Mobile dual arm robotic workers with embedded cognition for hybrid and dynamically reconfigurable manufacturing systems

Grant Agreement No	**:**	**723616**
Project Acronym	**:**	**THOMAS**
Project Start Date	**:**	**1st October, 2016**
Consortium	**:**	**UNIVERSITY OF PATRAS (LMS)** **PEUGEOT CITROEN AUTOMOBILES S.A. (PSA)** **SICK AG (SICK)** **FUNDATION TECHNALIA RESEARCH & INNOVATION (TECNALIA)** **ROBOCEPTION GMBH (ROBOCEPTION)** **DGH ROBOTICA, AUTOMATIZACION Y MANTENIMIENTO INDUSTRIAL SA (DGH)** **AERNNOVA AEROSPACE S.A.U. (AERNNOVA)** **INTRASOFT INTERNATIONAL SA (INTRASOFT)**

Title	**:**	Environment and process reasoning modules – Initial prototype
Reference	**:**	D3.3
Availability	**:**	Public
Date	**:**	31/03/2018
Author/s	**:**	ROBOCEPTION, TECNALIA, LMS, INTRASOFT
Circulation	**:**	EU, consortium

Summary:

This document provides an overview of the perception skills used for environment perception and process reasoning. The document contains descriptions of software pipelines for detection of objects and relevant process areas, and software interfaces that connect the developed modules to the overall system architecture of the Mobile Robot Platform (MRP) developed within THOMAS. Initial implementations on the use cases of THOMAS are described, coping with challenges for the selected processes in AERNNOVA and PSA.

Table of Contents

LIST OF FIGURES

1. EXECUTIVE SUMMARY

The main purpose of this document is to provide an overview of the prototypical implementations of perception skills used for environment perception and process reasoning. An initial description of the perception modules was provided in a previous deliverable, D3.1, “Environment and process perception modules – Design”. This deliverable updates the description of the software pipelines for detection of objects and relevant process areas, and describes software interfaces that connect the developed modules to the overall system architecture of the Mobile Robot Platform (MRP) developed within THOMAS. The deliverable includes:

- Detection of objects based on CAD object models
- Tag detection
- Initial evaluation of expectable accuracy and precision for the corresponding modules
- Integration of perception modules with the skill engine
- Integration of perception modules with the overall robot control

The software components were implemented and tested on different testbeds representing the two use cases of AERNNOVA and PSA. The description of the implementations is provided, and three video attachments are included to show the initial results of the implementation.

2. PERCEPTION PIPELINE

This chapter provides an overview of the current developments for environment and process perception and reasoning. The design of the modules was initially presented in D3.1, "Environment and process perception modules – Design"; this chapter presents the latest status of the implemented modules.

2.1. Perception modules

Two main perception modules have been developed within WP3 for process and environment perception, namely, a CAD based object detection module, and a module for reading Apriltags, used in THOMAS for detecting regions of interest in the execution of the use cases.

2.1.1. Object detection

The final pipeline for object detection based on CAD models of the object is illustrated in Figure 1, using the detection of templates for the drilling operation in the AERNNOVA use case as an example. The detection pipeline contains the following steps:

- Acquisition of images and pointcloud data with the *rc_visard* sensor.
- Initial segmentation. When information from the process is available, such as in the drilling operation, an adapted pipeline can reduce the complexity of further processing steps. In this case, the skin of the wing where the drilling operation is performed is a large flat surface, which can be easily segmented from the pointcloud. The largest dominant plane is selected and the corresponding points are removed from further processing. The cluster of points representing the template is then selected, and an initial pose estimation is performed through the computation of the COM and axis of inertia of the cluster.
- Final alignment. For this step, the information coming from the image stream of *rc_visard* cameras is used. The edges are extracted from the image, and also the edges from the initial pose are rendered, thus leading to two silhouettes of the template over-imposed on the image, as shown in Figure 1. The edges are then aligned to improve the detection accuracy

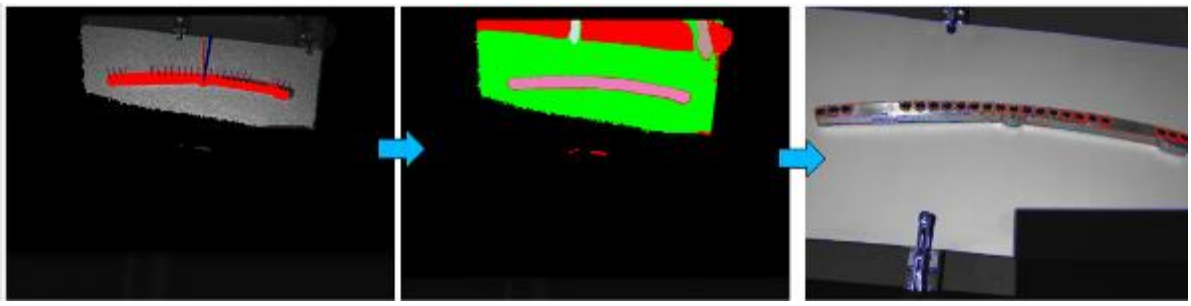


Figure 1. Pose refinement via edge alignment. From left to right: 3D data, segmentation and final alignment

An evaluation of accuracy was performed for the initial template detection, to provide an indication of the expectable accuracy for the real robotic application. Results indicate that the average accuracy is below 1 mm at a distance of 1 meter between camera and template.

2.1.2. Regions of interest – April tag detection

The detection of areas of interest is critical for the use case in PSA, where different locations are used for picking and placing components at several locations. To provide an initial guess of expected initial or goal pose for the desired objects, an Apriltag localization is developed. The robotic implementation will then look for the Apriltag, which provides a rough guess for the location of the objects, and then a CAD based detection is performed to obtain the required object pose for picking, or goal pose for a placing operation. Figure 2 illustrates this process.

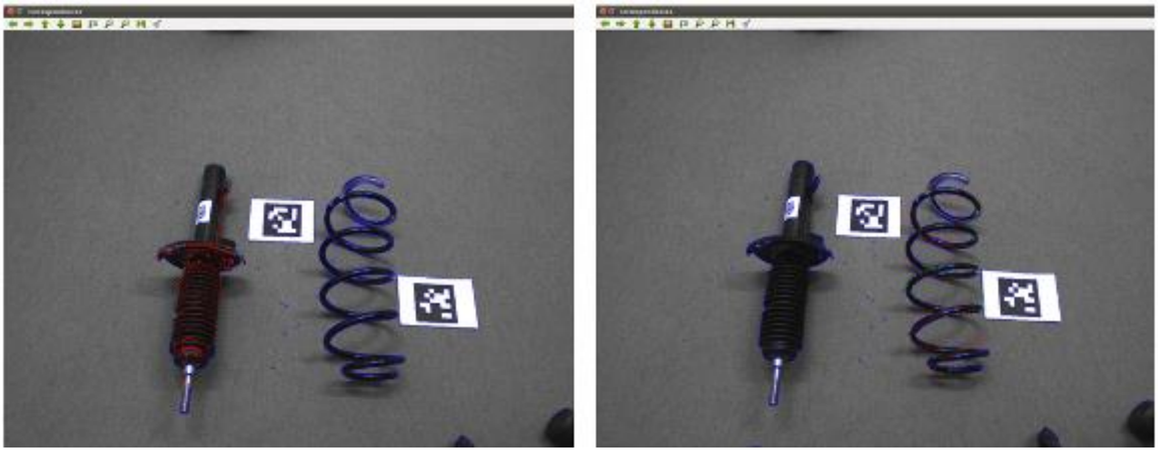


Figure 2: Localization of a damper (PSA use case) using Apriltags

The tag detection module developed at Roboception allows the detection of 2D bar codes and tags using the *rc_visard*. The component computes the position and orientation of each tag in the 3D camera coordinate system. Multiple tags can be located at the same time on the image. The tag detection module follows these general steps:

- Tag reading in the stereo image pair. The quality of this process can be controlled through a quality parameter that controls the downscaling of the images, which increases the speed but reduces the accuracy of the detection process.
- Tag pose estimation. For each tag detected on the image, the pose of the tag in the camera coordinate frame is computed. The coordinate frame for the tag is aligned as shown in Figure 3, with the Z axis pointing into the tag. The size of the tag is also estimated at runtime. For best pose estimation results, the tag must be accurately printed and attached to a rigid and planar surface.

