined by the user in a YAML file for each object detected. Therefore, the object needs to be identified and localized before the grasp selection process.
Object Segmentation based on DL (OSDL)	Provides an object detection pipeline, design for the detection of objects in cluttered environments for pick and place operation. The pipeline is mainly divided into two phases, 2D detection, and 3D point cloud segmentation. The first step, which receives as input a 2D image of the scene, identifies individual objects in the cluttered and proceeds to segment each one. This step was developed based on AI tools, that require high amount of date to train the specific models. Therefore, a tool based on blender was developed to allow the generation of data through simulation to expedite the training of the AI model. The second step is an algorithm that receives all the 2D segmentation and processes the 3D point cloud according to this segmentation.
Mobile Robot Navigation and Localization (MRLS)	Encompasses a set of tools for allowing a mobile robotic platform to autonomously navigate in cluttered environments. It is endowed with easy interface to set-up the robot map and navigation paths. It can be used on mobile platforms with different traction system (differential, tricycle, omnidirectional) and for localization it can resort to both artificial markers and natural contours. It also includes an algorithm for both decentralized and centralized multi-robot coordination solutions.
Weld Joint Localization (WJL)	System able to compute, using 3D vision, the real points that compose a welding trajectory, through the detection of the intersection of the faces of a joint. It is intended to improve the fast-teaching modes in human robot interaction, increasing the accuracy of learned poses to increase the welding quality.
Localization for Cut Openings (LCO)	Allows a collaborative robot to localize itself inside a block using 3D vision and a CAD model of the zone. The robot will perform a local scanning of the environment and with an approximate location provided by the worker.
Recognition of Parts for Bin Picking (RPBP)	Provides a tool, based a 3D vision system to preform bin-picking of parts with unknow CAD models, different shapes, weights, and dimensions. The solution detects different bins, where the parts are being stored. After the selection of the bin to focus on by the operator, the vision system returns a grasping point to the

	robot. This point is determined by the Centre of Gravity (CoG) of the part, which is calculated based on its material and contour detected (including holes).
Quality Assessment of Process	Enables a robotic arm to perform a quality assessment of the welding process performed by the robot, by providing feedback on the result of the welding performed in each path. The algorithm detects the outcome of the process to be monitored using a 3D vision system. The robot follows the welding path line and evaluates the effectiveness based on some predefined metrics.

Special attention was given to the software interfaces of the different PPM submodules to guarantee their interchangeability between robotic solutions and application needs.

2.2. Hand Guiding Module (HGM)

The Hand Guiding Module refers to a method of controlling robot's movements by an operator, typically through a hand held device, which is placed on the robotic end effector. The system evaluates the forces and torques applied by the operator onto the end effector either via a F/T sensor or motor torques evaluation, leading to an intuitive physical manipulation of the robotic arm. In particular, hand guiding offers significantly reduced programming time, as the operator can adjust the robot's movements in real-time based on the applied force, while also enabling the adaptability and the reconfigurability of the production point via the easy robot programming since it eliminates the need for the operator to program the robot when a change on the cell occurs or when a task is added in the process.

In the context of Mari4_YARD, two manual guidance approaches have been developed and integrated. The first includes a F/T sensor where onto it the whole tool is attached; after the system is calibrated once to compensate the forces applied from the tool payload, the operator can freely move the robot around the workplace, making it ideal in cases the operator wants to teach the robot where it should perform welding operations, or manipulate the robot in general. The second approach incorporates a F/T sensor installed on top of the robotic tool, thus enabling the gripper to grasp different and unknown weights that are also eccentric to the robot’s flange.

2.3. Control orchestration and planning (COP)

The COP module consists of a combination of human-centred tools that provide intuitive interfaces for monitoring and orchestrating the operations of flexible robotic systems. Two distinct levels of orchestration and planning were addressed: (i) the robotic domain, where a local orchestration module is responsible for the supervision of the execution of different modular robotic abilities, given the issued production task; (ii) the production domain where is needed coordination of the operation of multiple entities of different typology, i.e., robotic systems and workers.

In Table 2 the sub-modules that were developed are summarized. Their detailed description can be found in D2.2.

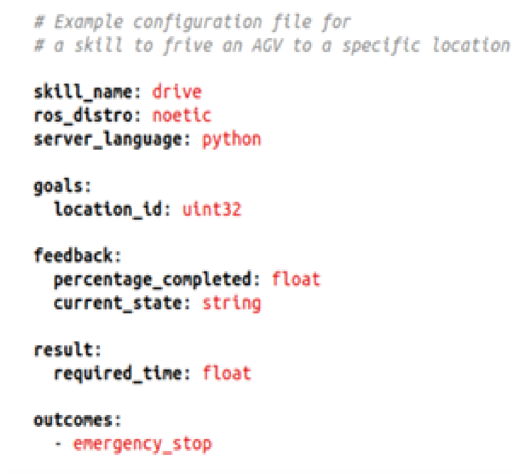
Table 2. COP sub-modules developed under WP2.

PPM sub-modules	Domain	Brief Description
Task Manager (TM)	Robot	The TM is a collection of ROS packages responsible for the orchestration and execution of the robotic tasks. It translates production tasks issued in the industrial standard SCXML (State Chart XML) notation to a structure intelligible and executable by robotic systems (i.e., robotic Skills), built based on the concept of ROS Actions. TM is also responsible for capturing operational data from the robotic system, to be integrated and processed in the upper layers of the manufacturing stack.
RoboGraph	Robot	The RoboGraph is a module based on ROS that allows to define sequences of tasks in a graphic way. The tasks are defined as Petri Nets, where in the places we can publish topics, and in the transitions, we can have as conditions a combined of subscribers where has evaluated the content of topics.
Production Manager (PM)	Production	The PM function is to assign and send one or multiple robotic tasks to be executed by one or more robots. These robotic tasks are a State Machine XML (SCXML) file, defining one more robotic Skills and how the robot executes them. Furthermore, this web-based module enables an intuitive programming environment, data visualization and operational supervision.

2.4. Human Robot Interaction Module (HRIM)

The HRIM module consists of a set of tools to expedite the programming and interaction with the robotic system.

PPM sub-modules	Brief Description
HRIM	HRIM is responsible for the safe and efficient interaction of the operator with the robot. Specifically, to support operators in robot programming Microsoft Hololens 2, Hololight Stylus XR and Voice commands using the mixed reality headset are utilized to teach the trajectories and the paths that the robot should execute minimizing programming time and making the process more accurate. HRIM also provides the operator the perception of the cell, including information such as the plan of the robot, safety zones visualization and notification for zone violation. The intuitive programming interface also allows the operator to modify paths and have full control of the trajectories that the robot should follow to perform welding operations.
AR Based Interface Submodule (ARIS)	AR system consists of a set of hardware and software designed to improve human-robot interaction and safety in collaborative workspaces. Through Microsoft HoloLens 2, the system monitors the operator's hands and head on the shop floor, allowing the collaborative robots to move concurrently with the operator as long as they maintain a pre-determined

	distance apart. The safety system described complements existing protective measures by creating a personal protective equipment that accompanies the operators throughout the entire shopfloor. The AR system also allows for easily programming production tasks through a codeless method in which the operator manipulates a holographic robot and programming panel.
Robotic Skills	The robot abilities, or behavior, can be encapsulated into Skills, built based on the concept of ROS Actions. Skills are based on three different but related ROS packages. This paradigm allows for the reusability of code, increases code maintainability, and eases the development of complex behavior by following a *divide-to-conquer* approach. Robotic skills consist of modular software units responsible for the execution of single and generic actions (e.g., open a gripper, close a gripper, or move a mobile manipulator to a specific point in space). To easy the creation of new SKILLS a SKILL generator was developed, in order to automate the generation of complete and operational robotic skills' skeletons. To use the Skill Generator, it is mandatory to create an *YAML* configuration file containing information about the skill to be created, as shown in the following Figure:  **Figure 1. SKILL Configuration file**

Robotic Skills

```
# Example configuration file for
# a skill to frive an AGV to a specific location

skill_name: drive
ros_distro: noetic
server_language: python

goals:
  location_id: uint32

feedback:
  percentage_completed: float
  current_state: string

result:
  required_time: float

outcomes:
  - emergency_stop
```

2.5. High Payload Robot (HPR)

The high payload robot that was utilized for the manipulation of heavy payloads and the final welding operations, having as a requirement to be safety rated for speed and separation monitoring based human-robot collaboration scenarios. To comply with the aforementioned requirements the robot that was used is the COMAU NJ 130 – 2.6, with the C5G safe controller option, which has integrated all the required safety systems to enable speed and separation monitoring. The safety option includes safe signals exchange with a safety PLC such as:

- safe workspace division for the robot to restrict robot motion outside of it.
- safe axis 1 position segmentation
- safe speed modulation

- forbidden zones to prevent robot accessing them.
- emergency stop

The specifications of the robot are presented below in Table 3:

Table 3 High Payload Robot technical specifications - COMAU NJ 130 – 2.6

High Payload Robot	
	
COMAU NJ130 2.6	
Wrist Payload	130 kg
Elbow payload	50 kg
Reach	2617mm
Degrees of freedom	6 rotating joints
Safety	Safe Robot
Repeatability	+/- 0.07 mm
Robot weight	1050 kg
Tool coupling flange	ISO 9409 - 1 - A 125
Motors	AC brushless
Position measurement system	with encoder
Total power installed	8 kVA
Assembly position	Floor

2.6. Portable Robot (PRO)

Portable robot concept refers to small/medium sized robots that can be easily held and carried by one or two operators because their reduced dimensions and weight, but large enough to accomplish with the tasks they are expected to do, like welding metal plates joints or cutting metal plates inside a vessel block, where confined spaces are very usual, and are very harsh environments to operators to work within them.

In this project development, two collaborative robots were chosen to work within the conditions above mentioned, and to help operators with the robots' deployment inside a vessel block, both robots can be placed onto the processing area with a magnetic base attached to them, which will provide a fast system to fasten them firmly to the vessel surfaces.

The first robot is an UR10 robot manufactured by Universal Robots, with a weight of 28,9kg and a maximum reach of 1300mm. It has a payload of 10kg, enough to hold an arc welding torch or a plasma cutting torch. The specifications of the robot are in Table 4:

Table 4 UR10 robot technical specifications

UR10	
Wrist Payload	10 kg
Reach	1300mm
Degrees of freedom	6 rotating joints
Pose Repeatability	+/- 0.1 mm per ISO 9283
Robot weight	28.9 kg
Footprint	Ø190 mm

Control Box Size (W × H × D)	462 mm × 418 mm × 268 mm
IP classification	IP54
Power consumption	350 W
Power supply	100-240 VAC, 50-60 Hz
Temperature	0-50 °C

The second robot is an Elfin5, by Shenzhen Han's Robot manufacturer, with a weight of 25kg and a maximum reach of 800mm. It has a payload of 5kg, not too much but enough to hold an arc welding torch. The specifications of the robot are in Table 5:

Table 5 Elfin5 robot technical specifications

	
Elfin5	
Wrist Payload	5 kg
Reach	800mm
Degrees of freedom	6 rotating joints
Pose Repeatability	+/- 0.02 mm

Robot weight	25 kg
Control Box Size (W × H × D)	445 mm × 319 mm × 360 mm
IP classification	IP54
Power consumption	180 W
Power supply	200-240 VAC, 50-60 Hz
Temperature	0-50 °C

The magnetic base to fasten the robots above mentioned, firmly during perception and processing is a MAGSWITCH COBOT MAGBASE equipment which can handle loads of almost 500kg (Nominal Maximum Shear) and 2250kg (Nominal Maximum Breakaway Force). The specifications of the magnetic base are in Table 6:

Table 6 Magswitch for UR10 technical specifications

<table>
<tr><td colspan="2"></td></tr>
<tr><td colspan="2">**MAGSWITCH COBOT MAGBASE UR10**</td></tr>
<tr><td>Nominal Maximum Breakaway Force</td><td>2250 kg for 19 mm thickness plate
1758 kg for 6 mm thickness plate</td></tr>
<tr><td>Nominal Maximum Shear</td><td>494 kg for 19 mm thickness plate</td></tr>
<tr><td>Full Saturation Thickness</td><td>12.7 mm</td></tr>
<tr><td>Net Weight</td><td>30 kg</td></tr>
</table>

Individual Magnetic Pole Footprint	71 mm x 296 mm
Overall Magnetic Pole Footprint	381 mm x 296 mm
Recommended safety factor	5:1

2.7. Mobile Robot (MRO)

In Figure 2, it is presented the mobile robotic platform explored in the context of the Mari4_YARD project. It is a mobile manipulator endowed with: (i) an omnidirectional mobile robotic platform, (ii) a collaborative robotic arm (UR10), (iii) a 3D sensor (Photoneo) and (iv) a Robotiq gripper 2F-140. In two opposite corners of the robotic platform, two safety lidar are installed. With this sensing set-up, it is possible to create safety zones that cover the entire 360º area around the mobile platform, allowing the implementation of the seep and separation monitoring stop type of collaboration. These lidars are also used for robot localization purposes.

Figure 2. Mobile Manipulator from INESC TEC

Table 7 presents the main technical specification of the robot solution.

Table 7. Mobile Robot Technical Specification

Parameter	Value
UR10	
Weight	28.9 kg
Payload	10 kg
Reach	1300 mm

Repeatability	+/- 0.1 mm
Footprint	Ø 190 mm
Degrees of freedom	6 rotating joints
Safety	Collaborative Robot
Robotiq 2F- 85/140	
Number Fingers	2
Weight	0.9 kg / 1 kg
Payload	4.7 kg / 2.5 kg
Grip Force	20 to 235 N / 10 to 125 N
Stroke	85 mm / 140 mm
Photoneo S	
Scanning range	384 – 520 mm
Point to point distance	0.174 mm
Resolution (3D Points)	Up to 3.2 M
Scanning area at sweet spot	360 x 272 mm
Weight	0.9 kg
Calibration accuracy (1 σ)	0.050 mm
Mobile Platform Parameter	
Traction System	Omnidirectional
Weight	400 kg
Safety	2x Safety Laser Sick covering 360º degree
Maximum Speed	1 m/s
Payload	300 kg

3 PARTS ASSEMBLY USING WELDING PROCESS

3.1. Description of use case

The shipbuilding industry still relies on manual labour for positioning and welding of parts, leading to unhealthy working conditions for workers. Challenges for automation include varying part types and characteristics, confined spaces, lack of CAD and 3D models, and the need for human-robot collaboration.

Figure 3 Current assembly and welding process performed by operators manually.

In Mari4_YARD, a solution is proposed that includes a high payload collaborative robot for part positioning, assisted by AR and hand-guiding technology. The operator performs the welding task with AR assistance and AI based process perception. The robot picks up varying parts upon operator request with magnetic grippers, holds it while the operator fine-tunes positioning with force control, stabilizes it with a welding torch manually, and fully welds it with operator guidance via a mixed reality intuitive interface. More info is provided in D2.1. The demonstrator including all the discussed modules is accompanied by a video that is submitted with the deliverable. The video is available in this Link.

Figure 4 Test bed of the high payload collaborative robot cell in LMS premisses

3.2. Integrated Technologies

3.2.1. High Payload Robot (HPR)

For the scenario as described in Section 2, a high payload collaborative robot was chosen to cope with the requirements for SSM and heavy parts manipulation. Onto the robotic flange a gripper with multiple modalities was designed and implemented to enable a variety of parts' manipulation while also it can perform welding operations. Additional sensors such as vision sensors (RGB-D cameras) and F/T sensors have been integrated in the tool, to facilitate the other modules developed in Mari4_YARD, such as the PPM and the HRIM modules. In Figure 4 the testbed is presented.

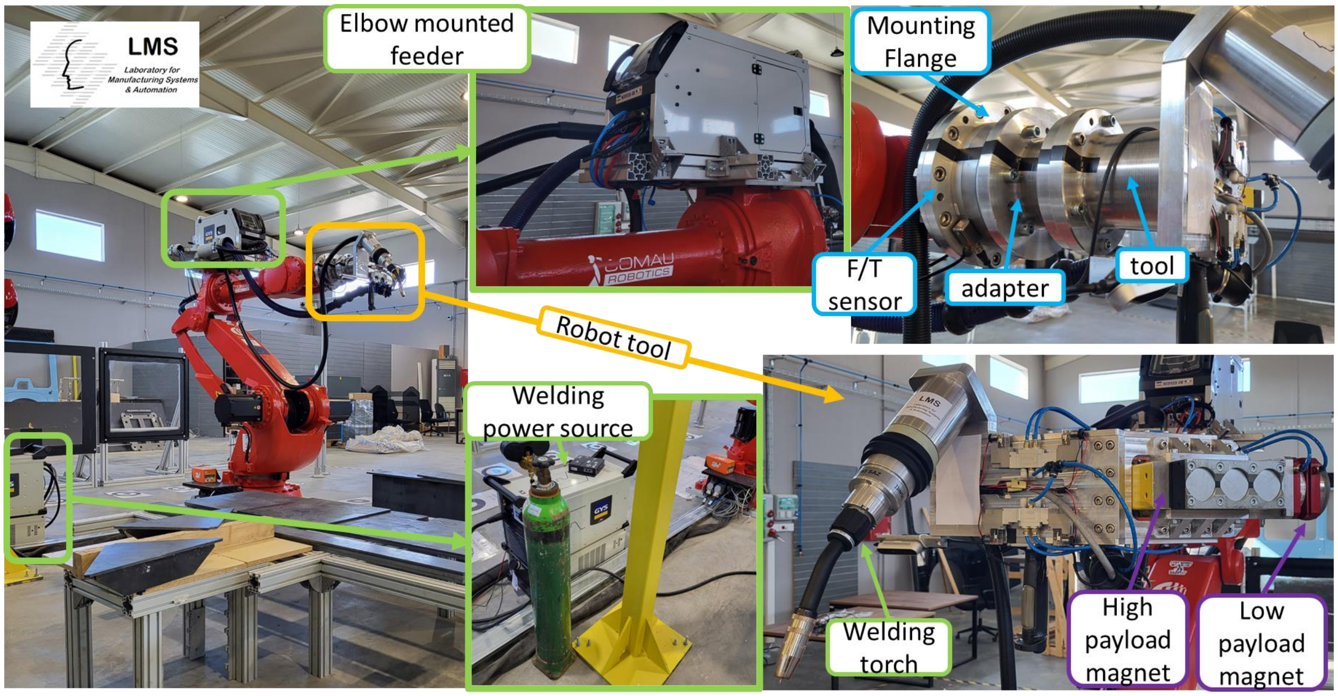


Figure 5 Robot tool system (welding system and magnetic gripper and F/T sensor)

The welding system that is used for the specific application is the GYS NEOPULSE 500 G which is a pulsed MIG/MAG welding power source with a separate wire feeder and cooling system. The power source is placed near the robot, and it is connected with the wire feeder which is place on the robot's elbow (Figure 5). The wire feeder is finally connected to the welding torch which is place on the end effector.

