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	:	THOMAS H-R safe interaction modules-Final Version
Reference	:	D2.5
Availability	:	Public
Date	:	11/12/2019
Author/s	:	TECNALIA, SICK, LMS, INTRASOFT
Circulation	:	EU, consortium

Summary:

This document reports the final implementations and tests of the components of the human detection and tracking system.

Table of Contents

LIST OF FIGURES

LIST OF TABLES

1. EXECUTIVE SUMMARY

This deliverable D2.5 "THOMAS H-R safe interaction modules–Final Version" reports the final version of the mechanisms allowing the interaction between humans and robots in the THOMAS scenarios.

The deliverable is of type demonstrator, reporting the actual implementations of the systems in the final versions of the various THOMAS systems. Thus, the deliverable focus more on implementation generals and testing, since more complete theoretical background and development details have been already reported in previous deliverables.

Several complementary technologies have been implemented, which can be classified as detection and interaction technologies:

- Detection:
  - 2D Laser based human detection, which takes advantage of the already installed safety laser scanners to detect and track human presence in the robot surroundings.
  - 3D based human detection, which uses 3D RGB-D cameras to monitor critical working spaces to detect and track human presence.
- Interaction:
  - Wearable device application or HRI, where a smartwatch application has been implemented to allow interaction between human operators and the production system, including the robots.
  - Human gesture and posture recognition, where a vision-based system is used to recognize human operator's commands, given using gestures.
  - AR based human operator support application, where the human operator is feed with relevant information about the robot and process status through wearable AR glasses.

2. Introduction

Barrier free co-existence of humans and robots in the workplace is one of the main goals of current industrial robotics research. From simply sharing the same space to full human-robot collaboration, two key elements are indispensable for this paradigm to be both efficient and safe: Robot's awareness of human presence and effective Human Robot Interfaces (HRI).

Human presence awareness is indispensable for safety, as robots need to know that there are humans in the surroundings and so avoid doing any action that can pose a risk to them. But it is also a requirement for co-working and collaboration, as the robot also needs to know what the human is doing, and what the human expects of it.

While the human can more easily rely on their own senses for robot's presence awareness, there are other some less evident information that humans can hardly get on their own, like the trajectory a robot will follow or the reach of its safety zones, that, if interfered, can have great impact on the system performance. Thus, to effectively share the space and collaborate with robots, humans need adequate HRIs to allow them to both obtain relevant information to the system and be able to provide commands to the robots in an easy, natural, fast and reliable way.

Workspace sharing and safe collaboration are key objectives of the THOMAS project. Several technologies are being explored as part of the project, are being tested in the developed prototypes, with are described in the following sections.

3. 2D LASER BASED HUMAN DETECTION

The primary objective of this research is the development of a module for the detection and tracking of humans that are located in close proximity of the MRP. This information can then be forwarded to and used by higher level modules such as the 3D-based human detection and integrated into the world model.

3.1. Motivation

As described in D2.4, the safety design of the MRP includes two SICK microScan3 safety laser scanners (Figure 1). Their primary purpose is to mitigate risks that are imposed by the movement of the MRP and its robot arms by monitoring pre-defined safety fields.

Figure 1: SICK microScan3 at the front left corner of the MRP

In addition to their usage as safety-rated sensors, their measurement data, which consists of a 2D laser scan that provides a discrete representation of the environment in a 2D plane, is retrieved and post-processed to generate further information and usage, such as the detection and tracking of surrounding objects and reflector or contour-based localization.

As one important part of this project investigates human-robot interaction, presence detection of humans near the MRP is of particular interest. Use cases based on 2D perception data mainly include preventive measures, such as obstacle avoidance during the navigation of the MRP's platform and also the robot arms or a preventive adaption of their speed. While the 2D laser scanners are limited in scanning only a 2D plane, they do cover a wide range that usually cannot be monitored with 3D perception sensors. In combination with 3D perception data, the previous cases can be extended to more complex use cases such as gesture and intention recognition.

3.2. Implementation

Please refer to the description of the detection and tracking algorithm implementation described in Deliverable 2.3.