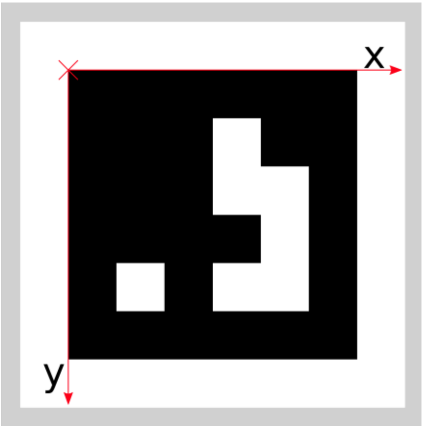


Figure 3: Apriltag reference frame

Systematic tests were carried out to determine the expected precision for the Apriltag detection, which mainly depends on the distance to the tag (and the camera resolution). The results indicate that at a distance of 20 cm, the positional error is below 0.5mm, and the angular error below 1 degree.

2.2. Interfaces and integration with other modules

2.2.1. Integration with skill engine

The integration of the perception modules with the skill engine is done using ROS services. To illustrate this integration, the drilling operation of the AERNNOVA use case is taken as example. Section 3 explains the details of the testbed setup for the AERNNOVA drilling operation. Two sensors are used for template detection and drilling. One is used for template detection from a fairly long distance (*rc_visard* 160) and one for refinement of the template hole positions from a closer distance (*rc_visard* 65). For a clearer understanding, they will be referred to as template detection sensor and hole detection sensor, respectively. The template detection sensor has been placed on the left arm of the robot and the hole detection sensor has been installed on the right arm (See Figure 14)

The ROS services that make possible the integration with the perception system are:

- *detect*: detect is called to detect the template position. It is necessary that the camera can perceive the whole template.
- *refine*: it is called to get a more accurate position of the template. It gets as input an estimated position of the template (coming from the previous module). This service does not require the camera to see the whole template.

Figure 4 shows the sequence of skills that make the robot able to navigate to the drilling area. It first puts the arms in a compact safe position, then goes to the drilling place using the robot´s navigation skill. Next step is to place the robot accurately in the exact position to start the drilling process. For that we use tag detection capability to place the robot at an exact position with respect to the tag. Finally, the robot extends its arms to start the drilling operation.

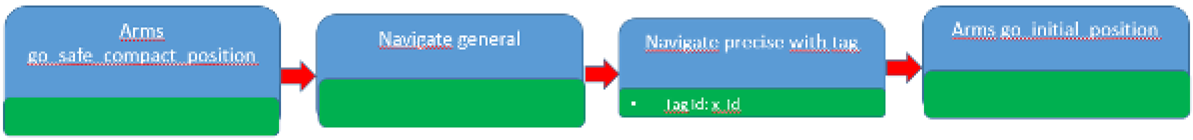


Figure 4: Skill sequence to navigate to drilling station

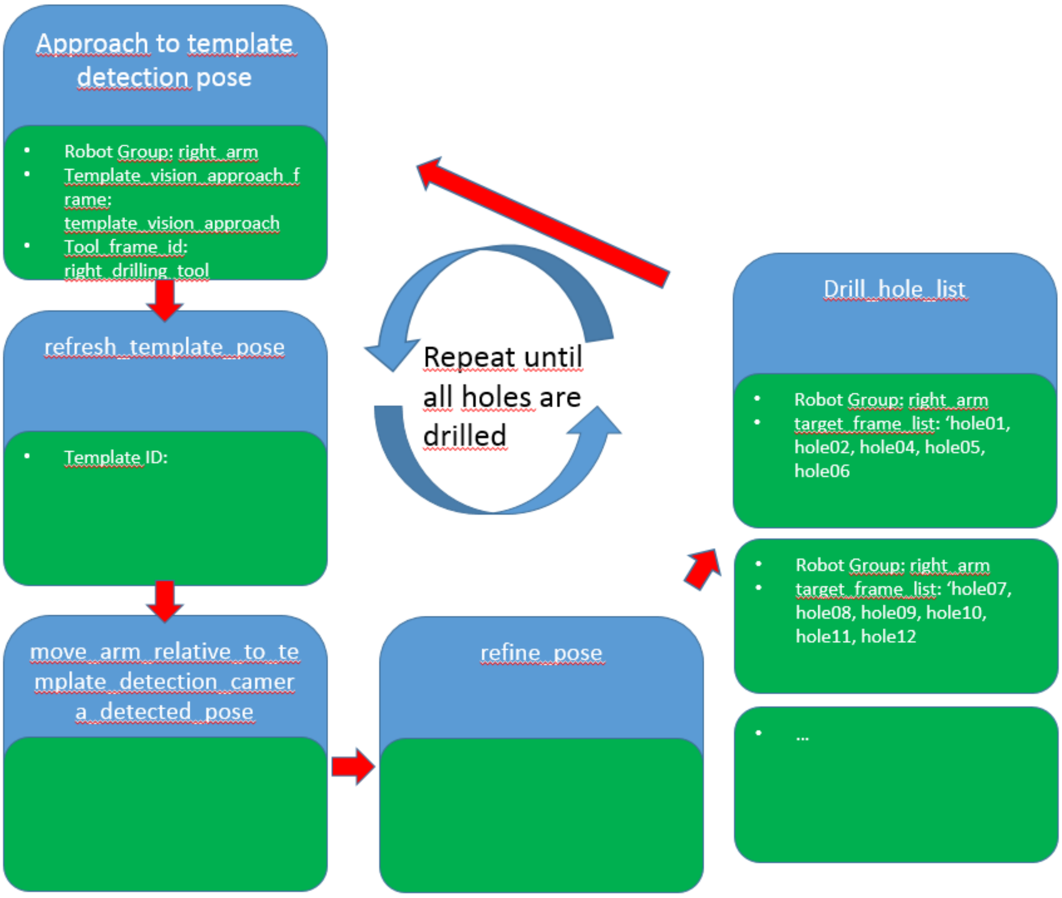


Figure 5: Skill sequence to perform all the drillings of a specific template

The detection and drilling of the holes is possible thanks to the skill sequence illustrated in Figure 5:

1. approach to template detection pose: Moves the drilling arm to a position where the camera can detect the template.
2. refresh_template_pose: This is the first detection done using the template detection camera. Used services are 'detect' and 'refine'. It is necessary to place the left arm in a position that ensures that the camera can see the whole template. Service 'detect' is called first. After obtaining a first position estimation, the 'refine' service is called using the same camera. This skill makes possible to get an initial estimation of the position of the template. The position returned by the 'refine' service is relative to the camera.
3. move_arm_relative_to_template_detection_camera_detected_pose: Using the pose detect by the previous skill, moves the robot arm, where the hole detection sensor is installed, to an adequate position to detect the first set of holes of the template. It uses the information of its parameter ´hole´ to move to a fixed position relative to the indicated hole.
4. refine_pose: Using the pose that the template detection sensor defined as base, the hole detection sensor makes a new detection focusing in the first set of holes to be detected. This gives an accurate position of the holes of interest.
5. drill_hole_list: Perform the drilling of the first set of holes given as argument to the skill.