The robotic tool also consists of two, different magnetic grippers, that have different payload capacities. The smaller one is the Goudsmit TPGC100088 with advised working load 390 N, (under ideal conditions - with safety factor 3 acc. EN13155) and a max. tear off force of 1180 N and it is used to manipulate smaller workpieces around the workplace. The second one is the Goudsmith MS-8140393 with advised working load 2450 N, (under ideal conditions - with safety factor 3 acc. EN13155) and max. tear off force 7350 N and it is used to manipulate heavier workpieces around the workplace. Both devices are permanent magnets that are actuated via 2 5/3 electro valves thus preventing workpiece drop in case of unexpected power failure. The magnetic devices maybe seen in Figure 5 and Figure 6.

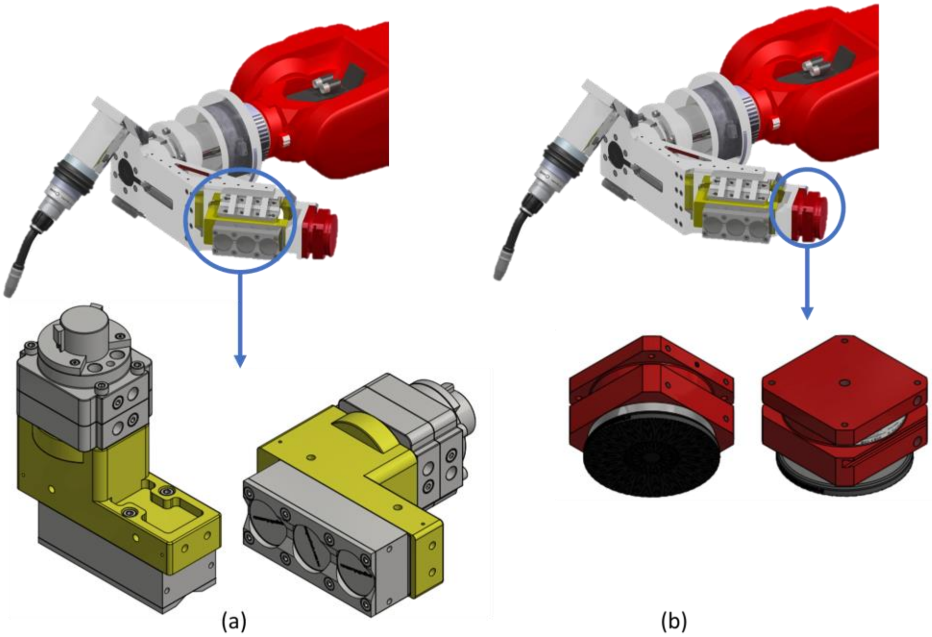


Figure 6 Magnetic devices onto the high payload robot gripper. (a) Goudsmith MS-8140393 (b) Goudsmit TPGC100088

The robot motion control, path planning and the actuation of the different mechanisms (such as grippers and welding machine) are performed based on the ROS [4] framework. A digital twin of the cell (Figure 7) is implemented, including the different parts that are placed around the workplace (fixtures, workpieces etc.). The motion planners of MoveIt! [5] were used for the autonomous navigation in fixed positions such as home position after welding operations or a couple of neutral positions that the robot reaches, after for example it has grasped a workpiece to then enable manual guidance for the operator in an accessible and ergonomic robot position that does not poses any danger for the human.

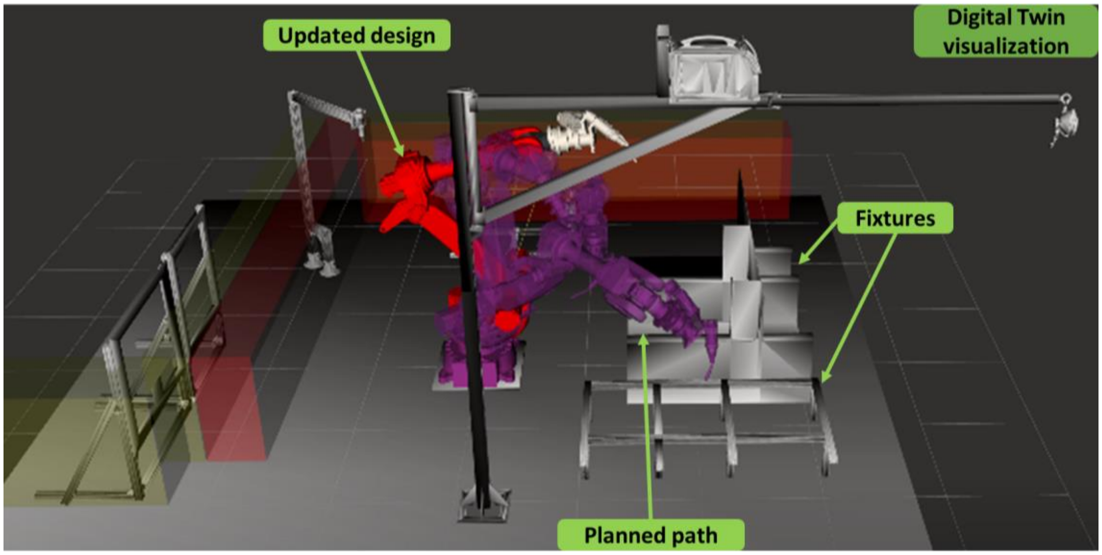


Figure 7 ROS implemented digital twin visualization in RViz [6]

The commands that are used to move the robot or actuate the electro valves via the robot are send using the dedicated ROS driver for the specific robot thus enabling full control for speed and acceleration modulation (as for example required for welding), trajectory execution command, and digital I/O reading and assignment [7]. In this context, in Mari4_YARD a similar ROS driver was implemented by LMS, for the GYS welding system using as a medium a Siemens S71500F PLC. The driver is oriented towards a service approach and uses two TCP/IP connections between the PLC and the ROS master to redundantly share the information/commands. The PLC is connected to the GYS welding power source communication module using Profinet fieldbus. This custom driver enables the handling the machine at its full potential, also allowing online welding parameters changing etc. In the figure below (Figure 8) the interconnection of the welding system is presented, between the robotic system, the PLC of the cell and the ROS master.

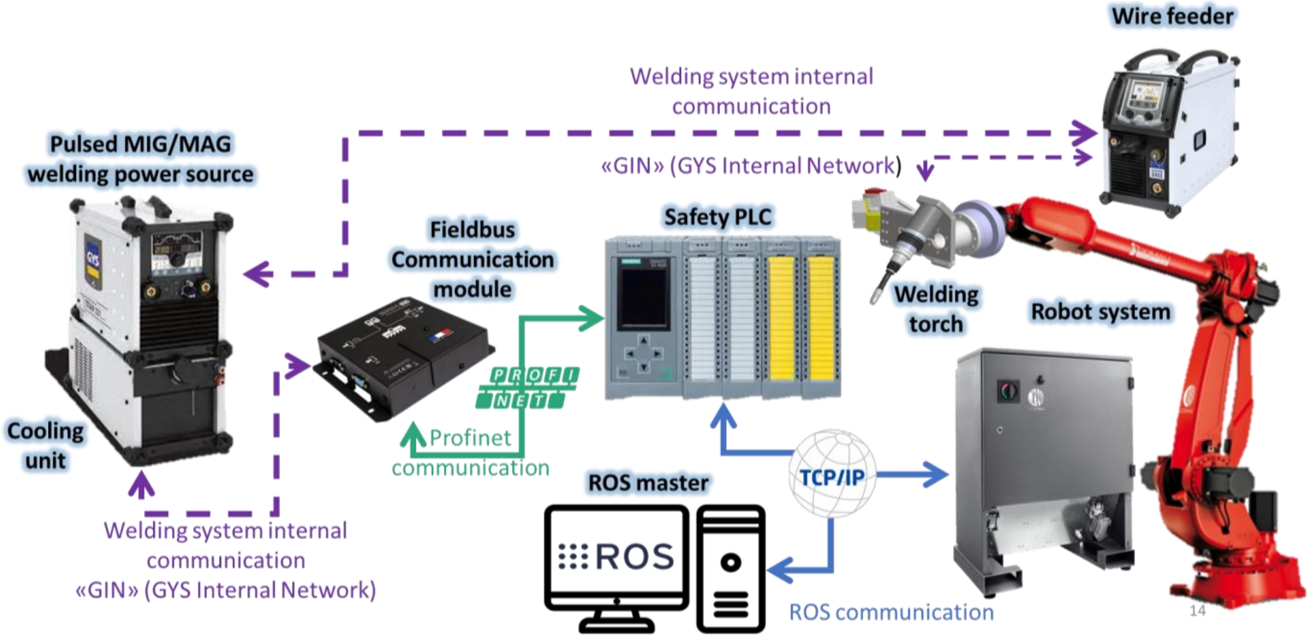


Figure 8 Welding system interconnection diagram.

3.2.2. Workspace Monitoring System

WMS solution incorporates a certified safety solution, the Pilz Safety Eye. By utilizing the Safety Eye's software, we have established multiple zones configuration using two types of safety zones: a warning zone (yellow) and a detection zone (red). If an operator enters the warning zone, the robot's speed is reduced. If the operator enters the detection zone, the robot stops to ensure the operator's safety. To resume the process, the operator must press a restart button located outside the cell to confirm that they have exited the cell. Additionally, according to the ongoing process and specifically the robot's pose related to the process, a different zone arrangement is activated to monitor zero access areas, thus changing they layout dynamically and allowing better workspace sharing among robot and human resources. The 3D camera system communicates with the robot via safety PLC, which communicates with the robot controller with Ethernet IP for the non-safety critical I/Os and with safety rated hardwired I/O connection for safety critical commands. More info can be accessed in D2.1. The overall solutions is presented in Figure 9

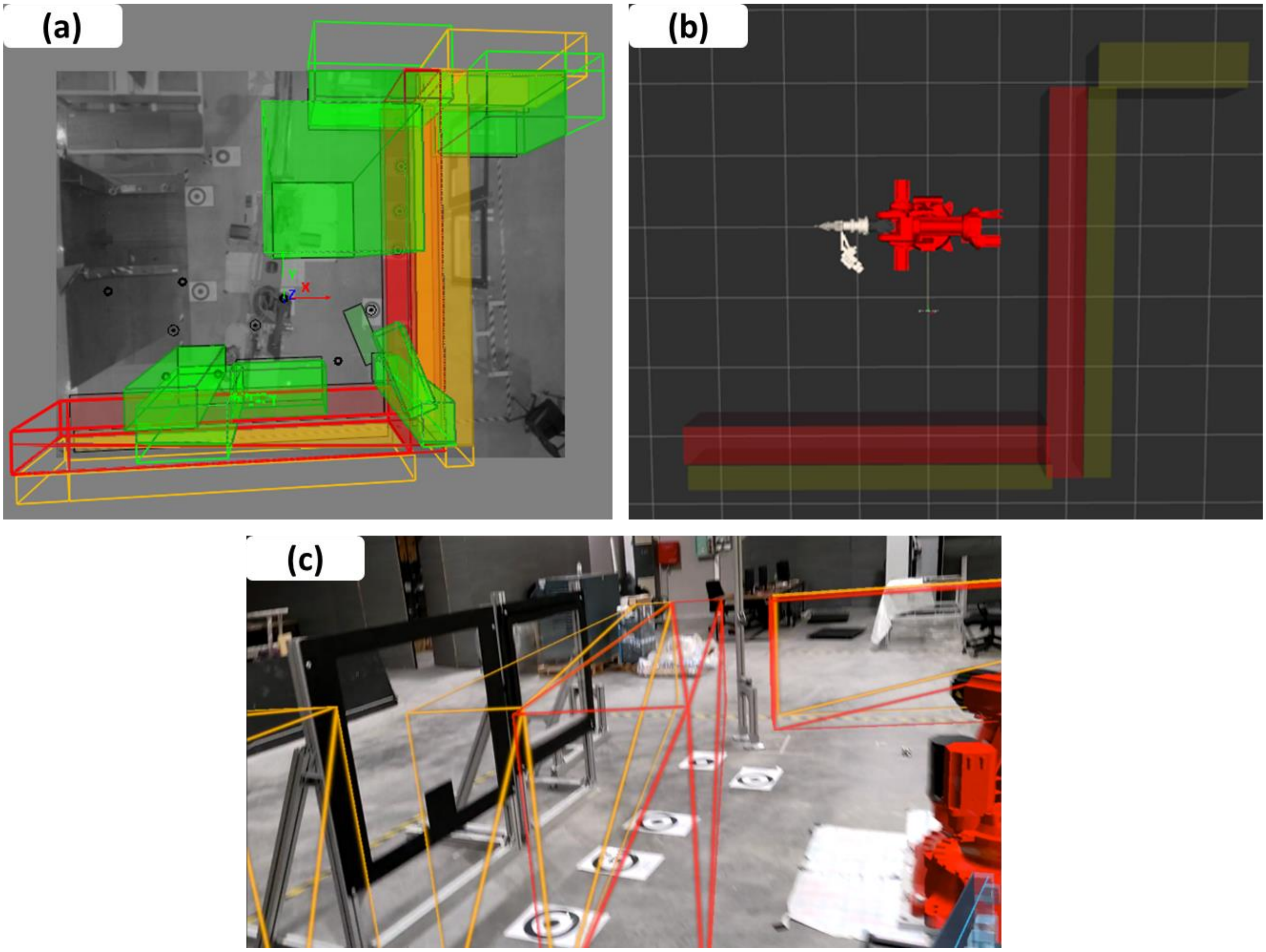


Figure 9 (a) SafetyEye Zones Configuration (b) ROS (RVIZ) zone Visualization (c) Hololens2 AR Visualization

3.2.3. Human Robot Interaction Module

Human and Robot Interaction (HRI) is a crucial aspect of the Mari4_YARD project. The project employs the Robot Operating System (ROS) and TCP/IP to facilitate seamless communication between the PC and the robot. LMS has created a Digital Twin of the cell at LMS premises using precise CAD models, as illustrated in Figure 10. RViz is utilized for visualizing the Digital Twin, which provides the opportunity to view the robot's trajectories in a virtual environment before executing them in the real world, thus reducing the risk of collisions. Additionally, the Digital Twin provides valuable information about the robot's TFs and safety zones visualization. The project also incorporates AR robot programming using Hololens 2 and Rosbridge server, which facilitates communication between the AR glasses and ROS. The AR programming feature enables the operator to use point and path teaching for easy robot programming, safety zones visualization with alerts when the operator violates a zone, and dynamic assignment and selection of parts.

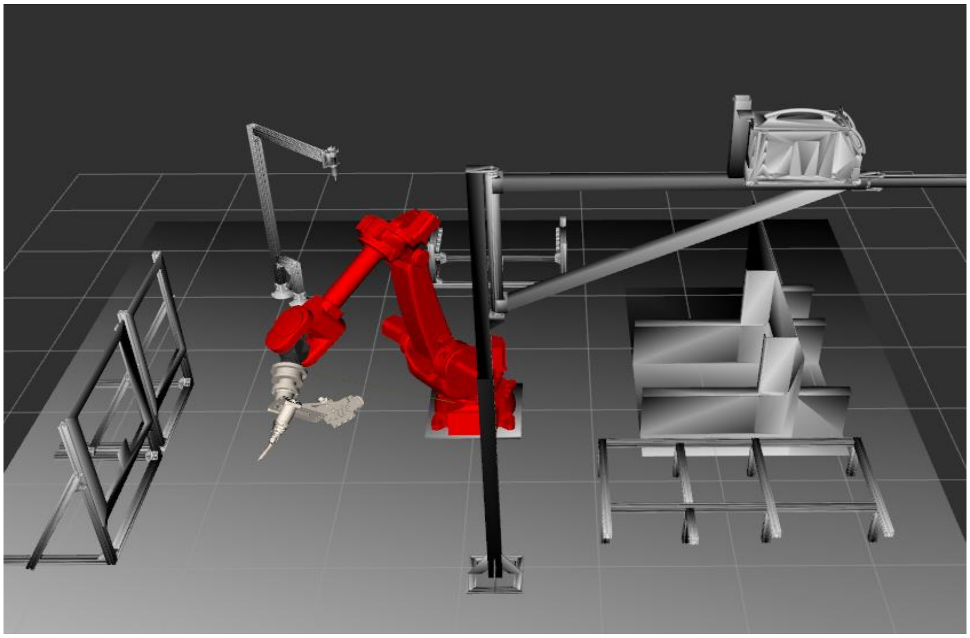


Figure 10 Robot digital twin visualization in RViz (ROS)

The path/point teaching is performed using Hololens2 and Stylus XR. The implementation consists of an application that offers the user the chance to design with the smart pen either dedicated points or paths that the robot should following and perform the desired welding, to complete the assembly (Figure 11).