3.3. Testing

After initial tests with a mock-up MRP platform at SICK in Hamburg (see Deliverable 2.3), the object detection and tracking system was integrated into the MRP at LMS in Patras and was subsequently tested and refined to cope with challenges of the new environment. A set of parameters are provided in order to fine tune the behaviour of the detection and tracking. In particular, Table 1 presents all the available parameters.

Table 1: 2D Laser based Detection parameters

Parameter	Description
INITIALIZATION_MIN_DYNAMIC_ RATIO	For a potential leg, what is the ratio of detected dynamic points that needs to be fulfilled, for this potential leg to be

	considered for object initialization. E.g at least 8 of 10 points need to be classified dynamic for a value of 0.8. o A low value for this parameter will lead to lots of false positives since stationary objects will be detected as well. If this is tolerable, it may be an option to detect stationary humans. *Range*: 0.0 - 1.0, *Recommend*: 0.8
INITIALIZATION_MAX_LEG_DIST	For two legs that potentially belong together, what is the maximum distance of the legs that is tolerated to initialize a new object.E.g. if two previously unknown legs are detected close to each other and they are less than this threshold apart, they will result in a new object being tracked. A large value will lead to more false detections since arbitrary potential legs will be merged. *Unit*: meter, *Recommended:* 0.5
DESTRUCTION_THRESH_POSITION_SIGMA	For a known object that is tracked over time, what is the maximum standard deviation of the position for the object to be valid. I.e. if an object is not detected for some time, the position variance will increase and if it gets too large the object will be deleted. A large value will lead to longer tracks in case an object is covered temporarily. A large value will also keep false positives for a longer time and lead to more false associations. *Unit*: meter, *Recommended*: 1.0
DESTRUCTION_STATIONARY_TIME_CONSTANT	Stationary objects are deleted after a certain time span to get rid of false positives. For a known object that is tracked over time, what is the time it needs to stand still completely until it is deleted. E.g. it will take at least this time before any known object is removed (if it is permanently detected, otherwise it be deleted due to the DESTRUCTION_THRESH_POSITION_SIGMA). A large value can be used to track humans for a longer time if they stop. A large value will also influence how long stationary false positives will be tracked. *Unit*: second, *Recommended*: 30.0
DESTRUCTION_STATIONARY_MIN_AGE	For a known object, what is the minimum number of scans after which the object may be deleted. Unit: Scans *Recommended* 125 (microScan3 runs at 25 Hz)
DEBUG_LEVEL	Level of output in the console. *Range*: 1 – 4, *Recommend*: 2

The recommended values are the same that have been applied in PSA mock-up demonstrator at LMS. Moreover, for visualization purposes and in order to be more intuitive the output of the detection, a visualization arrow marker for RViZ have been developed. Arrow’s direction represents the direction of the human and the length shows the magnitude of the velocity (Figure 2).