The sequence of skills from 1 to 6 is repeated until all the drills are done. In each round, parameter ´hole´ of skill move_arm_relative_to_torso_camera_detected_pose and 'target_frame_list' of skill drill_hole_list are updated accordingly to refine and drill the correct holes.

2.2.2. Overall integration architecture

The integration of the process perception modules to the overall THOMAS architecture is currently under development following the guidelines of the integration design that has been documented in D5.1 "Methods for dynamic work balancing of human robot collaborative environments - Design".

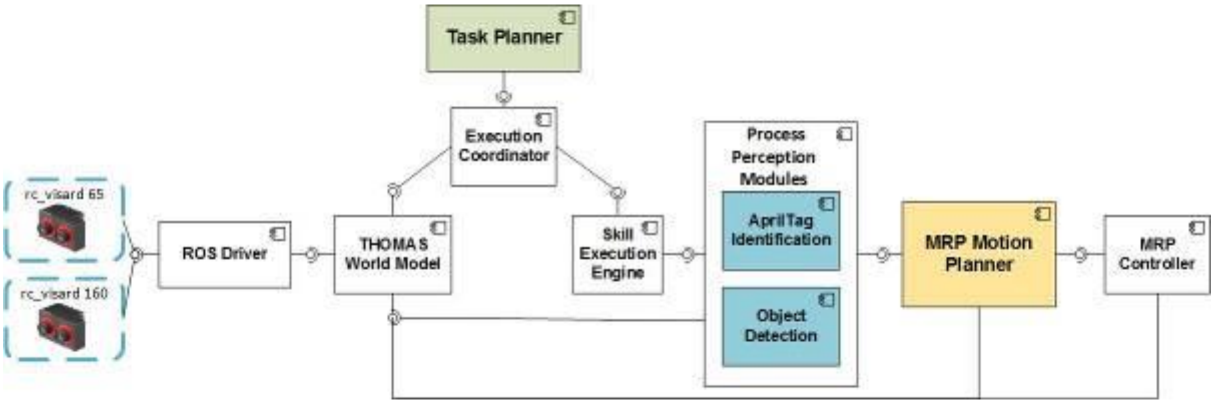


Figure 6: THOMAS Process Perception modules interaction with THOMAS system

As shown in Figure 6, the stereo cameras, *rc_visard* 65 and *rc_visard* 160, have dedicated interfaces to THOMAS world model through the provided ROS drivers by Roboception. In particular, once the *rc_visard* sensors have been connected in the system, THOMAS world model sensor manager is responsible for interfacing the sensors' ROS driver and registering their configuration data in the THOMAS data repository using the unified data model format. The generated 3D raw data, similar to the other involved sensors, are made available to THOMAS integrated system via the THOMAS Bus, using a publish/subscribe pattern as a communication mechanism as presented in Figure 7.

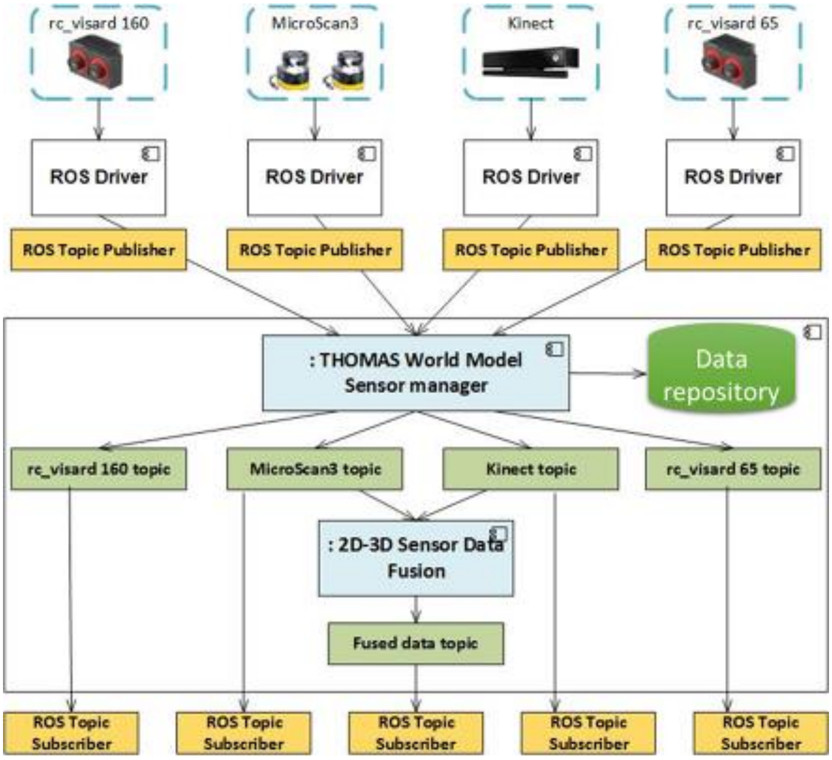


Figure 7: integration of *rc_visard* sensor into the THOMAS world model

The main contribution of the THOMAS world model sensor manager is the handling of multiple sensors “belonging” to multiple MRPs. In particular, each *rc_visard* sensor related topic (e.g. raw data publish topic) naming follows specific conventions in order to be identified and easily accessed by the interested modules, indicating also the resource ID where the sensor is attached. In the current implementation, two MRPs have been included in the system as visualized in Figure 8.