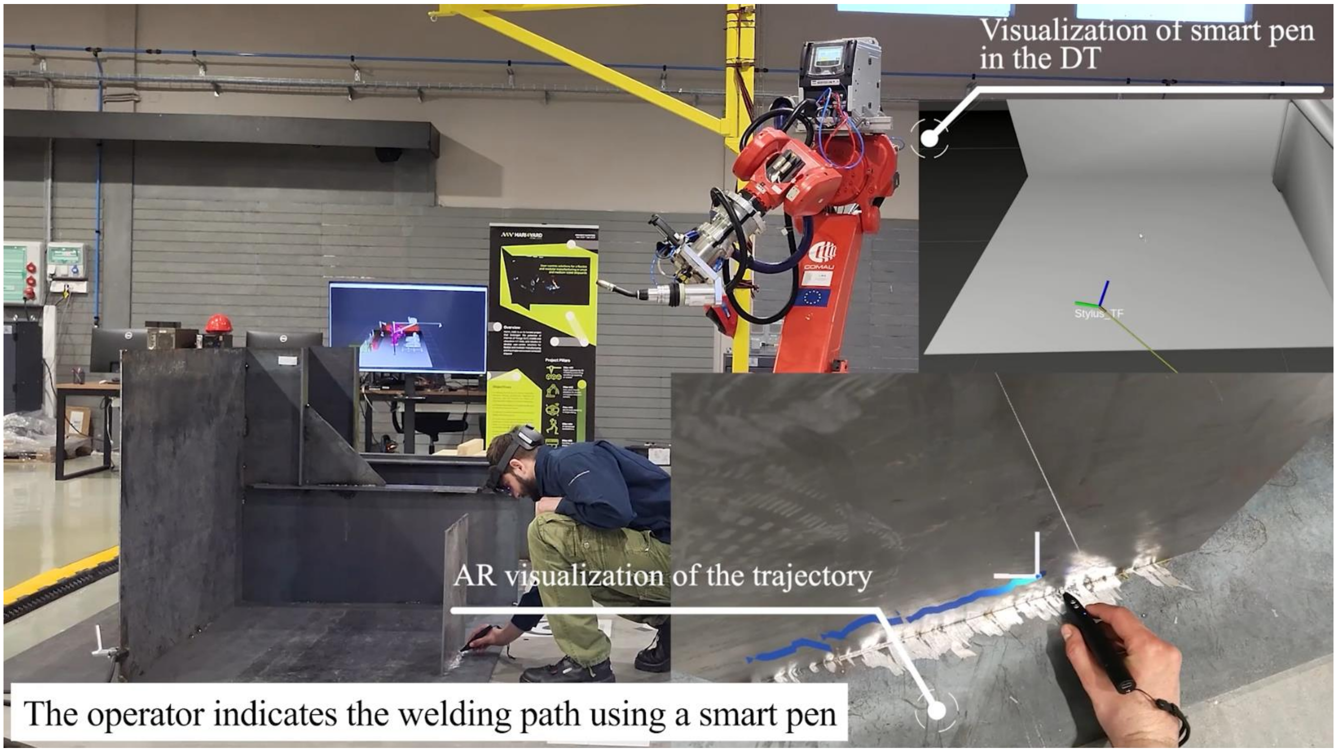


Figure 11 HRIM path teaching

Furthermore, in the context of HRIM, a Stylus welding trajectory correction feature has been implemented using edge detection in 3D space with a Depth camera (Intel Realsense D415). Once the Stylus Trajectory Correction is completed, the robot follows the trajectory and carries out the welding process on the intended part, thus preventing the inaccuracies that are a results of the AR path/point teaching interface.

3.2.4. Hand Guiding Module

In the Mari4_YARD project, hand guiding module (HGM) was tested in ABB and KUKA robots. It has been adapted for the robot COMAU NJ130-2.6 and its ROS controller developed by LMS. AIMEN has provided the docker image with the HGM, Schunk FTN-Omega191 force and torque sensor and the mechanised support to mount the sensor in robot tool.

To deploy this system in COMAU robot, following modules and data endpoints were required for its ROS integration, as shown in Table 8:

Table 8 Modules and data endpoints for HRIM

ROS Module	Interest topic name	Message type
Position Trajectory Controller	/position_trajectory_controller/command	trajectory_msgs::JointTrajectory
F/T Sensor driver	/schunk_ft_mini_driver/measures	geometry_msgs::WrenchStamped
Robot driver	/joint_states	sensor_msgs::JointState
Hand Guiding Module	/arm_controller/set_mode	std_msgs::Int8

COMAU robot control is executed in position commands, so we had to transform the velocity commands (original output from hand guiding module) into position, as we already did for KUKA implementation. Another crucial point is that communication rates and data provided must be at high rate. Specifically, sensor data must be at 200 Hz, robot driver at 250 Hz. Robot control commands are sent at 20 Hz.

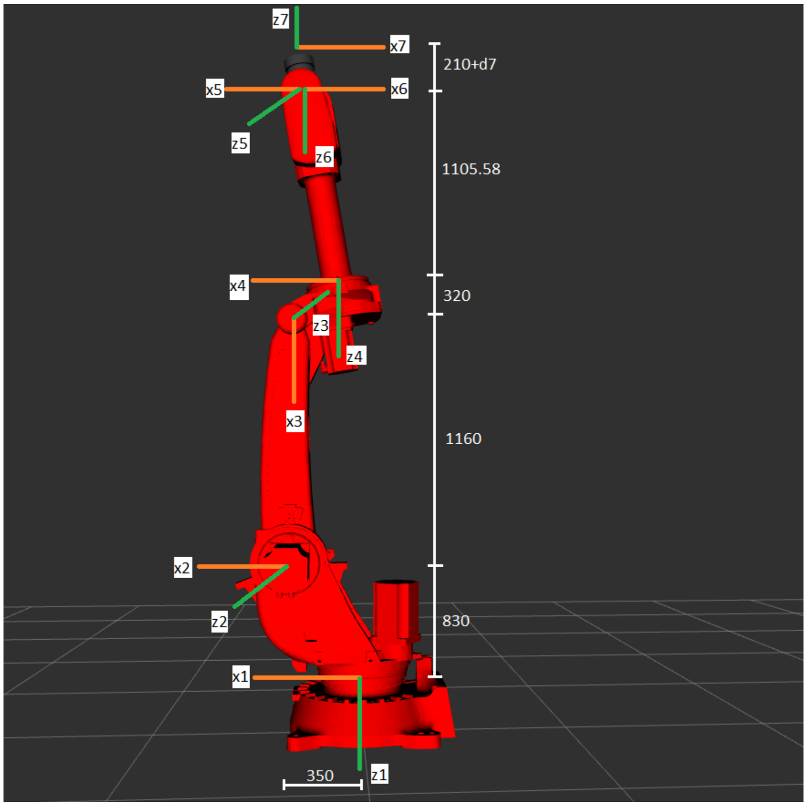


Figure 12. DH axis for COMAU NJ130-2.6. Units are expressed in mm.

Furthermore, robot configuration and sensor mounting are required to obtain an appropriate response of the sensor. Hence, Denavit-Hartenberg (DH) parameters, sensor positioning in robot flange coordinates and effort application point in F/T sensor reference system are defined. DH parameters are computed with Z axis following the positive movement direction of the tool and DH computing procedure. Considered axis are shown in Figure 12 and DH parameters table (Table 9) is gathered in.

Table 9 COMAU NJ130-2.6 DH Parameters, being d7 sensor length.

	ϴ (rad)	d (mm)	a (mm)	α (rad)
0	0	-830	-350	π/2
1	π/2	0	-1160	π
2	π/2	320	0	π/2
3	0	- 1105.58	0	π/2
4	π	0	0	π/2
5	0	- (210 + d_7)	0	π

Integration steps consisted of executing two steps to run the controller: calibration and controller parameter adjustment. For the sensor calibration process, we defined 6 positions to compute weight and centre of mass of the sensor by aligning each sensor axis with the gravity vector. Then, parameter adjustment was performed to move the robot by the operator with ease without applying a lot of effort.

Fulfilling all the previous requirements, HGM was deployed in LMS facilities in two different ways: with the sensor between the load and the wrist, and in isolated way. These two approaches were needed as in the second case we are working with unknown loads.

Fixed payload

In this case, sensor is mounted as it is shown in Figure 5 right upper corner. By this mounting, we can move the robot by running the HGM in an easy way with the correct controller parameters and robot configuration. This approach is really useful to move the robot for example for easy position programming.

Varying payload

When we start manipulating unknown loads, the mounting previously proposed cannot be used. If we had used the first approach, weight on the tool and centre of mass of the applied force will be displaced when we grasp the load, causing the controller to response wrongly. This is due to the fact that the controller calibration data is not corresponding to real configuration.

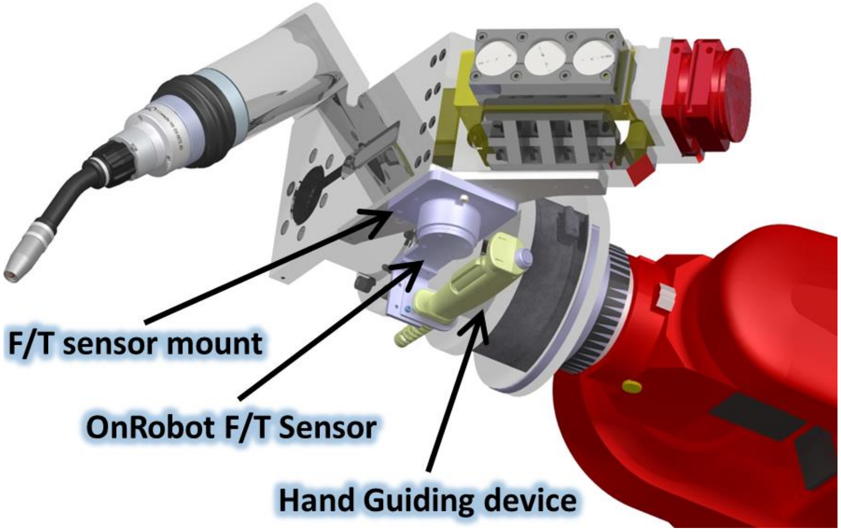


Figure 13 HGM hardware for varying payload force control

Therefore, sensor has been placed isolated from robot wrist (Figure 13). By using this setup, grasped load weight will not modify the centre of mass and we will only consider operator efforts to perform the HGM. We will have to modify DH last value, calibration file and controller parameters the first time we use it. Besides, when we grasp an element, we will have to reset sensor measurements to discard this weight and start the manipulation process.

3.2.5. Process Perception Module

The Process Perception Module (PPM) utilizes the librealsense2, OpenCV, and PCL libraries for point cloud calculation and processing. The ROS framework is implemented to facilitate data exchange between the submodule and the other modules. Initially, the RealSense camera streams are accessed and configured based on desired parameters and format. Two streams are accessed to receive the depth frame and colour frame for visualization purposes. Subsequently, a point cloud is generated and mapped to the coloured frame to match the points of the image. Using depth data, each point is placed in its original depth while maintaining its colour attribute. The point cloud is segmented based on the distance between each point to the other. For proximity search, the KdTree algorithm is utilized from the KDL library. The clusters generated are visualized in indicative colours to demonstrate differentiation. By computing the closest cluster, the nearest object to the centre of the camera can be accessed. The centroid of this cluster is then calculated, and its coordinates are used to determine the distance from the centre of the camera using depth data. Finally, the orientation of the cluster surface is used to generate the quaternion of the centroid for subsequent gripper guidance. The detection of the two different parts is presented in Figure 14 and Figure 15. More information regarding the PPM can be accessed in D2.2.

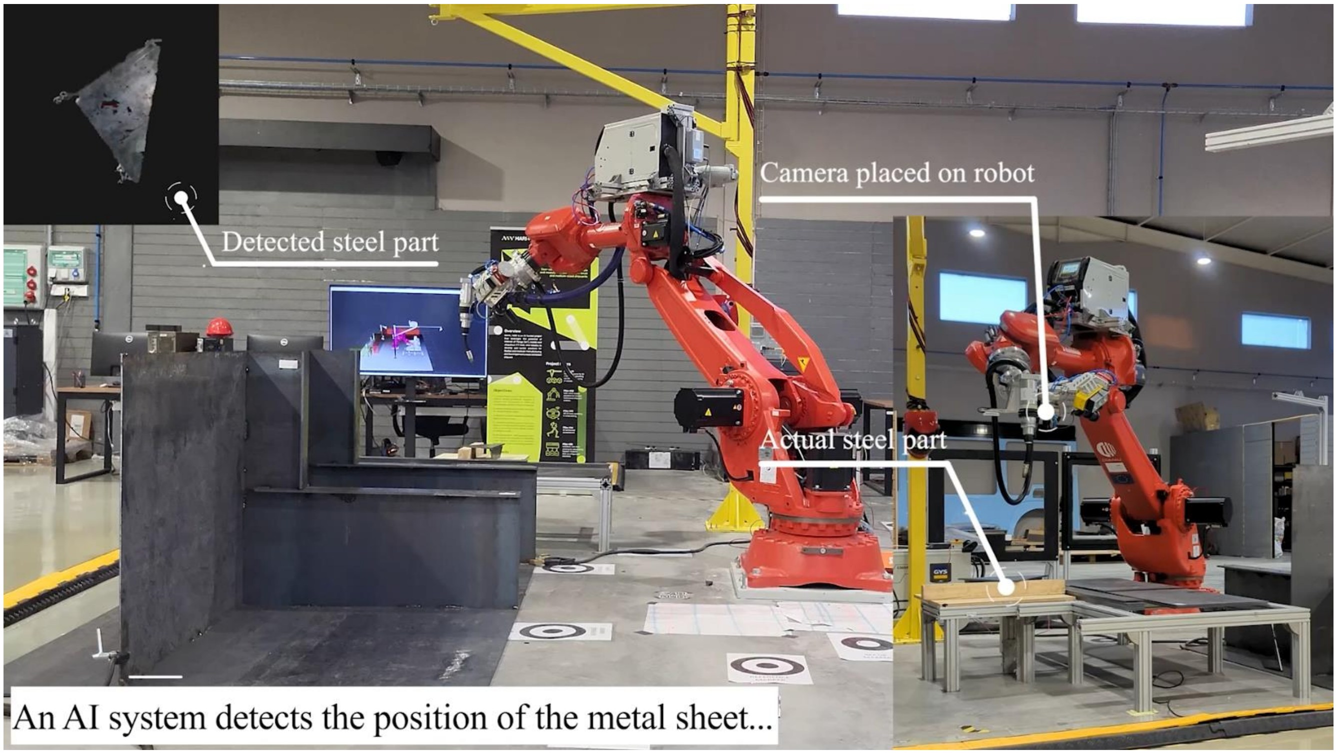


Figure 14 PPM estimates the CoG for the light/small-sized variant

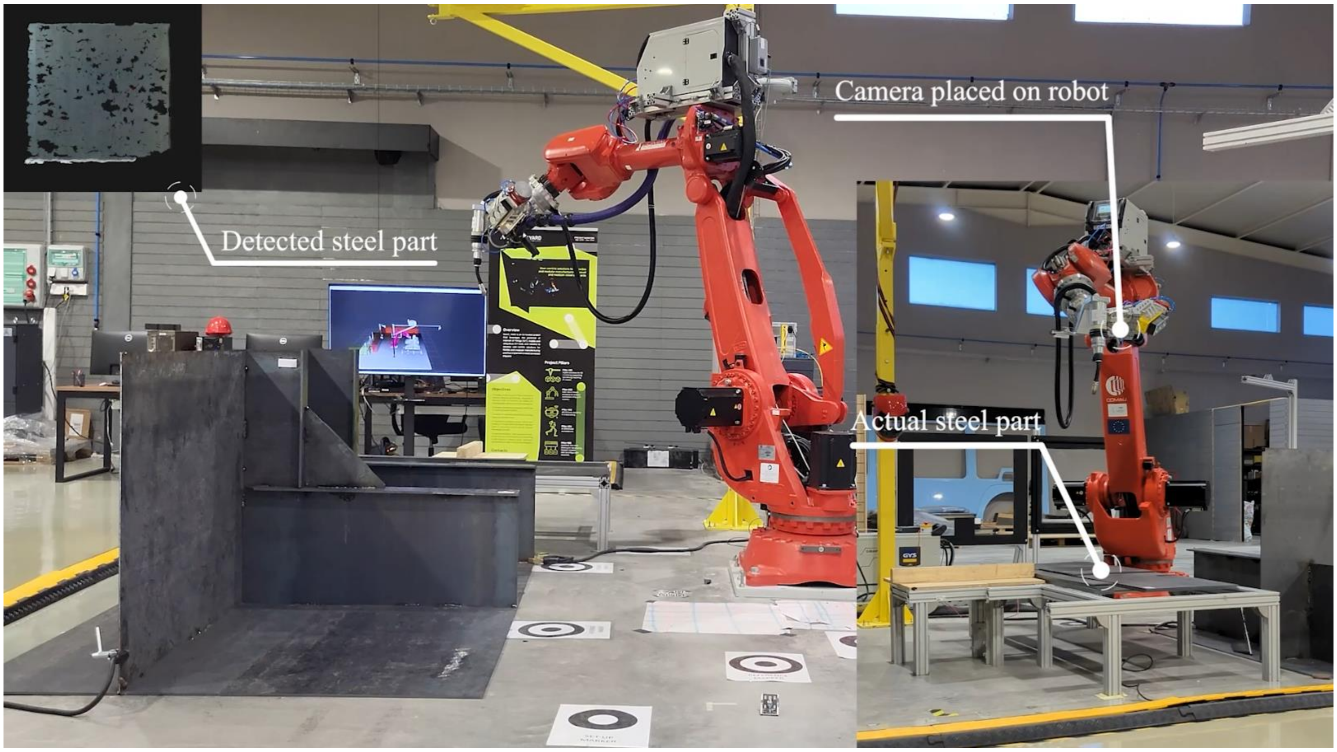


Figure 15 PPM estimates the CoG for the heavy/big-sized variant

3.2.6. Control orchestration and planning

There are two valid tools for control orchestration and planning the tasks to be executed: RoboGraph (AIMEN) and OSPS (INESCTEC). In this section we will demonstrate that it is possible to combine both by establishing communication between them.

Both tools are designed to work in ROS environments. OSPS is client-server based with synchronous communication through the ROS action library, where the server executes the task and the client launches the command; and RoboGraph is an application programmed in Java for modelling, executing, and monitoring Petri nets that can send and subscribe to any ROS message, including ROS action messages. This is the interface between the two tools and the point from which communication is established (Figure 16). Communication between an OSPS skills server and a Petri net is straightforward because, from the Petri net's point of view, it is exchanging messages with a conventional ROS node (the Skill Server) (Figure 17 right). That permits to reuse Skill Servers programmed for the OSPS with no modifications at all and the Petri net running as Skill Client (Figure 16).

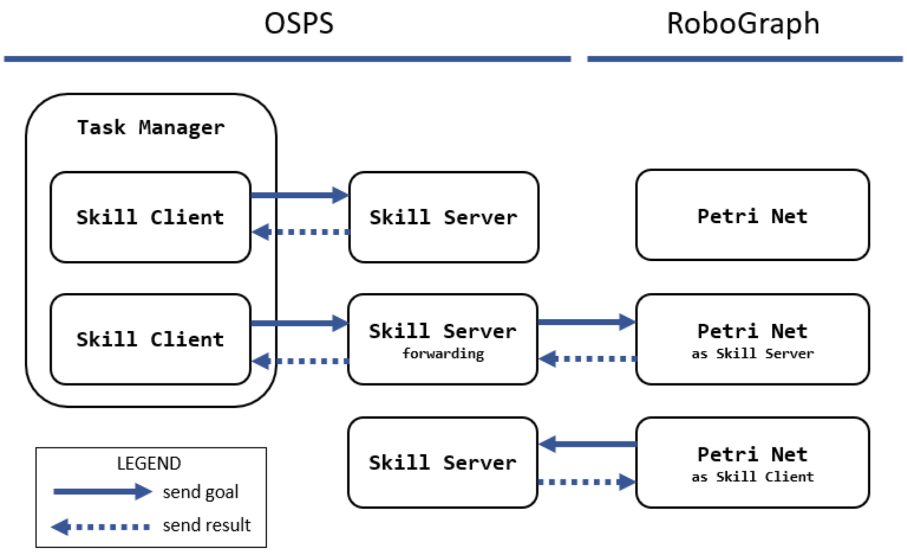
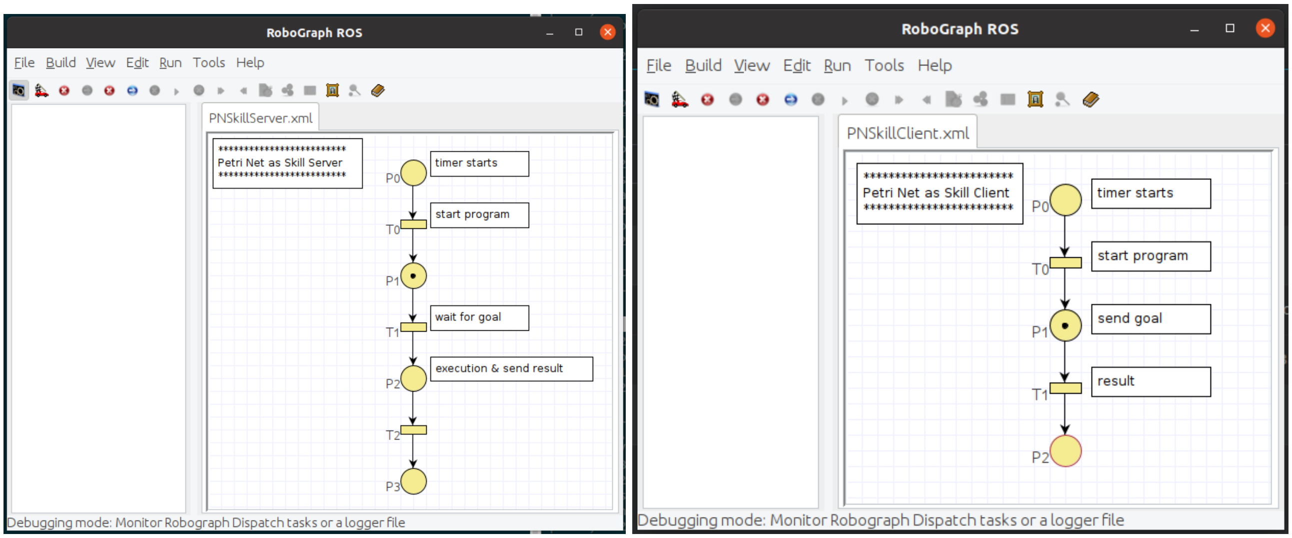


Figure 16. Different RoboGraph-OSPS integrations

During the course of this integration work, it was observed that it is not possible to establish direct communication between an OSPS Skill Client and a Petri net (Figure 17 left) without the presence of the OSPS Skill Server. This is due to the normal operation of the Task Manager from which the Skill Clients are run. It checks beforehand that the server to which each client wants to connect is running, and if it is not, the ROS action goal messages are not published, making it impossible to establish communication.

The solution to this problem is not to break the client-server pairing of the OSPS part and to program the server not to execute the task, but to act as a bridge between the messages to the Petri net so that it can send and receive them and execute the tasks. The Petri net runs as Skill Server, the Skill Server runs as a forwarding node for the Skill Client (Figure 16).



Debugging mode: Monitor Robograph Dispatch tasks or a logger file

Debugging mode: Monitor Robograph Dispatch tasks or a logger file

Figure 17. Examples of Petri Net as Skill Server (left) and Skill Client (right)

4. WELDING WITH COLLABORATIVE ROBOTS

4.1. Description

Welding with collaborative robots, aims to perform welding operations with high precision, leading to time consuming and high-quality welding for the needs of shipbuilding industry. In the specific scenario, a *Process Perception Module* developed in the Mari4_YARD project is used to detect and weld a joint, pushing the limits of existing machine vision technology. The deployment is achieved by utilizing, a UR10 and an Elfin 5 robot both magnet-mounted in the steel panels, making the solution portable and easy to integrate in different environments. The implemented vision algorithm utilizes an RGB-D sensor, the Intel RealSense d435. Besides, an HMI is used to visualize and configure the welding parameters of the HRC based welding solution.

4.2. Integrated Technologies

4.2.1. Portable Robot

To be able to cope with arc welding operations inside a vessel block, two types of collaborative robots were chosen. They are very similar in size, which is small enough to be handled and carried easily by operators.

The first robot is an UR10. It is able to manipulate loads around 10 kg attached to its tool flange. In this case, an arc welding torch was selected to perform welding tasks, besides this, a RealSense D435 camera was used to perform the perception tasks. these devices were attached to the robot flange using a designed and manufactured fixture.

The second robot selected, is an Elfin5, which is able to manipulate 5 kg on its tool flange. In this case it was attached an arc welding torch TBI RM 62G, and a Realsense camera by mean of a proper fixture to perform arc welding tasks.

Additionally, to be sure the robots were accurate enough to perform the assigned tasks, it was important to have a system to fasten them firmly and secure in the surroundings where the processes were to take place inside the vessel. Besides this, this system to fasten them, should be as much lighter as possible and quickly to deploy. The solution was to use a magnetic base attached to each of the robots' bases, able to be fasten it to any of the metal plates are made of a vessel block. With this solution, the overall equipment weight, robot + magnetic base, was not superior to 60kg, enough to be held and handled by a pair of operators.

Regarding this solution adopted, with the UR robot, a bunch of equipment was necessary to gather, and to integrate to perform the arc welding process. Within Table 10 are summarized all the equipment used:

Table 10 Equipment used for Welding with collaborative robots' technology.

Arc welding process equipment		
Equipment Name	Type of equipment	Weight
Robot	UR10	28,9 kg

Magnetic base	Magswitch cobot UR10	30 kg
Welding power	Fronius TPS 400i with cooler	36.5 kg
Welding torch	Fronius MTB 400i	3.16 kg
Welding wire-feeder	Fronius WF 25i	4.8 kg
Welding communication board	RI MOD/i CC ModBus TCP-2P	< 500 gr
Perception camera	RealSense D435	< 600 gr

In Figure 18. the devices during the trials done inside the vessel block are presented.

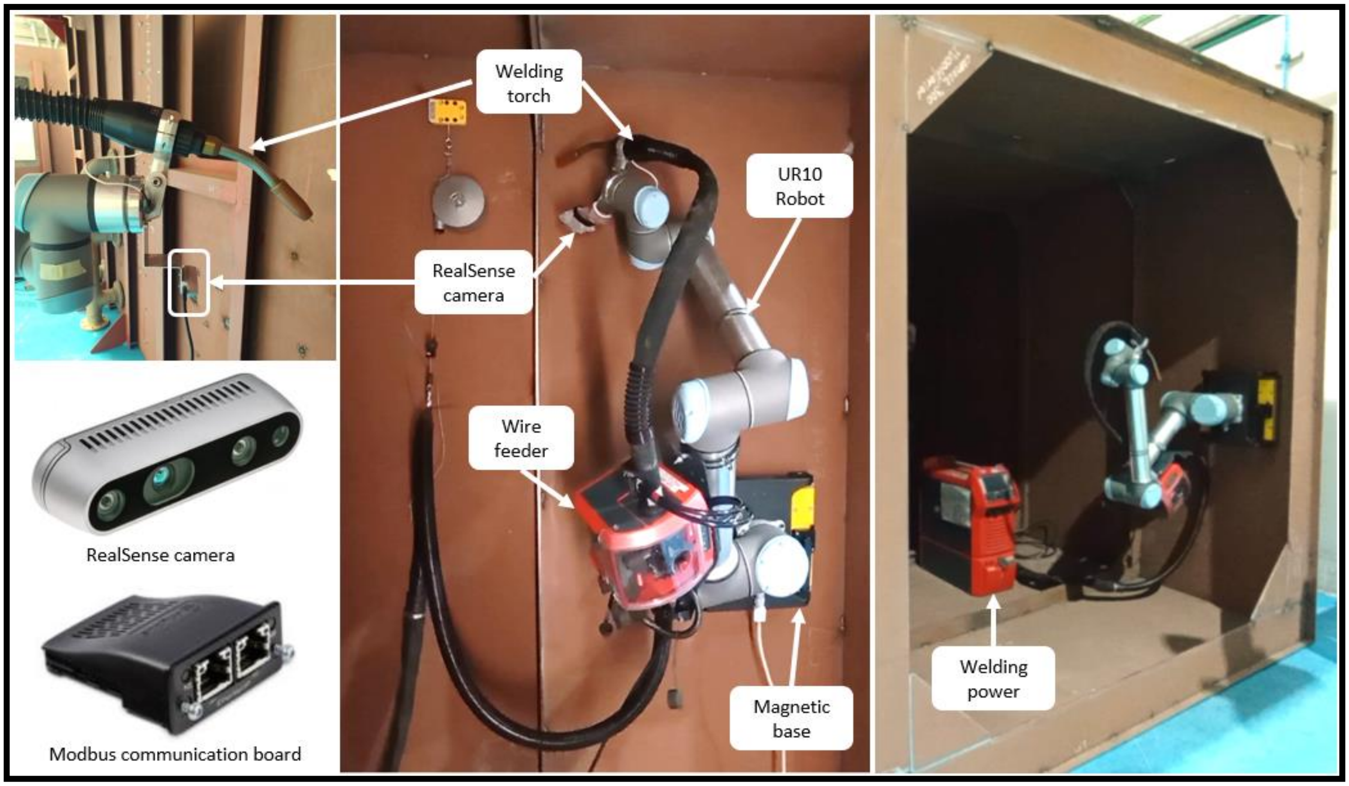


Figure 18. Devices used to perform welding joint inside the vessel block.

A PC master with ROS executing is the responsible for communicating with these devices, through different interfaces like USB3.0 to communicate with the camera, or TCP/IP to communicate with the robot or the welding power as shown in Figure 19.

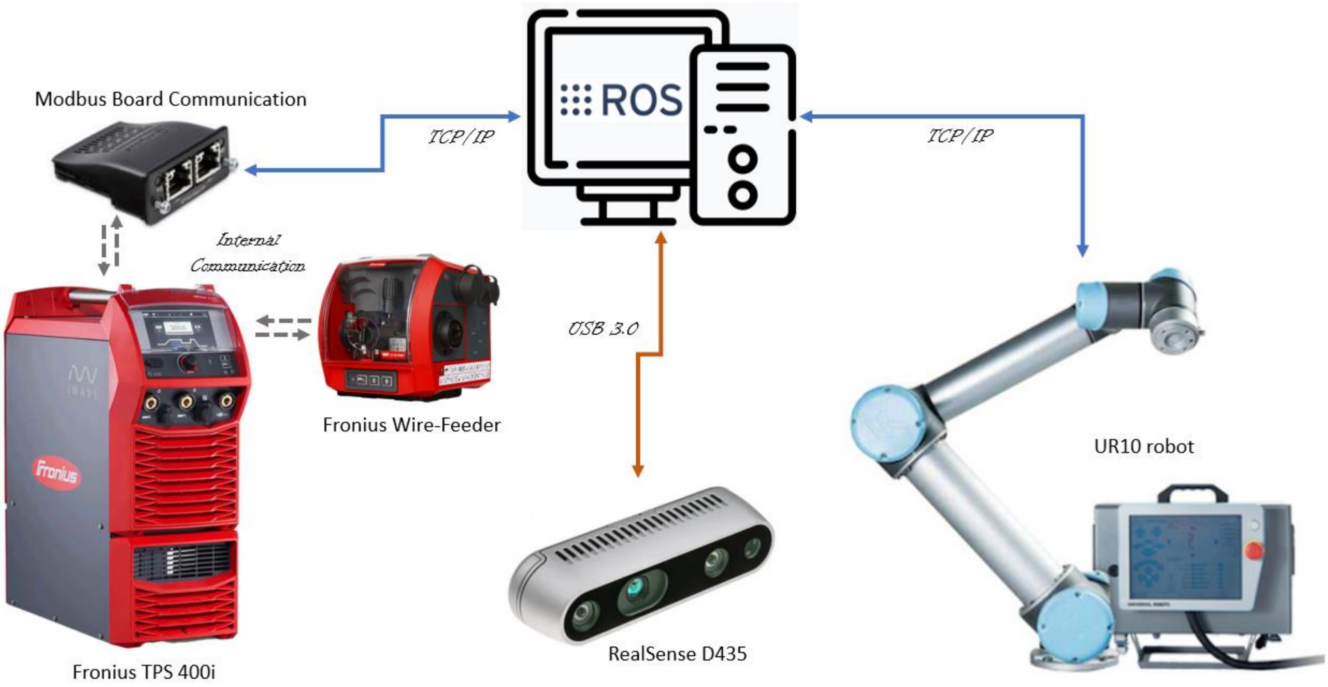


Figure 19. Communication diagram for welding application.

Regarding the solution implemented with the Elfin5 robot, several pieces of equipment were required to be gathered and integrated for the arc welding process. Table 11 below provides a summary of all the equipment utilized:

Table 11 Arc welding process equipment

Arc welding process equipment		
Equipment Name	Type of equipment	Weight
Robot	Elfin5	25 kg
Magnetic base	Magswitch cobot UR10	30 kg
Welding power	Megmeet Artsen Plus 500 QR	57.5 kg
Welding torch	TBI RM 62G	2.92 kg
Welding wire-feeder	Megmeet Robotic-wire-feeder	4.6 kg
Welding communication board	EthernetIP Communication module	< 500 gr

Perception camera	RealSense D435	< 600 gr

And next picture (Figure 20) shows all these devices during the trials done inside the vessel block.