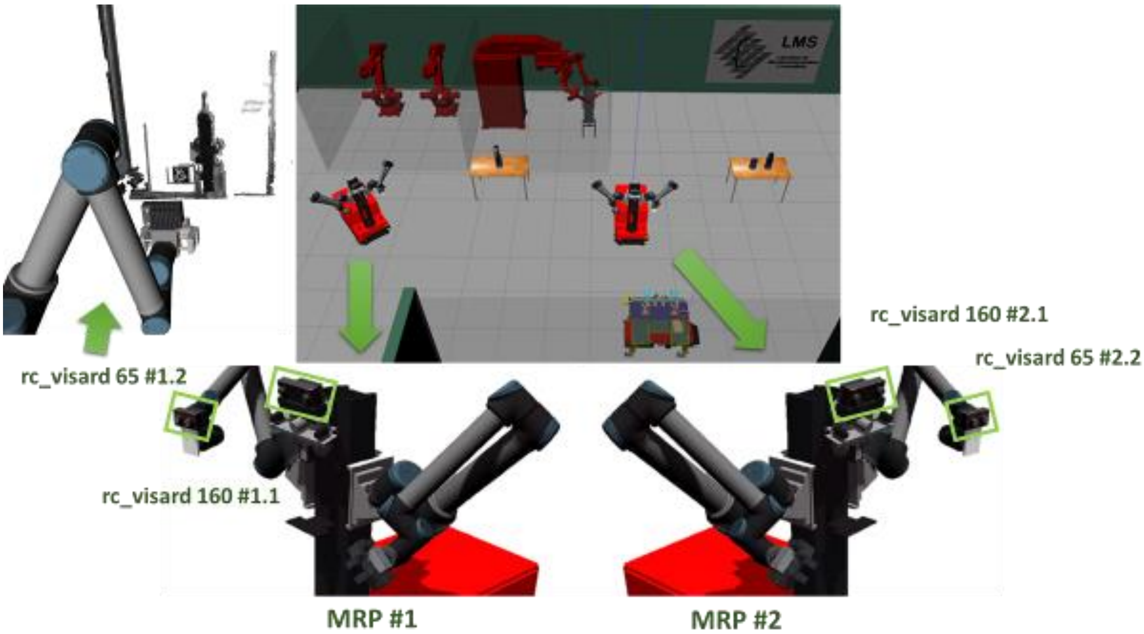
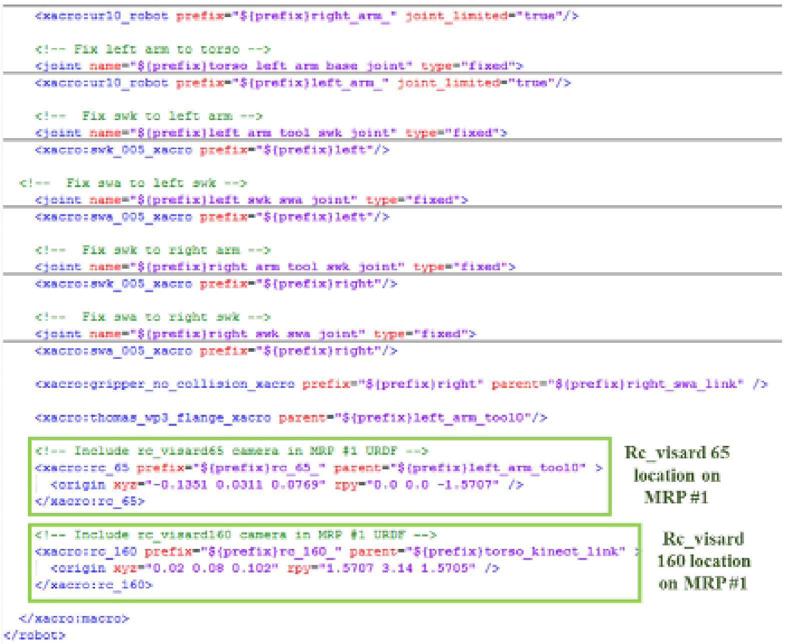


Figure 8: THOMAS World model – Multiple MRPs sensor handling

Each MRP has one *rc_visard* 160 attached in the torso and one *rc_visard* 65 attached in the right arm publishing raw data in dedicated topics. The exact position of the camera with respect to the related MRP is generated following the hand-eye calibration method provided by Roboception. The resource manager (see D5.2, “Dynamic work reorganization module – Initial prototype” for the detailed description) uses the extracted vector from the latter procedure in order to update the MRP URDF by adding an extra fixed link as visualized in Figure 9. The same course of actions is followed for all the sensors attached to all the involved resources, as well as to the stationary sensors.



```
<xacro:ur10_robot prefix="${prefix}right_arm_" joint_limited="true"/>

<!-- Fix left arm to torso -->
<joint name="${prefix}torso left arm base joint" type="fixed">
<xacro:ur10_robot prefix="${prefix}left_arm_" joint_limited="true"/>

<!--  Fix swk to left arm -->
<joint name="${prefix}left arm tool swk joint" type="fixed">
<xacro:swk_005_xacro prefix="${prefix}left"/>

<!--  Fix swa to left swk -->
<joint name="${prefix}left swk swa joint" type="fixed">
<xacro:swa_005_xacro prefix="${prefix}left"/>

<!--  Fix swk to right arm -->
<joint name="${prefix}right arm tool swk joint" type="fixed">
<xacro:swk_005_xacro prefix="${prefix}right"/>

<!--  Fix swa to right swk -->
<joint name="${prefix}right swk swa joint" type="fixed">
<xacro:swa_005_xacro prefix="${prefix}right"/>

<xacro:gripper_no_collision_xacro prefix="${prefix}right" parent="${prefix}right_swa_link" />

<xacro:thomas_wp3_flange_xacro parent="${prefix}left_arm_tool0"/>

<!-- Include rc_visard65 camera in MRP #1 URDF -->
<xacro:rc_65 prefix="${prefix}rc_65_" parent="${prefix}left_arm_tool0" >
  <origin xyz="-0.1351 0.0311 0.0769" rpy="0.0 0.0 -1.5707" />
</xacro:rc_65>

<!-- Include rc_visard160 camera in MRP #1 URDF -->
<xacro:rc_160 prefix="${prefix}rc_160_" parent="${prefix}torso_kinect_link" >
  <origin xyz="0.02 0.08 0.102" rpy="1.5707 3.14 1.5705" />
</xacro:rc_160>

</xacro:macro>
</robot>
```


Figure 9: MRP #1 URDF

The stereo cameras have been configured and integrated in the THOMAS world model during the offline phase. During the online execution phase, the process perception modules are subscribing to the respective stereo camera topics having a continuous interaction, as shown in Figure 6. Figure 10 presents the sequence diagram and information exchange among the different modules during the execution of a Pick_Up task, slightly updated with respect to the design presented in Deliverable D5.1, submitted on M12. The process perception modules are integrated in the system skill engine (see previous section for the implemented communication interface) while the skill engine is triggered by the Execution Coordinator (see D5.3 for the detailed Execution Coordination – Skill engine communication interface) for the execution of the required tasks.

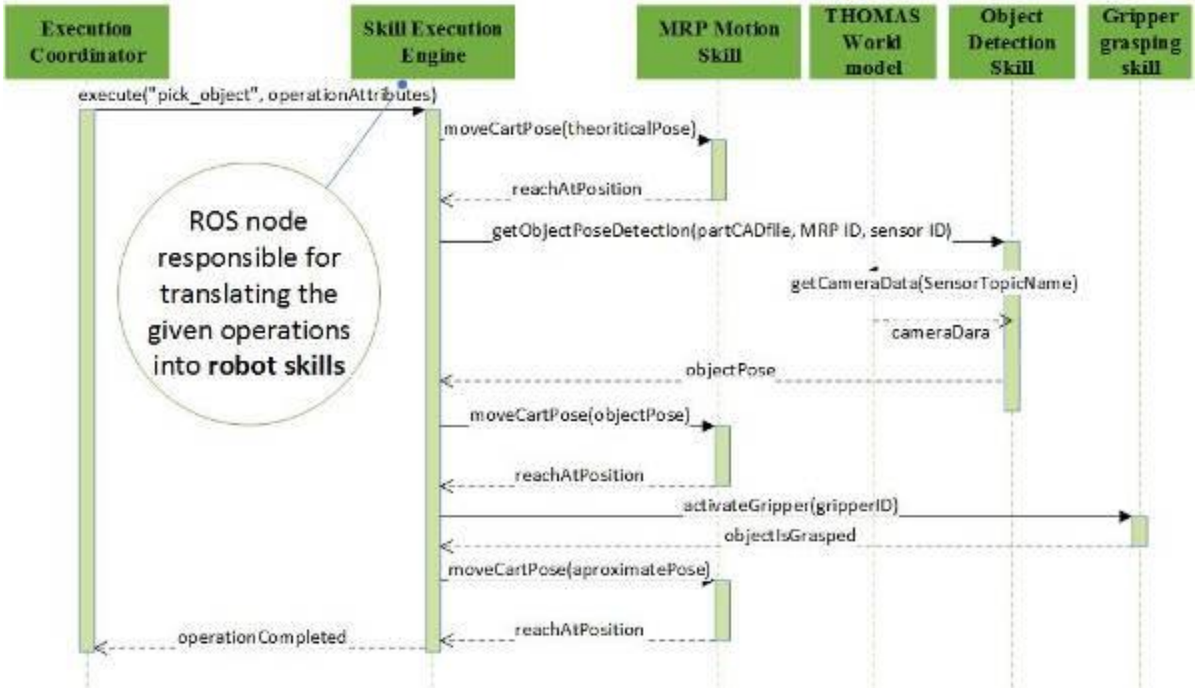


Figure 10: Pick_Up Task execution sequence diagram

3. AERNNOVA USE CASE

The use cases at the AERNNOVA plant were described and analyzed in D1.1, “Use-case definition and evaluation metrics”, and D1.2, “End users requirements analysis”. Three operations were analyzed: drilling, rivet inspection and paint sanding. The MRP platform has undergone several improvements to fit all the necessary devices (including the stereo cameras required for perception processes) to perform the tasks of each use case in the AERNNOVA plant. The initial focus for development of the perception components was placed on the drilling process, as explained in D1.2.

3.1. Description of testbed

To facilitate the development of the perception components, Roboception created a testbed to verify the performance of the developed modules on the selected operation. The setup includes a KUKA Agilus KR 6 R900 sixx, an *rc_visard* 160 with fixed position, and a pattern projector (video beam). Two templates were placed in front of the robot, on top of a wing section provided by AERNNOVA, as illustrated in Figure 11.

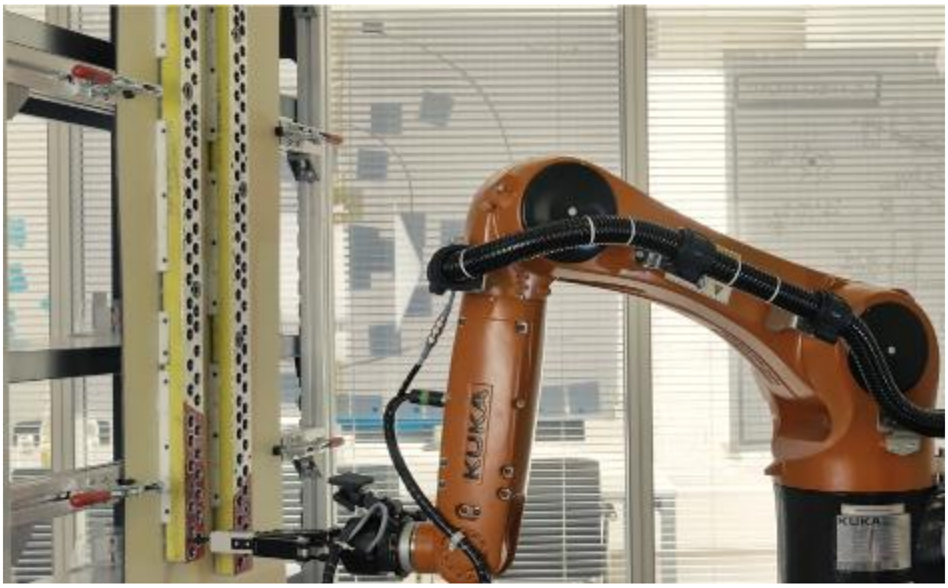


Figure 11. Testbed at Roboception for AERNNOVA use case

The detection process uses two images, one with a projected pattern (to improve registration of the pointcloud), and one without projected pattern (to acquire the raw image for the refinement step), as illustrated in Figure 12.

Figure 12. Acqusition of images with and without projected pattern

The pipeline described in Section 2.1.1 is then executed to locate the drilling templates. Each hole in the template has a corresponding coordinate system extracted from the CAD model, as shown in Figure 13. The location of each hole is then provided as output of the detection module, to perform the drilling process afterwards.

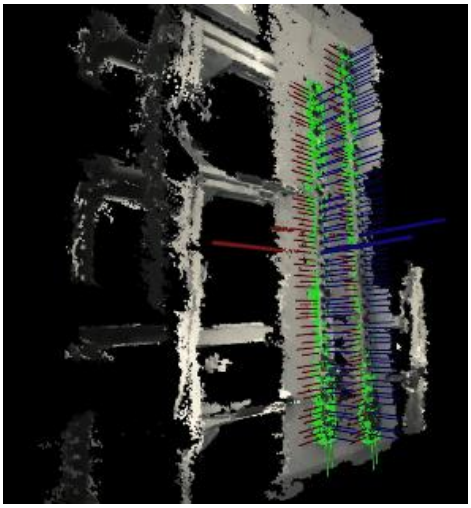


Figure 13. Detection of the template holes using the CAD based object detection process

3.2. Description of prototypical implementation

To perform the first tests for the drilling use case, an open space of 30 m2 has been reserved at Tecnalia, where the robot is able to navigate autonomously, perform a precise positioning in front of the structure to be drilled, and after that, drill the holes. For a correct execution, it is necessary that the robot is precisely positioned in front of the drilling structure. To do this, the Kinect camera placed on the upper torso of the robot and an accurate marker are used. The positioning algorithm is based on the detection and correction of the position of the MRP by a proportional control based on a given reference.