Figure 20. Devices integrated inside the vessel block for the trials

A PC master with ROS executing is the responsible for controlling the Perception System it communicates with the camera through USB3.0 and wirelessly with the GUI interface. The GUI interface runs of a tablet. In the GUI application the welding program is defined. The application sequencies the execution of the program by communicating wirelessly with the Elfin5 robot and with the power source and the ROS master PC that is in charge of the Perception System.

4.2.2. Control orchestration and planning

For control orchestration and planning of the welding tasks, the robotic solution relies on RoboGraph. It is a Java application able to design and execute Petri nets that communicate with ROS nodes publishing and subscribing to topics. Petri nets are a powerful tool to model, design and analyze distributed, sequential, and concurrent systems. The user can model perception and welding tasks using the GUI (Figure 21) and saving in a xml file for later execution by a Dispatcher. For debugging and tracing, there is a monitor that shows the state of the current running Petri nets.

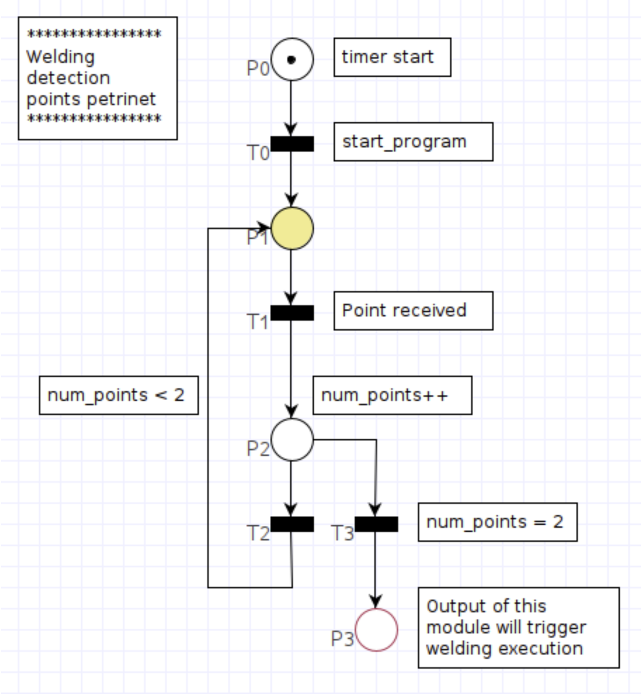


Figure 21. Example of Petri net for perception and welding

RoboGraph can work in three different modes: Editor, Monitor and Play Logger. In editor mode, the user can create new tasks by pick & place different components that conforms a Petri net. The elements are Places, Transitions, Arcs and Marks. Actions (associated to Places and Transitions) are automatically generated by RoboGraph in a list where the user can select them and then define their parameters. Actions can start a timer as well. Conditions (associated with transitions) are also shown in a list generated by RoboGraph. A condition can be the arrival of an ROS message event, a condition on some ROS message parameter, any logical expression on several parameters over the same or different ROS messages or the value of a timer. Global variables are used to store data, share conditions and events in different places or transitions.

In Monitor mode, Dispatch oversees the loading of the xml file that contains the Petri net, executing and cancelling the Petri net under request, publishing messages associated with the places and transitions, and management of the event synchronization.

In Play Logger mode each time a change in the status of a Petri net (start, stop, evolve) or in the waiting queues (new requests added or removed) is produced, a new ROS message reporting that change is issued for GUI monitor mode and stored in the log file.

4.2.3. Human Robot Interaction Module

The Human Robot Interaction Module allows interfacing all aspects of the welding system, between them:

1. Complete robot manual controls
2. Complete power source manual controls

3. Definition of a welding program
4. Execution and monitorization of the welding process
5. Simulation of the welding process to check program welding program correctness.
6. Optionally, interfaces with the Perception System to help the operator take the welding points with precision.

The Human Robot Interaction Module is multilingual and written in the C# programming language. Currently, it works only on Windows.

The module communicates with the following systems:

1. The robot: uses wireless communication and sending the control messages using an Ethernet communication protocol supported by the Elfin5 robot.
2. The power source: using wireless communication though EthernetIP.
3. The Perception System computer: using wireless communication.

To explain the application, we are going to focus on its two most important screens:

Welding execution screen

This screen allows for a flexible and intuitive welding operation (Figure 22).

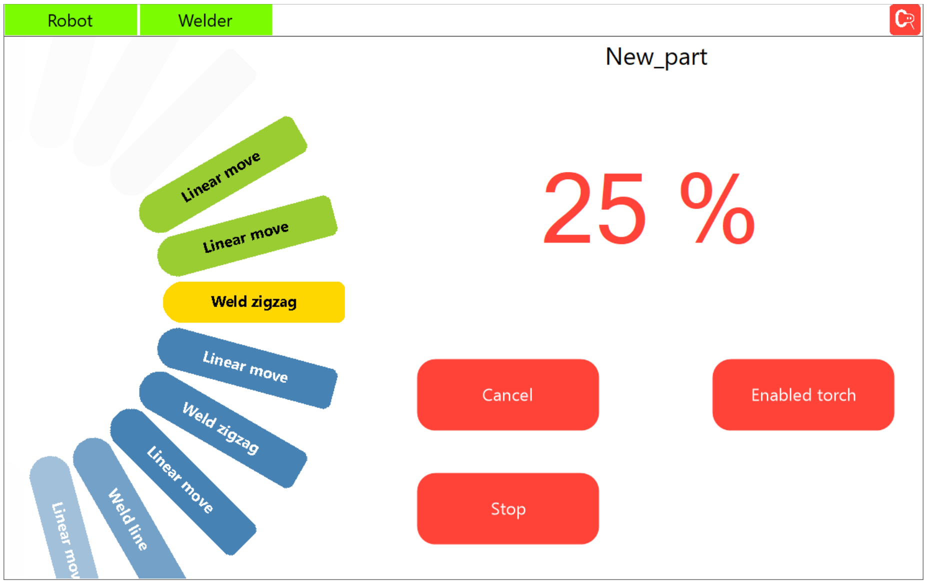


Figure 22. Welding Execution screen

On the top bar of the screen, we can see the current state of the Robot and the Welder. If the state is faulty we can reset the state of the devices by tapping on the respective controls.If during the welding process, an incident happens an error message will appear in the right part of the top bar. On the left part of the screen, it appears the instructions that compose the welding program and its state in which you can see what instructions have been executed, the current one in execution and the next welding instructions to be executed. In the right part of the screen, we can see the percentage of the welding program that has been completed. Lastly, in this area, we have the buttons to start the welding process, stop it, or enable and disable the welding torch.

Welding execution screen

This screen allows for the edition of a welding program using a touch interface Figure 23.

New_part

Name	X	Y	Z
zigzappose	-346.5	-32.90	165.21
linear	-377.8	41.52	158.64
P1_DR	-355.9	-97.16	17.63
P1_UR	-349.6	-101.5	273.98
P2_DL	-356.5	168.74	27.25
P2_UL	-355.6	169.48	274.77
P3_L	-359.7	169.95	19.11
P3_R	-358.6	-102.0	16.63

Figure 23. Welding execution screen.

On the top part of the screen, we can edit the name of the program. Next on the left part we have the column with the different welding instructions at our disposal. By dragging and dropping these instructions to the column on the right we create and add a new instruction to the program. This column on the right contains the sequence of all the instructions that compose the welding program. To parameterize a welding instruction, touch on it and the corresponding configuration screen for this instruction will appear. For welding instruction, the most common parameters are target poses and the welding profile that assigns and easy to remember name to a job number that is going to be sent to the welder during the weld instruction operation. On the right part of the screen, we have the list of poses that have been defined to be used by the welding instructions. The values of these points are taken with a simple teach procedure that is done also through the interface allowing to manually move the robot or enable the teaching mode in which the operator can hand guide the robot.

4.2.4. Process Perception Module

The purpose of this system is to detect two points in the welding joint, being initial and final point, respectively. All this development has been made by using OpenCV, Point Cloud Library (PCL) and it was integrated in ROS. Developed module has available the current pointcloud and capture is triggered by ROS topic. Output module of the system is the detected point represented in camera frame. These output points are used then to generate the trajectory, with an initial and final lineal approach to these target points.

Therefore, to orchestrate the application, five ROS modules are required: hand to eye calibration module, camera sensor driver, pointcloud processing module, robot driver for sending position commands and a program to automatically send the commands. Previous steps before being able to execute joint points localization is to perform camera eye in hand calibration and the robot tool calibration. In the Figure 24, required reference frames are depicted.

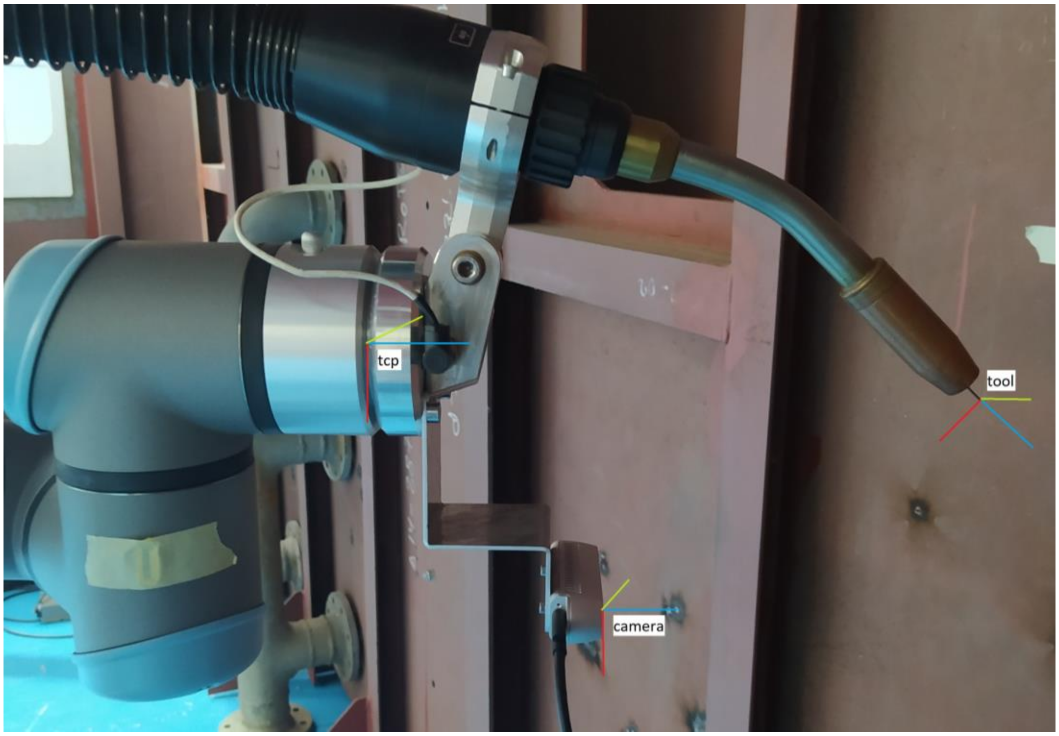


Figure 24. Camera and tool calibration frames for welding application.

Eye in hand calibration

This step is crucial for sending then position commands to the robots, as we will have all kinematic chain complete from robot base to camera frame. This module is also integrated in ROS, and we have as inputs a series of data composed by image and robot TCP position, and chessboard size pattern to successfully detect the corners using OpenCV functionalities. The result of this calibration will be the transform (position and orientation) between the camera frame to the robot TCP frame. Some of the input images for eye in hand calibration are represented in Figure 25, although a minimum of 10 images are required to calibrate this system. These images must contain the pattern seen from different points of view, always keeping the pattern static.

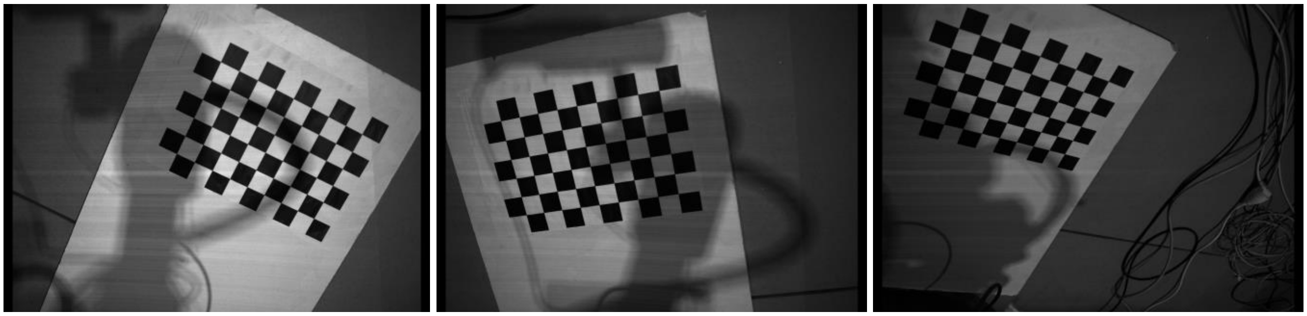


Figure 25. Eye in hand calibration images.

Robot tool calibration

For the UR10 robot, movement is orchestrated from ROS Master PC, we need to know the transform from the tool to the robot TCP. This information is required to transform camera information to the reference system of robot tool and send the command in TCP coordinates. To obtain this transform, we had used the procedure from UR10 which requires 4 points and 1 axis alignment to obtain position and orientation, respectively. For position, we have to reach a fixed point with different tool orientations. In the orientation computation, we must align tool Z axis with basis Z axis. By using robot teach pendant PolyScope procedure we will obtain tool calibration.

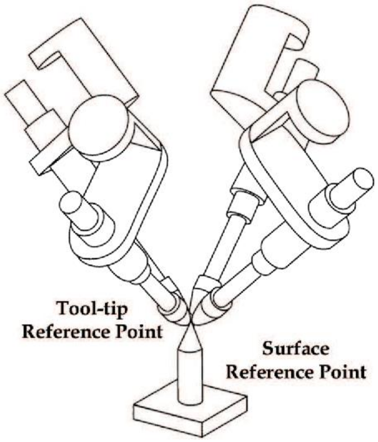


Figure 26. Robot TCP Calibration Procedure. Source: [8].

Data acquisition sequence

Two captures are needed to obtain initial and final welding points. This procedure has been automated in ROS for UR10 robot. In this case, it will have a start initial position, moving to the capture positions passing through start point in the middle. When capture positions are reached, this will trigger the capture by publishing in its respective ROS topic. For Elfin5 robot, its movement is done externally to ROS and trigger is manually done by topic publication.

Vision algorithm

Firstly, an aspect to take into account is that capture distance between the camera and the joint must be of 30 cm approximately (Figure 26Figure 27). A configuration with a reduced field of view is used to only perceive the two interest planes (Figure 28Figure 27, right).

Figure 27. Capture position and reduced field of view (only corner appears).

Once capture is asked, RANdom SAmple Consensus (RANSAC) algorithm [9] is used to detect the two planes whose intersection is our welding joint. This detection is shown in Figure 28.

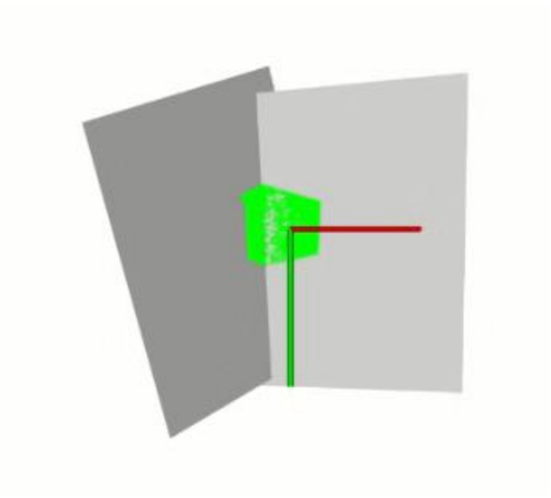


Figure 28. Detection of the two planes.

Next step is to compute line intersection by using a defined PCL function, which is represented in blue in Figure 29. Then, point is computed by the intersection between the detected line and Z axis of the camera. Orientation of this point is arbitrary set: X (red line, Figure 29**¡Error! No se encuentra el origen de la referencia.**) axis is in the direction of the camera, Z (blue line, Figure 29) axis in the direction of detected line and Y (green line, Figure 29) is computed with cross product. This output point is published to ROS as *geometry_msgs::Point* message.

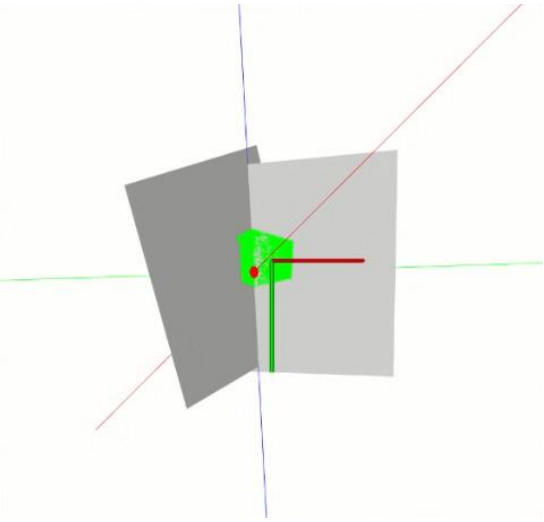


Figure 29. Detected welding point.

Welding execution and results

To perform the joint weld, we will take these two points following this sequence: beginning from the end point programmed in vision module, we will move to the first point with an offset. Then, with linear movement we will approach to the first one and move linearly to the second one. Later, robot will perform a linear offset to move away and go to rest position again. Furthermore, automatic start and end of the welding machine is done during the linear trajectory between detected points.

The complete process of transporting the robot, the welding machine with its accessories, vision process, human robot interaction module and welding execution is shown in this video: Welding Joint Detection - YouTube. The welding joint execution results at AIMEN facilities are presented in Figure 30.

Figure 30. Welded joint

5. POSITION/CUT OPENINGS

5.1. Description

The early stages of projects often overlook defining crucial openings in structural parts, resulting in a lack of information for cutting plates using machines. This issue is especially common with pipe and cable tray penetrations, where imprecise measuring tape methods are used to determine their position and shape. Consequently, the openings are cut from bulkheads or decks after incorporating them into the block or vessel, rather than in the workshop. On-site cutting of larger openings like doors often requires modifications to reinforcements, impacting the 3D model. This reliance on inaccurate paper drawings and measuring tapes can lead to errors in positioning, geometry, and reinforcement placement.

In this particular instance, the development involves the collaborative efforts of both AIMEN and INESC. AIMEN has contributed by developing the Process Perception Module, which serves the purpose of accurately localizing the UR10 robot within an indoor environment. Subsequently, the module facilitates the precise cutting of the designated area, as defined in the 3D model, utilizing a plasma cut machine. The control of this machine is accomplished through the utilization of input signals from the robot.

5.2. Integrated Technologies

5.2.1. Portable Robot

To be able to cope with plasma cutting operations inside a vessel block and deploy AIMEN's developed vision software, a collaborative UR10 robot was chosen. The robot is small enough to be handled and carried easily by operators, and it is able to manipulate loads around 10 kg attached to its tool flange. In this case, a plasma torch was selected to perform cutting tasks. Besides this, a RealSense camera was attached to perform the perception tasks. These devices were attached to the robot flange using a designed and manufactured fixture.

Additionally, to be sure the robot was accurate enough to perform the assigned tasks, it was important to have a system to fasten them firmly and securely in the surroundings where the processes were to take place inside the vessel. Besides this, this system to fasten them, should be as much lighter as possible and quickly to deploy. The solution was to use a magnetic base attached to the robots base, able to be fasten it to any of the metal plates are made of a vessel block. With this solution, the overall equipment weight, robot + magnetic base, was not heavier to 60kg, enough to be held and handled by a pair of operators.

Regarding this solution adopted, with the UR robot, a bunch of equipment was necessary to gather, and to integrate to perform the plasma cutting-process. Within Table 12 are summarized all the equipment used:

Table 12 Plasma cutting process equipment

Plasma cutting process equipment		
Equipment Name	Type of equipment	Weight
Robot	UR10	28.9 kg
Magnetic base	Magswitch cobot UR10	30 kg
Plasma power	Hypertherm Powermax 85	28 kg
Plasma torch	Hypertherm Duramax Hyamp 180	3.5 kg
Perception camera	RealSense D435	< 600 gr

And next picture (Figure 31) shows all these devices during the trials done inside the vessel block:

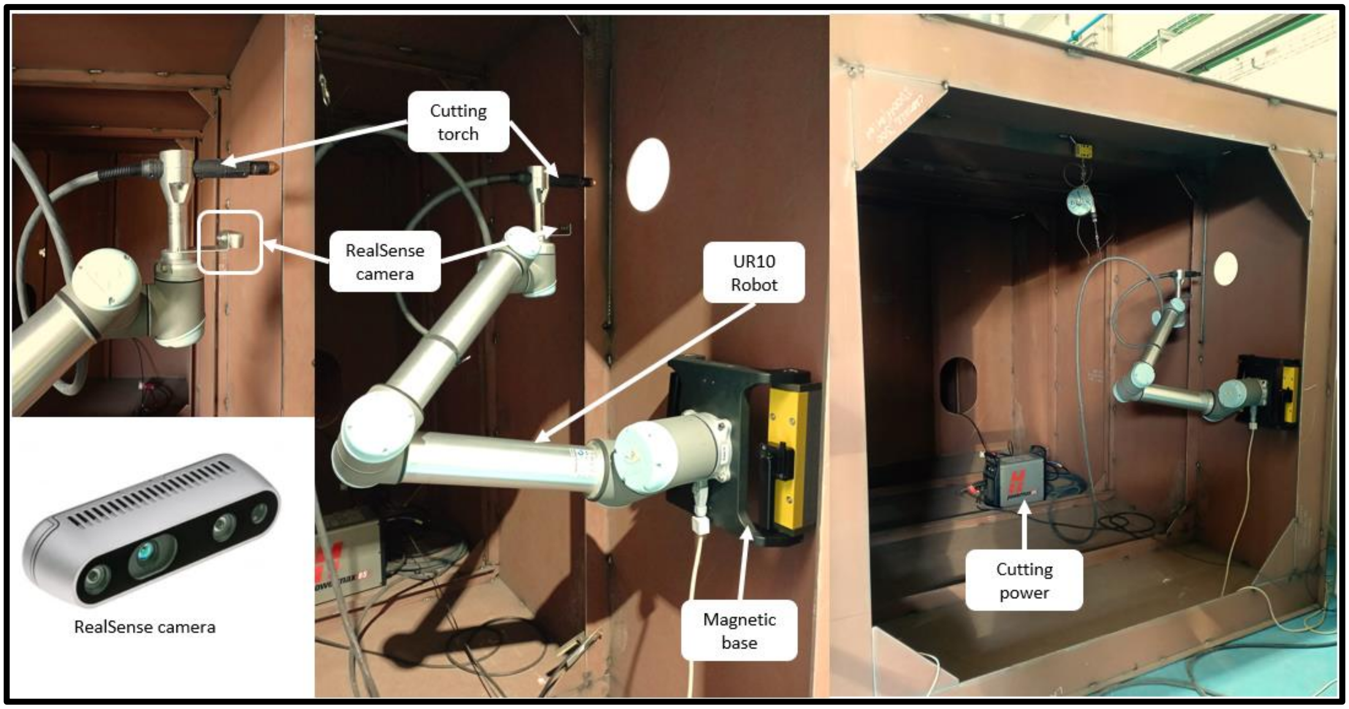


Figure 31. Devices required to deploy plasma cut application.

A PC master with ROS executing is the responsible for communicating with these devices, through different interfaces like USB3.0 to communicate with the camera, or TCP/IP to communicate with the robot. However, the communication with the plasma power should be done through digital inputs and outputs with the robot I/O board interface because the plasma power did not have any other communication bus (Figure 32).

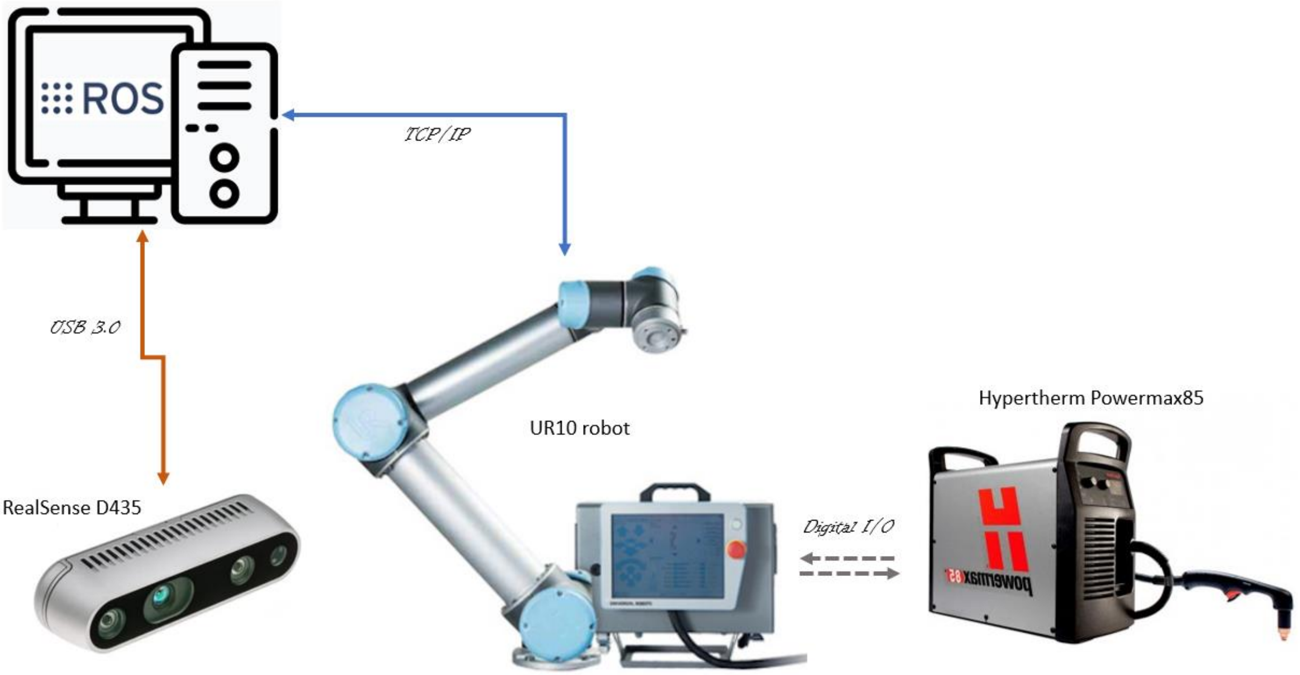


Figure 32. Communication diagram among application hardware devices.

5.2.2. Process Perception Module

Perception module developed by AIMEN in the Mari4_YARD project consists of performing 3D localization of the robot by the comparison between a 3D model of the scene and a real reconstruction of an area of the scene. To obtain the results, UR10 robot magnet-mounted and sensor Intel RealSense D435 camera are used. This development has been done with OpenCV Point Cloud Library (PCL) and it works in ROS environment.

To deploy and successfully locate robot in this confined space, next ROS modules are required: eye in hand calibration, camera driver, indoor localization module, robot driver and cut opening execution. In the same way it happened with welding joint detection, it is mandatory to compute eye in hand calibration and tool calibration. We will follow the same procedure already explained in (*Eye in Hand Calibration and Tool calibration*). Figure 33 shows the reference frames to obtain for current system mounting.

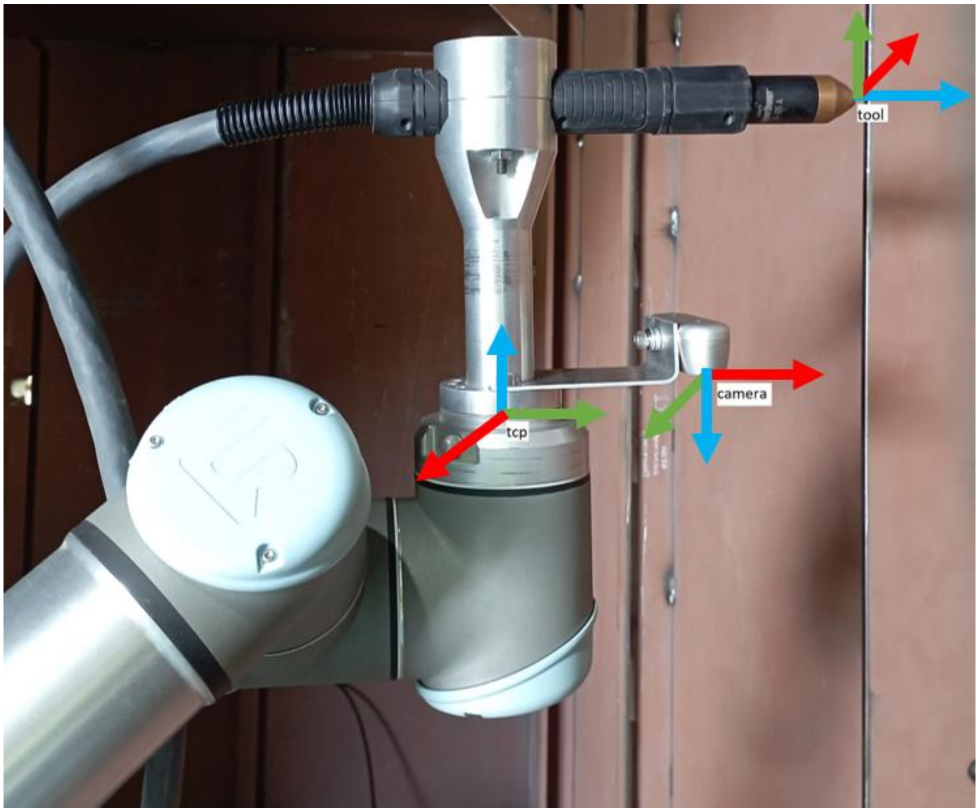


Figure 33. Camera and tool reference frames for cutting application.

CAD Preparation

First step is to obtain a pointcloud from the STEP model (Figure 34). This process has been made with CloudCompare. This tool provided us the methods to obtain a *pcd* file by sampling STEP meshes. Besides, in this cloud is represented the cut opening position. The generated file is shown in Figure 35, in which black circle is the target cut opening with a diameter of 20 cm.

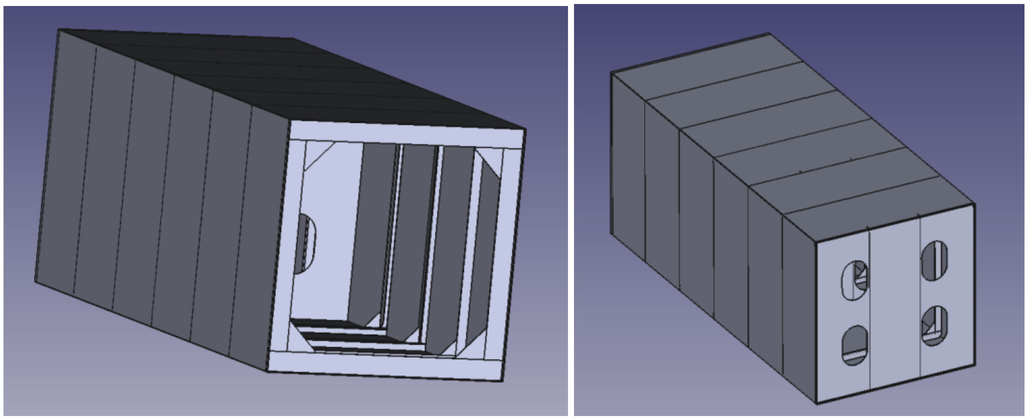


Figure 34. CAD / STEP model.

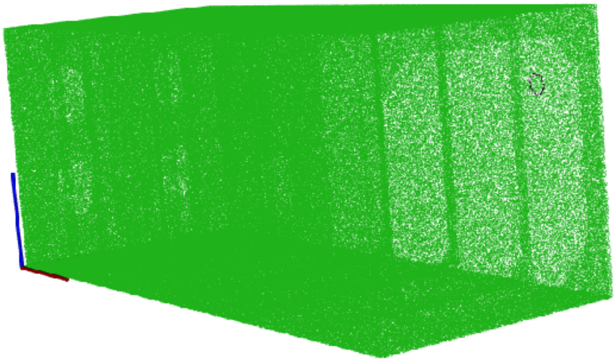


Figure 35. Sampled *pcd* file from STEP.

Algorithms

To succeed with the cut opening, we have first to locate the robot inside this closed environment. Therefore, a reconstruction process of an interest area is performed. In this case, we have chosen the holes that already exists in this design as it will result easier to match real and cad-generated pointclouds. In this case, we have defined two positions to combine, captures shown in Figure 36. This process is done by matrix transformations due to the fact that we know what movement the robot has done from ROS *tf* topic and URDF. Also do we know the relative position from the camera to the robot flange. Final reconstruction pointcloud is shown in Figure 37.

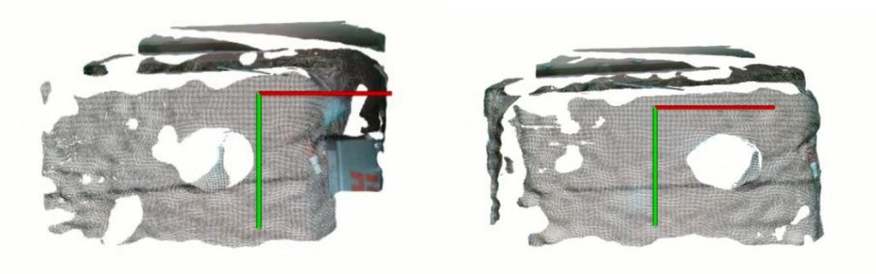


Figure 36. Reconstruction partial captures.

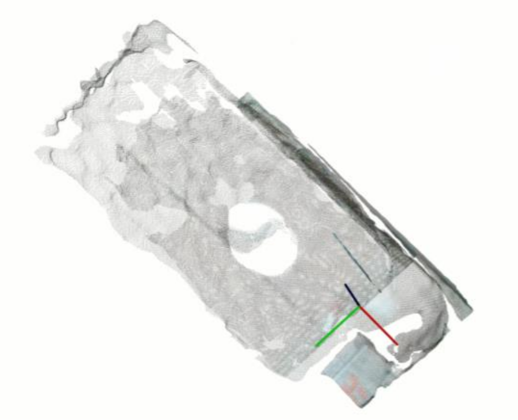


Figure 37. Reconstructed pointcloud.