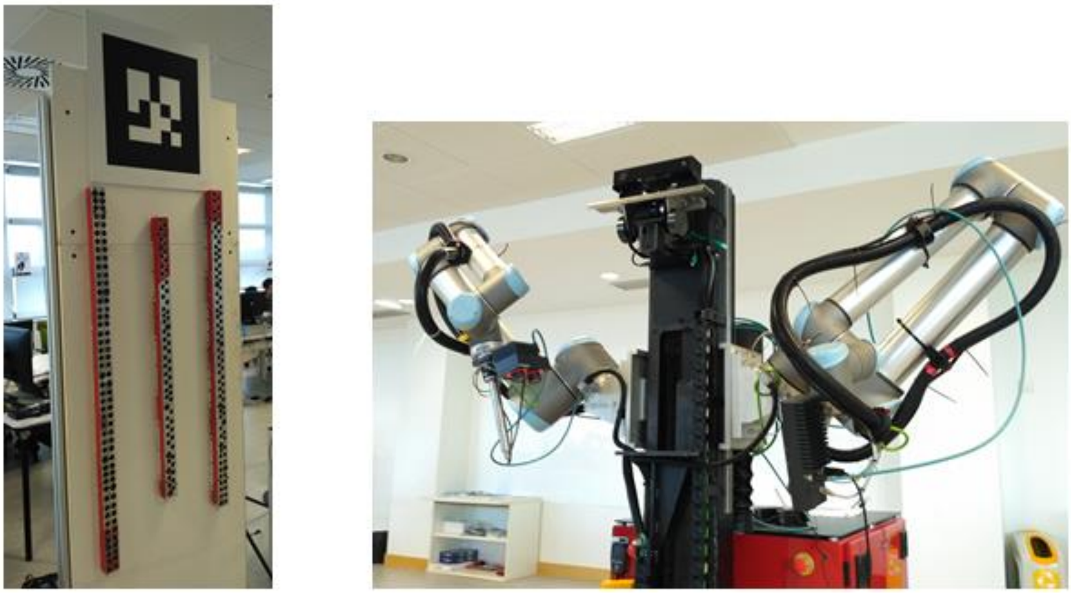


Figure 14: Precise positioning of the robot respect to the tag marker

Figure 14 shows templates in the structure placed vertically, thus recreating the real use case where templates are placed above the actual surface of the aircraft wing. Two sensors placed in the robot are used to reference the templates. As explained in section 2.2.1, the perception system has a sensor that references the template (template detection sensor), which needs more than one meter of distance between the robot and the template, so, after studying all the possibilities to place the sensor in the robot, it has been decided to place it in one of the arms of the robot. The other arm has mounted the precision sensor (hole detection sensor) that works optimally in the vicinity of the template, in a workspace easily reachable by the arm.

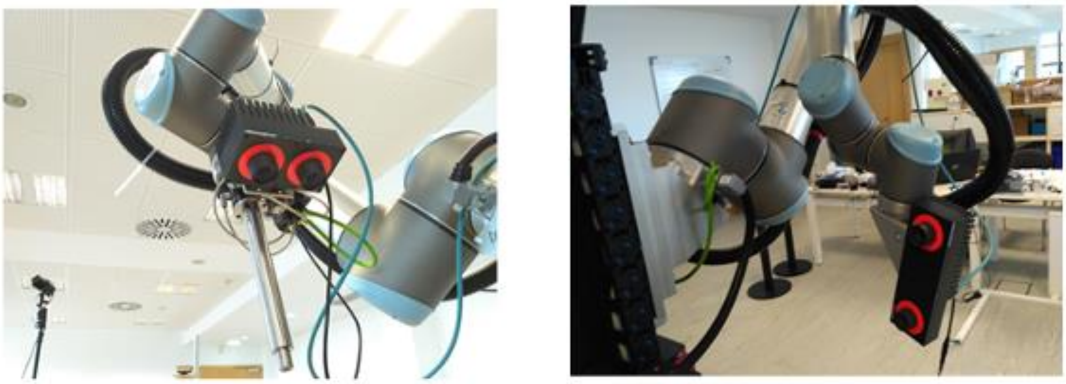


Figure 15: Hole/template detection sensors installed in the arms

To perform the drilling process, the robot incorporates an Advanced Drilling Unit (ADU). The ADU has been fixed to the robot together with a pneumatic system that provides the capability to change tools at run time and autonomously by the robot. To operate the ADU we use a vacuum source installed on the laboratory floor. This source provides a large amount of air, necessary for the proper operation of the ADU. Using the precise positioning technique discussed above, the robot is capable of docking in the station. This station supplies the robot with high air flow to operate the drill. The robot can successfully simulate the drilling process in different sections of the templates.

Figure 16: Drilling process

4. PSA USE CASE

In this section, the initial prototypical implementation for the Process Perception system testing under the automotive use case is described in detail. In particular, a physical set up has been deployed at LMS following the requirements of the PSA use case (Figure 17).

Figure 17. Scenario considered in the PSA use case. From left to right: parts on the Dolly, compression machine, and assembly and screwing on the AGV.

4.1. Description of prototypical implementation

THOMAS vision for the automotive scenario has been described in detail in D1.1, "Use-case definition and evaluation metrics" and updated in D2.1, "Perception for HR interaction - Design". The envisioned scenario has been graphically represented and validated in terms of feasibility using 3D simulation tools. Based on this analysis, the final version of PSA use case was constructed, the MRP will be responsible for picking and placing the damper in multiple positions of the cell as shown in Figure 18. In particular, the MRP will be responsible for damper's transportation between the pre-assembly area, the compression machine and the MPP. Towards enabling the adaptation of the robot behavior to environment changes, MRP arms will be equipped with the *rc_visard* sensor, implementing the Object detection functionality. The design of the required modules to perform damper's detection as well as the software interface of camera have already been introduced to deliverable D3.1 "Environment and Process Perception module - Design" while the initial implementation has been presented in Section 2 of this document.

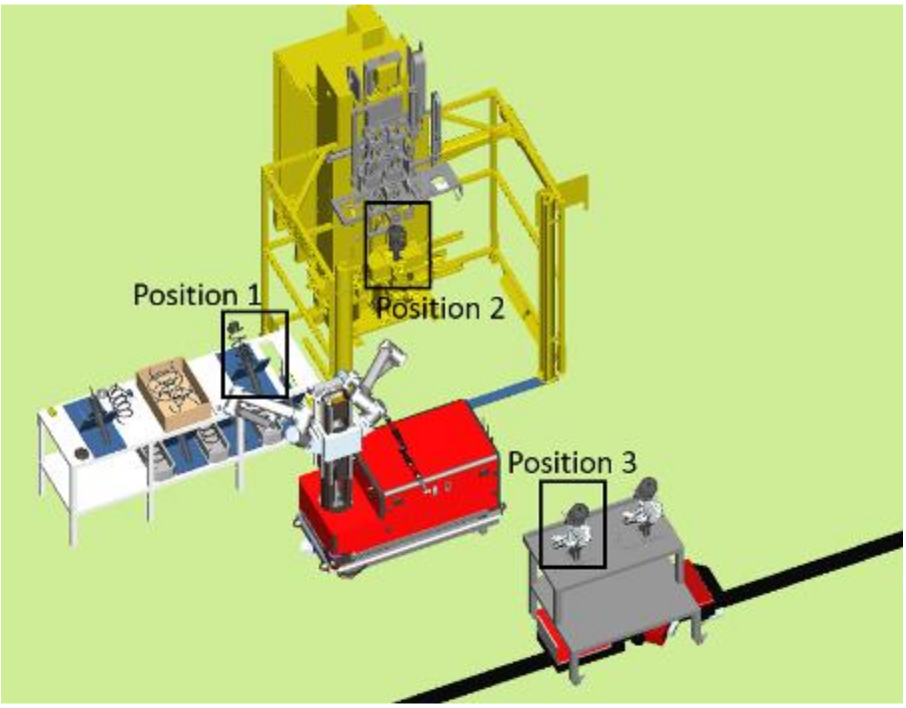


Figure 18. Scenario considered in the PSA use case. Position 1: Damper on the pre-assembly area, Position 2: Damper on the compression machine, Position 3: Damper on the AGV.

Based on the designed scenario, a physical prototype set up has been deployed at LMS for testing, validating and refining the damper detection functionality. The setup of the prototype cell is presented in Figure 19.

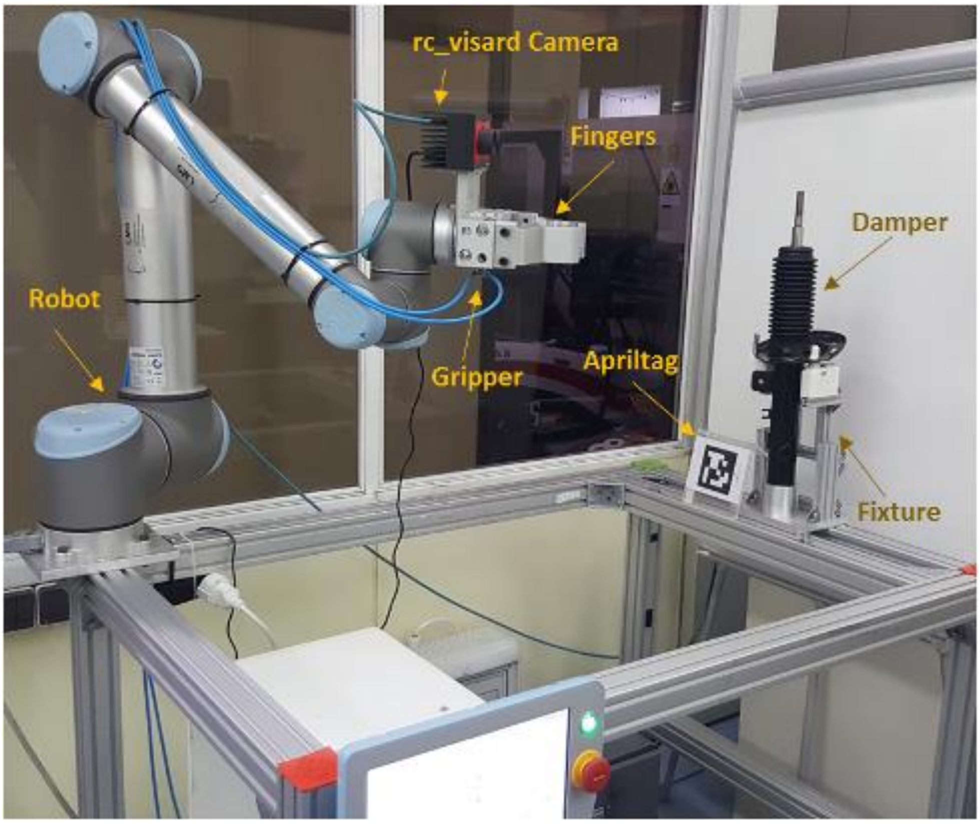


Figure 19. Prototype implementation cell at LMS

The produced cell has the following hardware components:

- A **stationary Universal Robot UR10 robotic arm**. The selection of the robot type was made specifically for covering THOMAS requirements, since the MRP will be equipped with two UR10 manipulators.
- The ***rc_visard* 65 sensor**. This sensor has been attached to the UR10 end effector flange in order to be able to detect the damper part in its field of view. The final position of the camera on the robot arm will be selected based on multiple tests under development.
- A **pneumatic parallel gripper –** SMC MHL2 25D has been installed as the UR10's end effector. The total weight of the assembled damper and the geometry of the compression machine fixture were factors considered for the gripper selection. However, the focus was the design of the gripper's finger in order to be able to grasp and manipulate the damper part. The final grasping point on the damper comes from the analysis of damper's geometry, as the larger grasping area on the body of the damper and the less uniform rotation relative to the grasping point are required. Based on the geometry of damper's body at the selected grasping point, a pair of fingers has been designed and produced by LMS. The functionality of the fingers was tested in the physical setup achieving the efficient grasping of the damper (Figure 20). Using these fingers placed on the parallel gripper and the data coming from damper's detection by the *rc_visard* sensor, the robot is able to move toward the damper and perform picking actions.

Figure 20. Damper manipulation operations. Left: gripper and fingers. Right: gripper's functionality

- **Compression machine fixture.** Towards replicating the compression machine environment and the pick & place operation parameters, the compression machine fixture has been replicated, manufactured and integrated in the physical set up based on the specifications provided by PSA, as shown in Figure 21

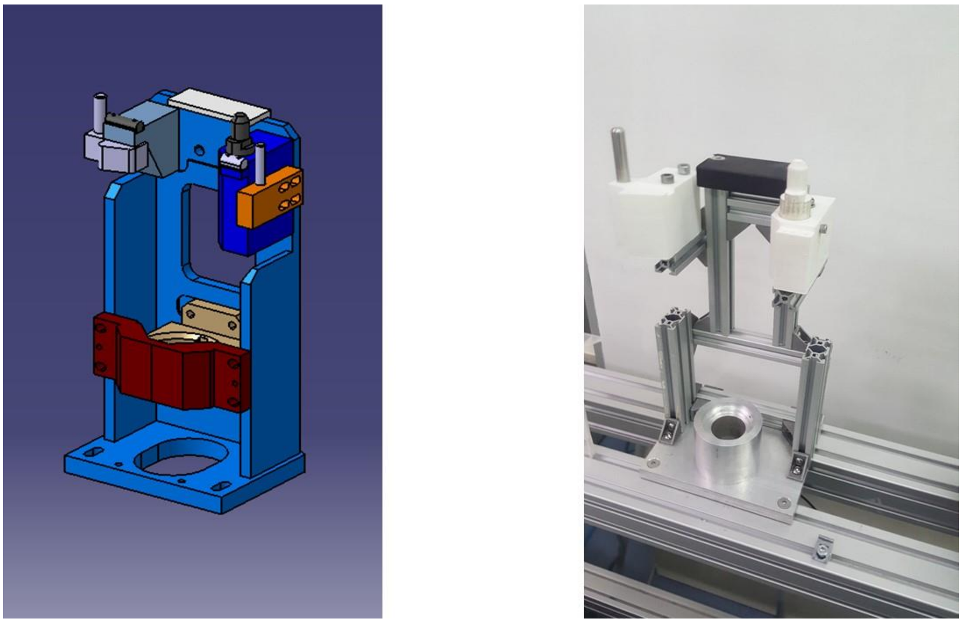


Figure 21. Compression machine. Left: Fixture from PSA's compression machine. Right: Fixture from LMS experimental scenario

Following the setting up and integration of hardware components of the cell, several tests haven been performed for the Object detection functionality following the steps detailed in Section 2:

- **Hand Eye Calibration process:** The output of the calibration process is one position and orientation vector, which describes the position of the origin frame of the camera relative to robot's toolframe. The hand-eye calibration process is presented in Figure 22. In this way, we ensure that the physical camera's position and the position of camera in ROS environment are the same. Using the updated robot's ROS transformation tree (TF), we are able to execute robot motions relative to the frame of the camera.