Next step is to set an initial guess of where the camera is placed inside the area. This is done by interactive selection of this estimation pose on the visualizer (Figure 38). This step is necessary for using later for Iterative Closest Point (ICP) algorithm.

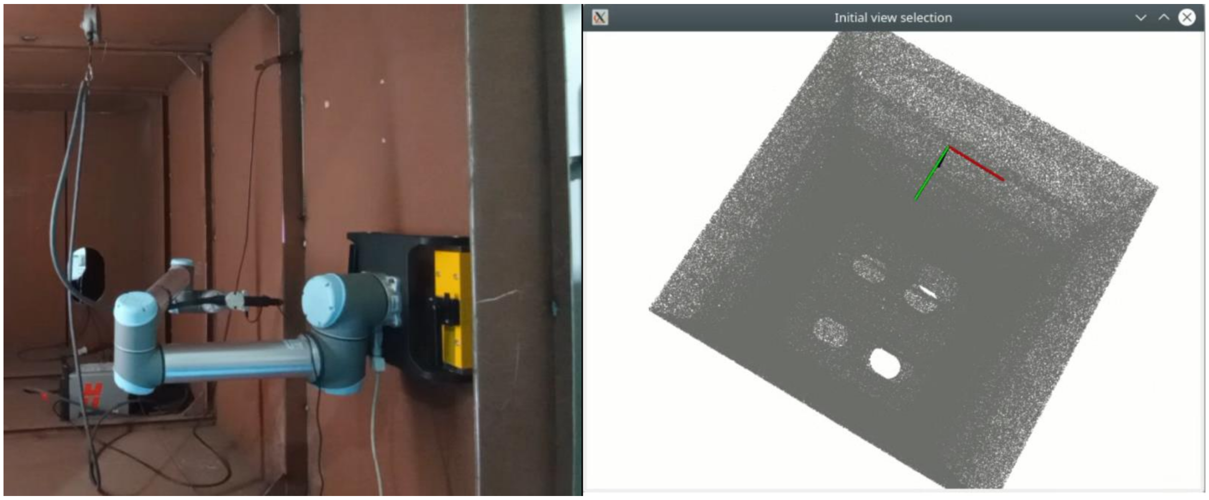


Figure 38. Real robot position (left) and initial guess of camera position estimation (right).

ICP needs to have two pointclouds, being these the CAD cloud and the obtained reconstruction pointcloud. With these input data it will perform a series of iterations minimizing the distance between the points. Hence, ICP algorithm requires these data to be as similar as possible in an initial state to refine this localization and getting the least distance between the points of both pointclouds. Nevertheless, when we have an area which is quite similar without special characteristics, the algorithm could converge in a wrong result. In Figure 39 initial state of ICP it is shown. This representation is done using the initial guess provided in the step before. Then, after computing different iterations to reduced distances, we matched the pointcloud with CAD model (Figure 40). This refinement process will modify the initial robot and camera position and it will publish it into ROS topic.

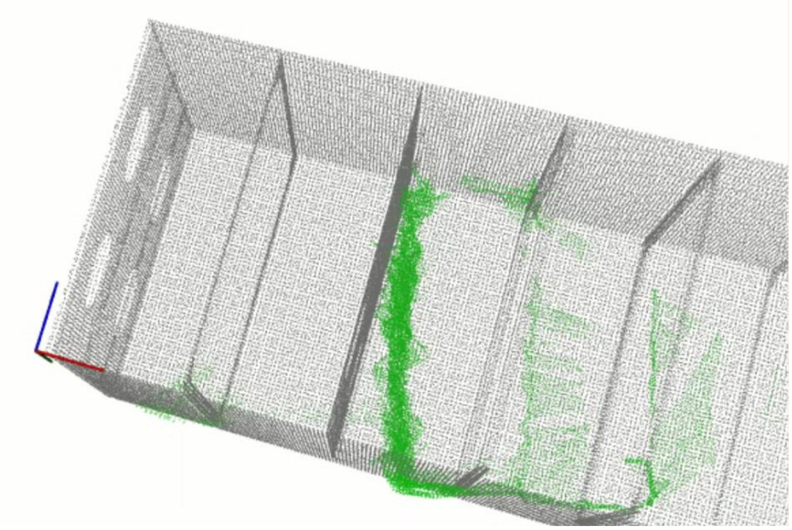


Figure 39. ICP initial state.

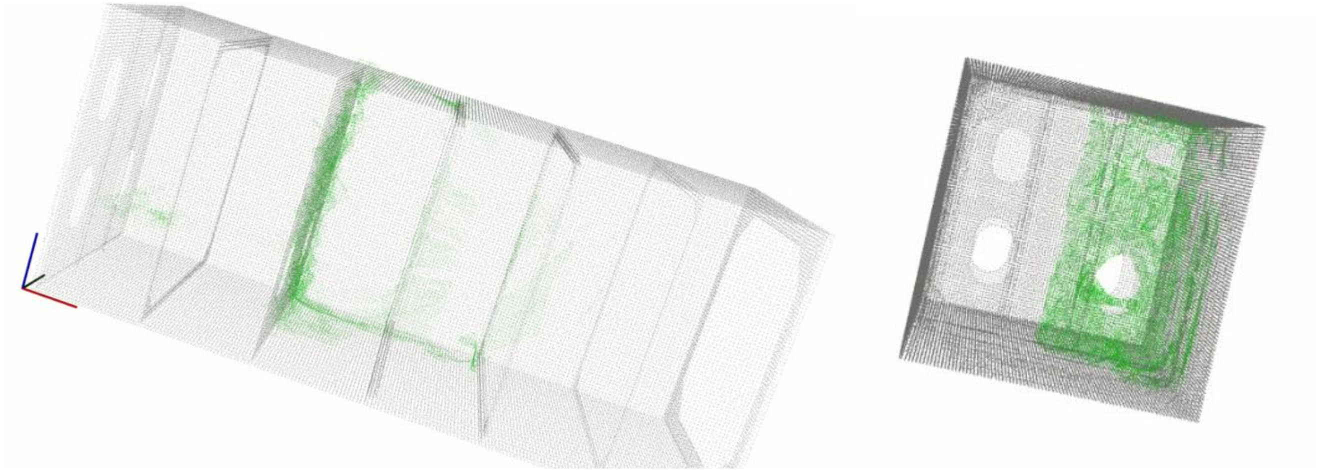


Figure 40. ICP final state.

Cut opening execution.

Once having located the robot inside the real scene, we can perform the cut with the plasma machine. The trigger of the welding machine is integrated in the robot signals. To cut the surface, we had approximated the robot to the initial point and then go through this wall once enabled the plasma cut machine and then, start the circular trajectory. When the robot arrives to the last point, plasma cut machine is disabled.

Results

Apart from the process images in the "*Algorithms*" section, in Figure 41 it there are two pictures of plasma cut execution. A video of the complete process (reconstruction, localization and plasma cut) is shown at: Robot localization and Plasma cut - YouTube

Figure 41. Plasma cut process.

6. TRANSPORT OF PARTS

6.1. Description

At todays shipyards, the transportation of raw materials and/or manufactured parts between stores and workshops, and from workshops to subassembly areas, is still heavily reliant on human operators. This transportation is typically performed by hand or by using self-propelled, pulled, or pushed platforms. According to "The Digital Shipyard Report – Opportunities and Challenges" from 2021 [1], the logistical complexity of the shipyard extends across warehouses, work sites, various parts, and components. During the shipbuilding process a wide range of components including structural steel, pipes, cables, valves, and outfitting are supplied, handled, and transported. These parts are normally stored in warehouses or pallets, and are placed on shelves, big containers and/or boxes. Please, refer to Figure 42 for some examples.

Figure 42. Figure illustrating the current situation provided by BIS MARY4_YARD project Partner.

These logistic tasks are dull, dirty, and dangerous for the human operator, owing to the specific characteristics of the parts to be transported, such as geometry and weight, as well as due to the unstructured environment that is common to see in shipyards. Moreover, and due to the ageing of the European population it is important to empower the current human workforce to other tasks that effectively contribute not only to the added value of the product produced, but also to the wellbeing of the human operator. So, there is a high interest, not only in the shipbuilding sector, but also from many other sectors to automate as much as possible these intra-logistic operations.

To answer these challenges the Mari4_Yard proposed the development of a mobile manipulator solution, reported initially on D2.1, that allows the picking of individual parts from containers. The major advantage of Mobile manipulators when compared with traditional AGV/AMR solutions, are that they combine not only the capacity to transport the load on top of the mobile platform but add the manipulation capabilities well recognized in industrial robotic arms. The deliverable is accompanied also with a video, demonstrating the solution, and may be found in this link.

6.2. Integrated Technologies

6.2.1. Mobile Robot – General Architecture

The mobile robot platform (MRO) used to build the described solution, is the one previously presented in section 2.7. In this mobile manipulator a complete software stack was installed, that enables its operation in different environments and for different applications scenarios. Please refer to Figure 43.

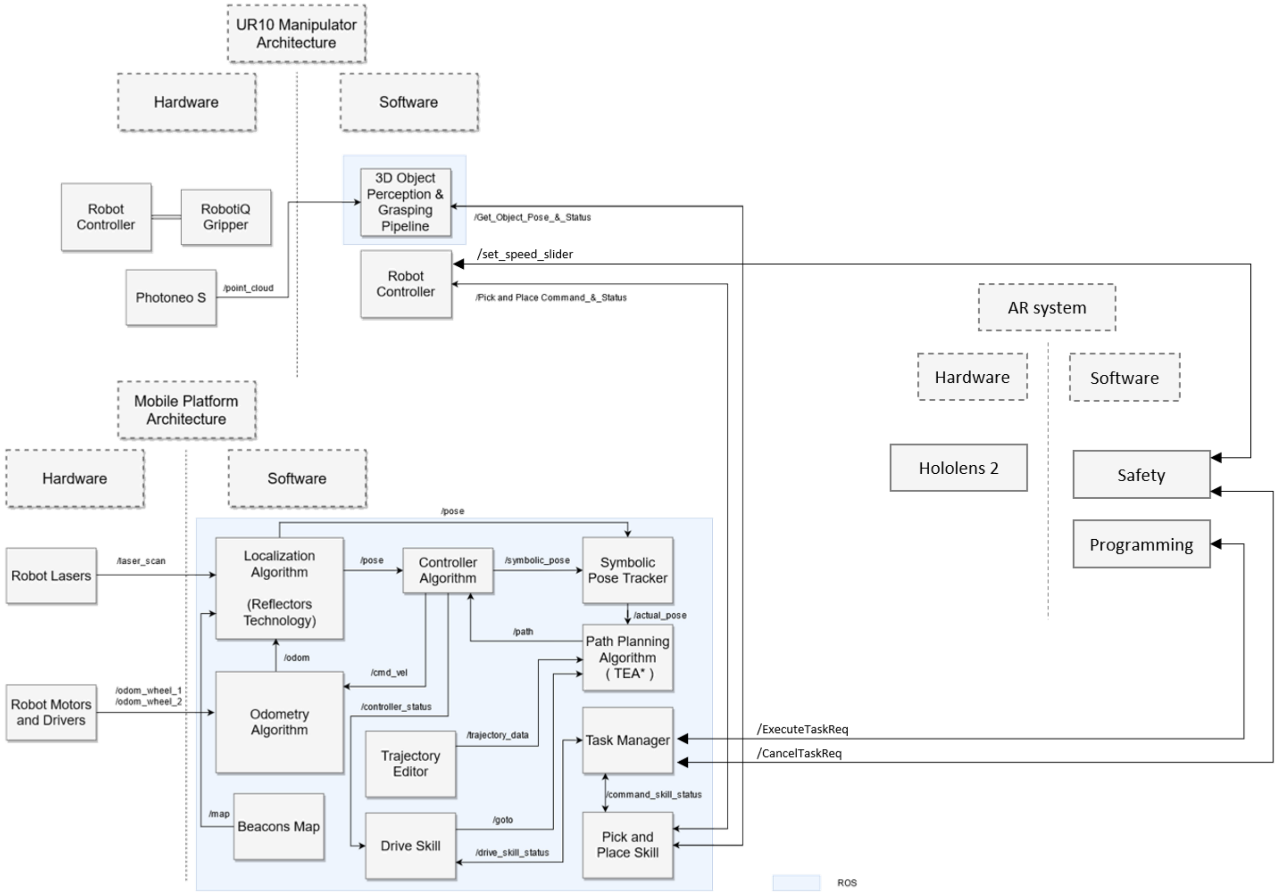


Figure 43. **Mobile manipulator software architecture**

More in detail, and following the robotic skills concept presented earlier in section 2.4, the software stack is endowed with:

1. A multi-robot coordination and navigation pipeline, the PPM MRLS sub-module presented in section 2.1, encapsulated under the Drive Skill, that ensures the proper localization and navigation of the mobile robot in the environment, and its coordination with other autonomous robots. For localization, the robotic system combines odometry data with data from two Sick Lidar lasers, which allow the extraction of the natural contour of the environment or detect artificial markers. Furthermore, these same lasers are used to ensure the safety of operation by detecting the presence of obstacles and triggering a reduction in speed or even a stop of the robotic platform.
2. An object 3D perception and grasping tool, combining the PPM 3OPS, OGPS and OSDL sub-modules presented in section 2.1, contained under the Pick and Place Skill, that allows the robotic arm to localize and manipulate the different types of objects. It resorts to the Photoneo 3D sensor and the Robotiq gripper, mounted on the robotic arm Tool Center Point, for acquiring the scene information and pick the detected objects.
3. regarding the interaction with the human operator, the robot system also allows the usage of an augmented reality system (AR) based on HoloLens, the HRIM ARIS sub-module described in section 2.4, to visualize robot status and programming production tasks using skills through an intuitive interface. During these interactions, the system allows for the creation of a 3D safety fence to provide the operator additional safety.
4. The Production Manager and Task Manager, presented earlier in section 2.3, acting s as the task planner and local orchestration modules, being responsible, respectively, for the assignment of the logistic tasks to the robotic system, supervision and to perform the supervision of the execution of different modular robotic skills, such as Drive Skill and Pick and Place Skill. Task Manager is also responsible for capturing operational data from the robotic system, to be integrated and processed in the upper layers of the manufacturing stack.

In the next subsections, each of the main modules of the developed solution are presented in more detail.

6.2.2. Workspace Monitoring Module

The Workspace Monitoring System contributes to increasing the workers' awareness and safety. The MRO has two complementary solutions to implement it.

The first safety solution installed is built around two sick Lidar safety sensors placed in opposite corners of the robotic platform. Please refer to Figure 44. With this configuration, it is possible to monitor the entire workspace surrounding the robotic platform and allow for a reduction in speed or even a complete stop of the robotic system based on the proximity of the obstacle.

Figure 44. Robotic platform safety lasers localization

The second safety system is based on the AR solution. The Microsoft Hololens 2 allow for the definition and monitoring of virtual safety zones (3D volumes), visible in Figure 45, to reduce the risk of collision with people in the path of the robots. Holograms of the safety zones are projected on the shop floor, making the operator aware of the robot's workspace and potential collision risks. It also allows reliable safeguarding robots by tracking the worker's hands and head. Whenever safety zones are breached, workers receive immediate visual and audio warnings. In addition, the system triggers the robot to change its speed (allowing the object to move away from the robot's path) or even to come to a complete stop, thus preventing imminent collision.

Figure 45. AR virtual safety fence

6.2.3. Human Robot Interaction Module

The robotic system is programmed using a task-based paradigm, in which simpler units of behaviour known as robotic skills are combined to form more complex units of behaviour known as robotic tasks. Robotic skills are made up of modular software units that execute single and generic actions (e.g., open a gripper, close a gripper, or move a mobile manipulator to a specific point in space). The following Skills were programmed using the Skill Generator tool for the current use case: WaitSkill, AbortSkill, MoveArmSkill, PhotoneoSkill, Gripper Skill, ObjectRecognitionSkill, GraspEstimationSkill, and DriveSkill.

Resorting to the AR system, it was possible to improve human-robot interaction by implementing a codeless robotic task programming approach. This approach was designed to allow non-robotics experts to program industrial and collaborative robots more easily. It consists of manually moving the robot arm through a set of waypoints. The difference to the traditional hand guiding approaches is that, in the AR system, the operator manipulates the hologram of the robot arm instead of the physical robot. An intuitive holographic interface, visible in Figure 46, allows the user to assemble and parameterize a sequence of robotic skills, and therefore, create a full robot task. Once finished, it is stored in an SCXML file and it is possible to send the production task to the robot Task Manager.

Figure 46. SKILL based robot programming using the AR system.

Another option for programming robotic tasks is to use the Task Creator, a Production Manager tool, which is discussed in the following section; however, in this case, some of the skill parameters (e.g. robot position, speed, other) must be manually entered into a configuration file.

6.2.4. Control orchestration and planning Module

For control orchestration and planning, the robotic solution resorts to the Production Manager and Task Manager.

The Production Manager (PM) is a web-based application that combines various applications for intuitive, data visualization and operational oversight, also allowing robot task programming.