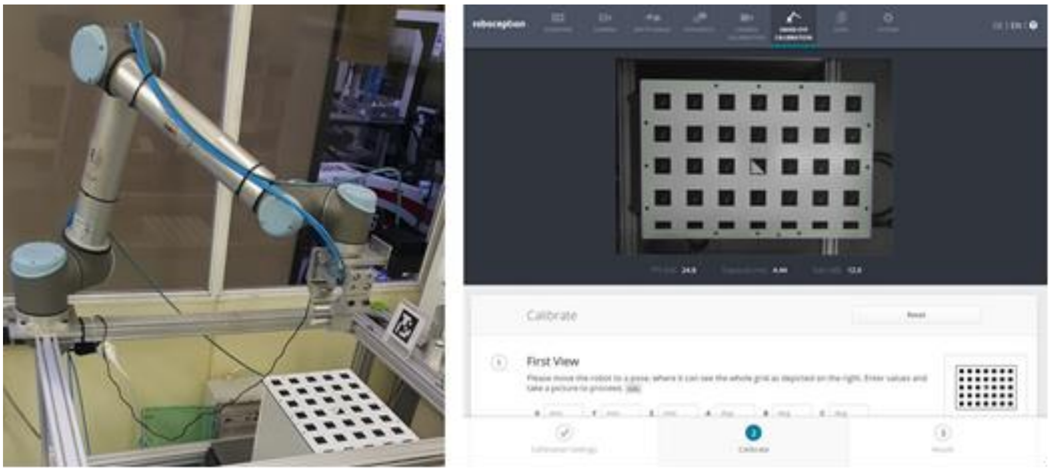


Figure 22: Hand-Eye Calibration Process

- **Apriltag detection:** At the beginning of the process, the camera detects the apriltag located at the left side of the damper and at the same height of damper's origin frame. In this way, using one transformation matrix between the damper's origin frame and one apriltag's corner, the position of damper relative to the apriltag is defined. Note that the apriltag only provides a rough estimation of the pose of the damper, which is refined in the next step.

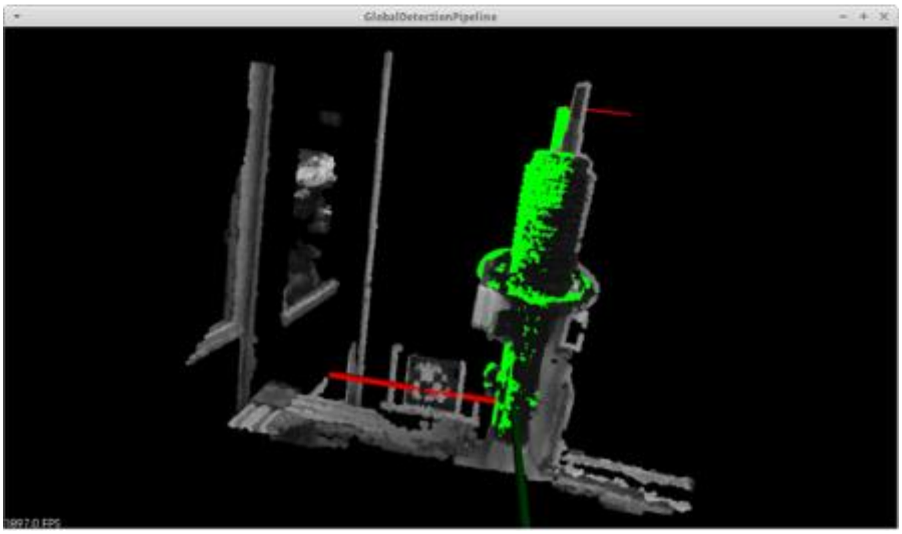


Figure 23: Apriltag Detection

- **CAD model matching:** To refine damper's position, this software compares the position of the physical damper and the position coming from the detection of the apriltag (Figure 24). After this comparison, the final position of damper is refined and the position of the detected grasping point relative to the camera frame is returned as output from the object detection algorithm.

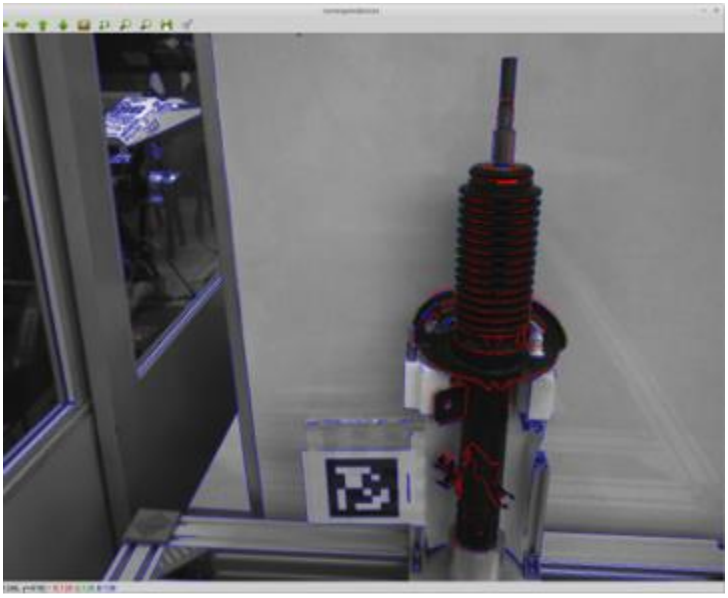


Figure 24. Apriltag detection and CAD model matching

- **Motion Planning to reach the detected point**: Given the coordinates of the detected grasping point, the motion planning ROS node using MoveIt is triggered (RVIZ visualization of robot motion is shown in Figure 25). Using the output from the object detection algorithm, the robot is able to move toward the detected point with deviations below 1 mm, and pick and place the damper.

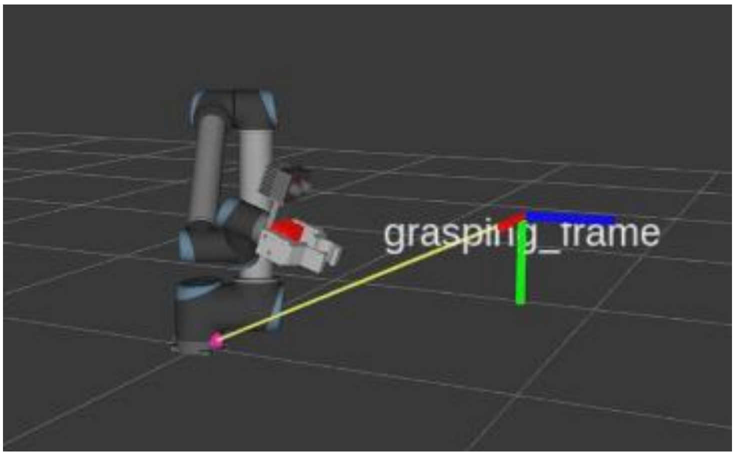


Figure 25. Motion planning and execution of robot to the detected grasping point

5. CONCLUSIONS

This document has presented the prototypical implementations of perception pipelines for object detection and process awareness within the THOMAS project. The description includes two modules, CAD based object detection and Apriltag detection. The software pipelines and integration with the skill engine and MRP control were also presented. Description of initial implementations were also included for three testbeds:

- Testbed for CAD based object detection using a KUKA arm at Roboception
- Testbed for drilling operation using the MRP at Tecnalia
- Testbed for object grasping using a UR arm at LMS

These testbeds serve to evaluate the functionality of the components in relevant scenarios, and will be further refined toward the final implementation of the perception skills on the MRP.

The work presented in this deliverable is work in progress, and the descriptions here presented might be modified, enhanced or improved during the remaining development time in WP3, environment and process perception.

6. GLOSSARY

MRP	Mobile Robot Platform
MPP	Mobile Product Platform
HRC	Human Robot Collaboration
HRI	Human Robot Interaction