For robot task programming, the PM has an application called Task Creator, which allows the creation and editing of robotic Tasks via a web-based graphical tool. Please refer to Figure 47. It enables users to configure individual robotic Skills and link them together to create the desired execution flow. The tool creates valid SCXML files that can be used by robotic agents later on.

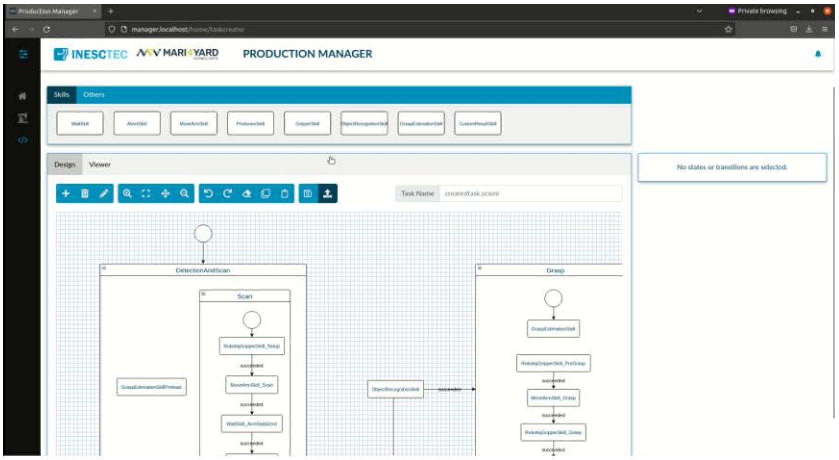


Figure 47. Robotic task creator

The Production Manager can also communicate with multiple robotic agents at the same time and capture log messages printed by ROS nodes running inside the robotic agents. These log messages can be used to monitor, test, and debug robotic applications. The tool saves all captured log messages for future reference. When a robot is successfully added, the capture of log messages begins automatically (assuming it is turned on and operating).

Users can use the Production Manager to assign and request the execution of robotic Tasks, as well as inspect their execution in real time via a graphical user interface. It is possible to obtain up-to-date information on all robotic Skills within the overall Task. Please refer to Figure 48.

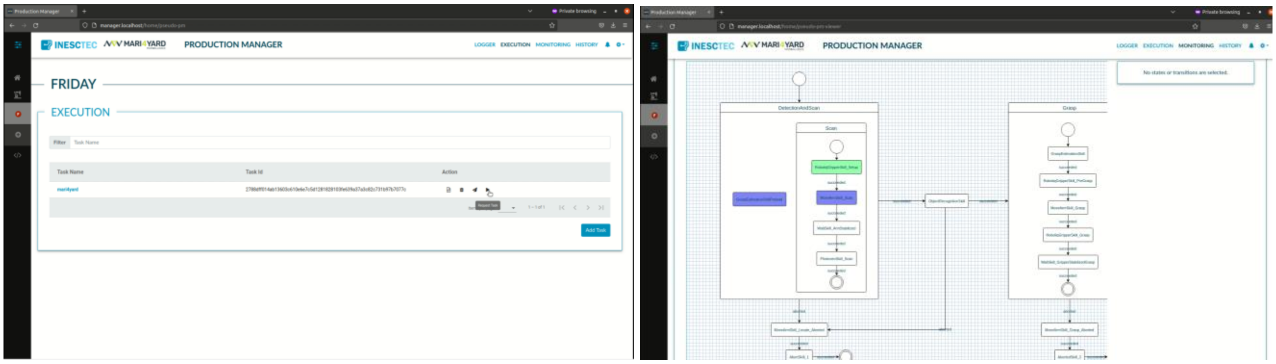


Figure 48. Robot task assignment and monitoring of its execution

The Production Manager stores all information concerning the execution of robotic Tasks. This information can be later consulted if so required.

Concerning the Task Manager (TM), which runs at the robotic level, it receives as input from the Production Manager the task assigned to the robotic system in the form of a SCXML file, interprets it, and ensures the execution of each Robot Skill by the robotic system, providing feedback to the Production Manager on their execution, as seen in Figure 48.

6.2.5. Process Perception Module

The PPM solution implemented in MRO platform is composed of four different submodules: segmentation (OSDL), perception (3OPS), grasping (OGPS) and mobile robot navigation and localization (MRLS). The three first submodules together enable the robotic arm to pick a wide set of parts (with different geometry and weight) stored inside a box or a bin. For a detailed explanation of this pipeline please refer to [2][3], which was published in the context of the Mari4_Yard project. The MRLS sub-module allows the mobile platform to navigate in the environment.

A brief presentation of each sub-module and individual results are here presented.

Beginning with the robotic arm, and with segmentation submodule, it oversees finding a solution to the segmentation issue that accurately perceives and segments objects in cluttered picking scenarios and that is consistent with observed industry patterns. This procedure is based on deep learning and point cloud-segmentation methods enabling the detection and segmentation of the objects. Then, based on a set of Heuristics, such as the highest object, or the object with the most surface area the object to be grasped is selected. Figure 49 present some results of this segmentation process.

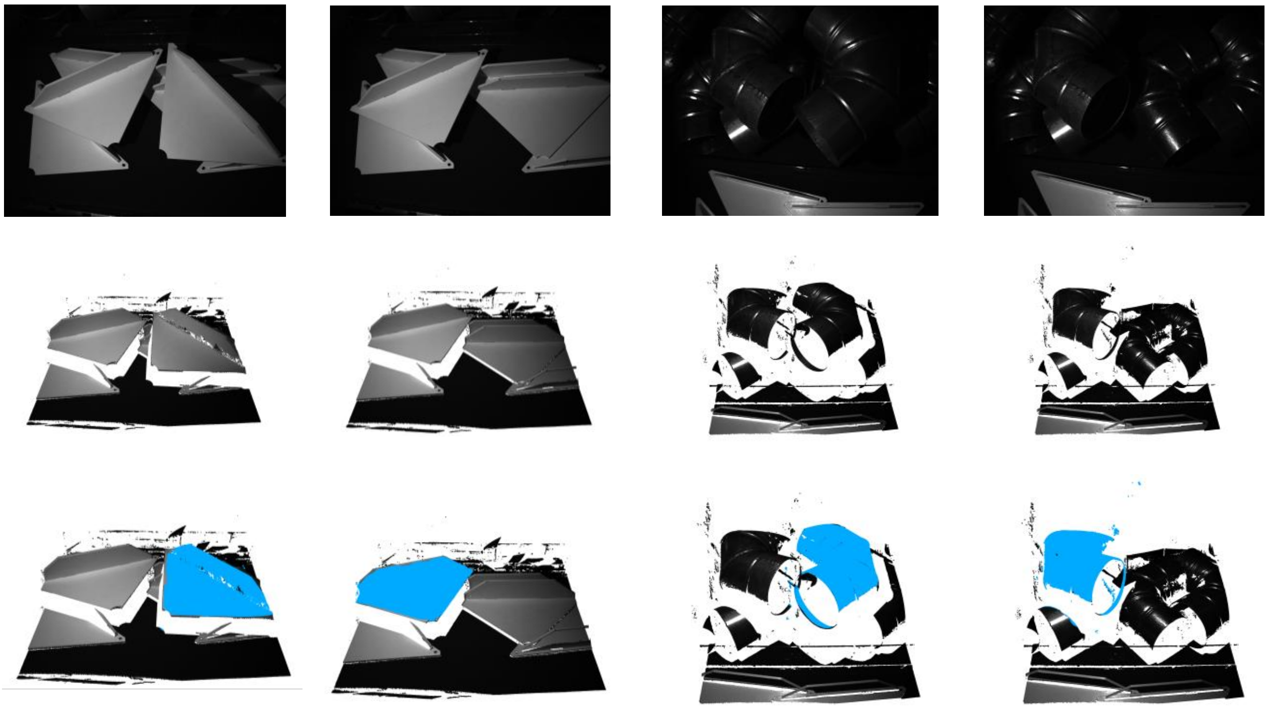


Figure 49. Several illustrations of the segmentation process's output. In the first row it is presented the sensor's 2D raw image. In the second row it is presented the sensor's raw 3D point cloud. In the third row it is presented the result of the process of segmentation and selection of the object to be grasped, represented in blue colour.

From the outputs of the segmentation submodule, it is then required to estimate the position of the object to grasp based on a reference point cloud. For this purpose, the perception submodule is used. This submodule performs the registration of the point cloud returned by the segmentation model with the CAD reference model (or a point cloud sample) of the object, returning its 6-DoF position and orientation. Figure 50 presents the results of this perception process.

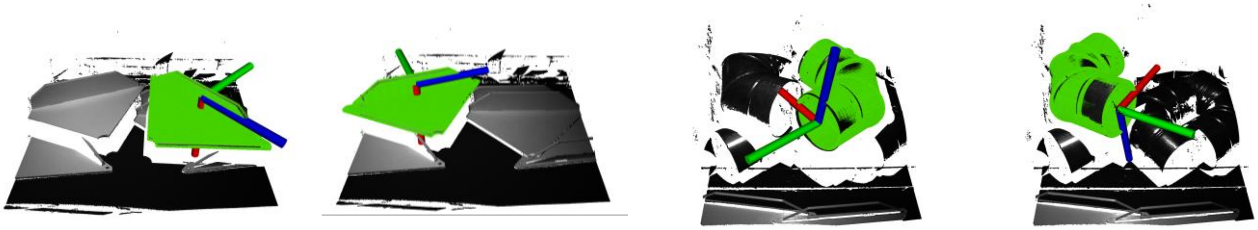


Figure 50. Several illustrations of the perception process output, presented in green colour.

Finally based on the perception module outcome, the grasping selection submodule is applied. Processing for this submodule is split into two phases. The first stage, which occurs offline, is in charge of automatically creating a set of grasping configurations (more than 100 configurations) oriented to the object 3D CAD model, and the second stage decides which configuration is appropriate at run-time. Several metrics and heuristics are used to evaluate the estimation of the chosen grasp candidate, including: the effort needed to move the mobile manipulator to the estimated point, calculated by the roll, pitch, yaw, and Euclidean distance; the collision with the workspace or another object, excluding grasp candidates with collisions; and, finally, the joint space, excluding candidates outside this space because the manipulator is not capable of moving outside this space. In other words, the grasp candidate with the lowest effort trajectory, inside the joint space, without any collisions, will be chosen as the best grasp pose. In Figure 51 presents the results of the grasp selection process.

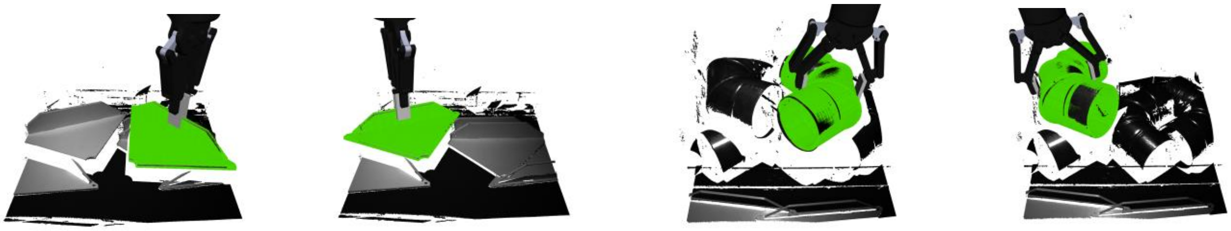


Figure 51. Several illustrations of the grasping selection process output, presented in green colour.

Finally, we have the MRL sub-module, that enables the mobile manipulator to move around the environment, pick up parts, and deliver them to various manufacturing or warehouse locations. This submodule resorts to the information retrieved by the sick safety laser to localize the mobile manipulator in the environment. It can use natural contours or artificial markers, independently or simultaneously. Please refer to Figure 52. In this same Figure it is also presented the robot navigation graph, which basically defines the paths (graph vertices) that the robot must follow from one position to another (graph vertices).

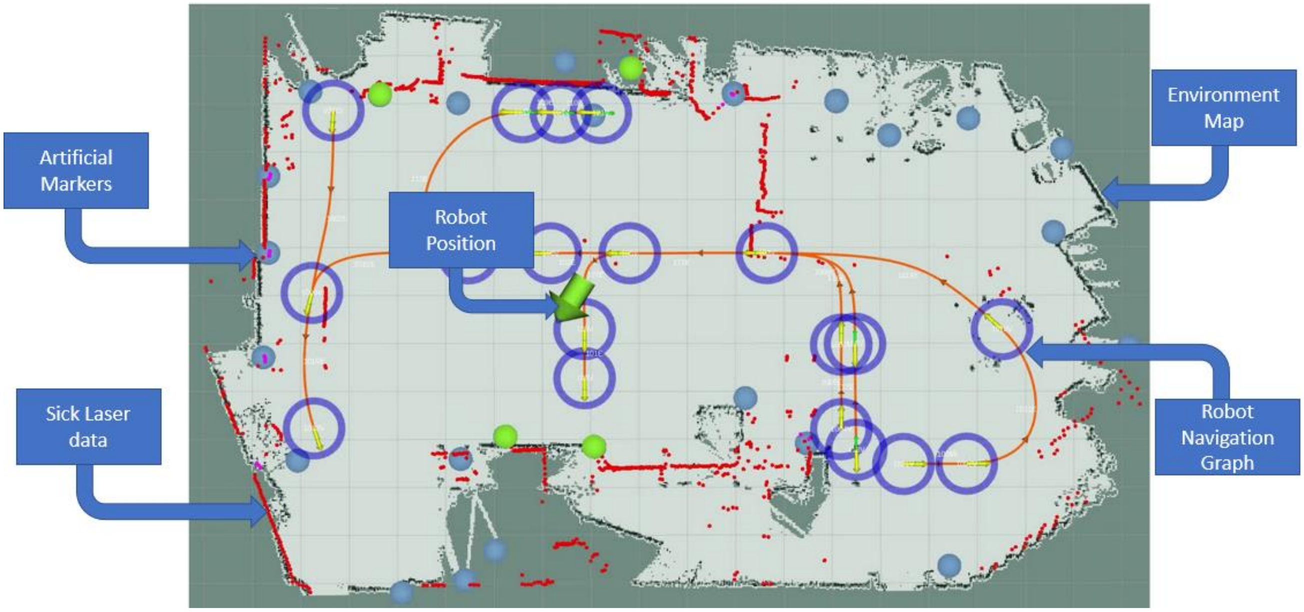


Figure 52. Mobile Platform navigation map.

7. CONCLUSIONS

In conclusion, the complexity and diversity of tasks performed in the shipbuilding sector necessitates an advanced and integrated approach to automation. In the context of Mari4_YARD project, D2.3 reports various developed modules that offer significant potential for enhancing efficiency and safety in shipbuilding processes. These technologies have been demonstrated to be crucial across various applications in the sector, such as parts assembly using welding process, welding with collaborative robots, precise positioning and cutting of openings, and efficient transport of parts. The demonstrators of D2.3 showcase the synergistic functioning of these modules, each playing a vital role in the intricate and dynamic environment of shipbuilding. Ultimately, these integrated technologies are shown to bring about a transformative effect, redefining the landscape of shipbuilding towards greater productivity, precision, and safety.

REFERENCES

[1] John Spoehr, Ryan Jang, Kosta Manning, Arvind Rajagopalan, Cecilia Moretti, Ann-Louise Hordacre, Sara Howard, Peter Yaron and Lance Worrall The Digital Shipyard, Opportunities and Challenges, March 2021, Flinders University - Australian Industrial Transformation Institute

[2] Cordeiro, A.; Souza, J.P.; Costa, C.M.; Filipe, V.; Rocha, L.F.; Silva, M.F. Bin Picking for Ship-Building Logistics Using Perception and Grasping Systems. *Robotics* **2023**, *12*, 15. https://doi.org/10.3390/robotics12010015

[3] Cordeiro, A., Rocha, L.F., Costa, C., Silva, M.F. (2023). Object Segmentation for Bin Picking Using Deep Learning. In: Tardioli, D., Matellán, V., Heredia, G., Silva, M.F., Marques, L. (eds) ROBOT2022: Fifth Iberian Robotics Conference. ROBOT 2022. Lecture Notes in Networks and Systems, vol 590. Springer, Cham. https://doi.org/10.1007/978-3-031-21062-4_5

[4] Stanford Artificial Intelligence Laboratory et al. (2018). Robotic Operating System. Retrieved from https://www.ros.org

[5] Coleman, D., Sucan, I., Chitta, S., & Correll, N. (2014). Reducing the barrier to entry of complex robotic software: a moveit! case study. arXiv preprint arXiv:1404.3785.

[6] Kam, H. R., Lee, S. H., Park, T., & Kim, C. H. (2015). Rviz: a toolkit for real domain data visualization. *Telecommunication Systems*, *60*, 337-345.

[7] B, C. "'Ros Drivers Suite' for Comau Robots." Sherlock Project, 19 May 2021, https://www.sherlock-test.eu/ros-drivers-suite-for-comau-robots/.

[8] Frank Shaopeng Cheng (2008). Calibration of Robot Reference Frames for Enhanced Robot Positioning Accuracy, Robot Manipulators, Marco Ceccarelli (Ed.), ISBN: 978-953-7619-06-0, InTech, Available from: http://www.intechopen.com/books/robot_manipulators/calibration_of_robot_reference_frames_for_enhanced_r obot_positioning_accuracy

[9] Cantzler, H. (1981). Random sample consensus (ransac). Institute for Perception, Action and Behaviour, Division of Informatics, University of Edinburgh, 3.