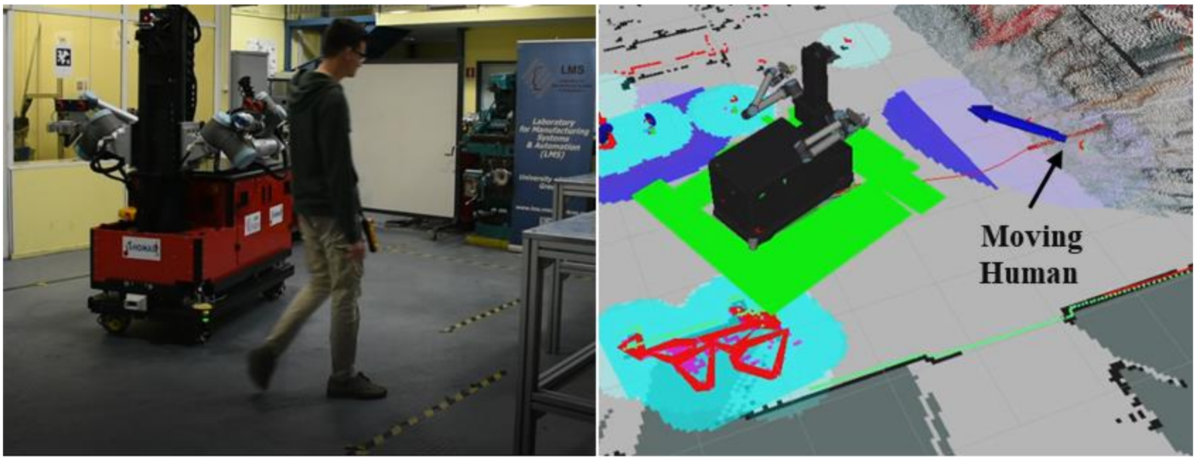

Figure 2: Representation of 2D human detection

3.4. Conclusion

The prototype of the 2D laser scanner tracking system has been developed with special focus on robustness and modular usage. First tests were carried out on a mock-up of the MRP at SICK, then the system was transferred to the actual MRP at LMS, which lead to further refinements in the object tracking.

4. 3D BASED HUMAN DETECTION

Collaborative environments are becoming the new common working areas in the manufacturing sector, turning the factories in spaces where robots and human operators work together. The time when the robots work isolated in enclosed environments is becoming a thing of the past as the robots become friendlier. To achieve the objective of full collaborative working plants, there are still some safety issues that must be take it in account. Our solution is based on the creation of a monitored area composed with an infrastructure of multiple RGB-D cameras.

The configuration of the workspace depends on the size of the working area of the robot, trying to maximize the view of the environment. For that three Intel RealSense D435 cameras have been positioned in different areas of the working area. These cameras allow to check how safe is the area surrounding the robot and depending on the situation the monitoring system can modify or abort the current task according to the situation.

A safety approach that implies a multiple camera configuration has several details that must be taken into consideration, like the coordination of multiple data sources, the saturation in the communications and the inconsistences detected. Multiple sources help to have a more complete overview of the workspace, as the combined fields of view avoids blind spots produced by the robot and other elements and covers all the area where the robot and the operators are.

However, multiple cameras can have overlapping fields of view. While this avoids blind spots, it also causes point clouds with overlapped layers of depth points. A filter that combines and create a new clean point cloud is used to solve this dispersion, concentrate the detection and reduce the amount of data to be processed. Also, points outside of the workspace are considered outliers and filtered.

To determine the risk level of each situation, several concentric safety areas are defined around the robot with associated risk levels from 0 to 3. They are visually showed in the interface using a 360º marker. When an operator is detected inside one of these areas, the execution can be continued, slowed or stopped depending on the risk level. Due to the mobility of the robot, the safe area moves along with it through the workspace. Thus, a static operator that is initially out of the risk area can also enter in it without realizing it when the robot is navigating.

4.1. Implementation

The system implementation has suffered some changes in latest stages of development. Currently the system is integrated with the robot in a real industrial environment in TECNALIA’s workshop, instead of the laboratory. This test environment is very similar to the final real test site in AERNNOVA’s demonstrator, including multiple workstations. Also, the hardware configuration has changed to an external PC based on ROS kinetic running on Ubuntu 16.04, using the required ROS packages for the Intel RealSense cameras. These packages are similar in functionality to the old OpenNI packages for the Microsoft Kinect V2. This similarity has eased the transition between the two types of cameras.

The setup of the workspace monitoring is made of three cameras positioned to get the best view possible of the work area. The MRP is equipped with a dual-arm configuration that allow it to perform multiples skill, both using a single arm or the two arms simultaneously. The mobile torso helps to work in multiple heights and positions, and a mobile base to navigate in the work area. This combination of dual arm and torso creates a pretty large reach volume around the robot. The defined safety area takes this into account. Moreover, as the robot is actually a mobile base, the safety area moves with it, being able to be dynamically adapted to the movement of the robot and the operational state of the arms.

Also, as the robot can move while the monitoring cameras are static, the distance between the robot and the cameras is variable. This distance can vary from 1 up to 4 meters. Spatial resolution changes greatly depending on the detecting distance, impairing the detection of the human in larger distances. Cameras are thus placed where the best performance region falls in the zone of the workspace with higher risk.

Optimal resolution from the cameras is 848x480 px for depth images and 1920x1080 px for RGB camera, generating a huge amount of data. To reduce the load in the USB 3.0 controllers, an additional

USB 3.0 PCI card was installed to the controller PC. Communications between the monitoring system and the robot is based on wireless technologies. This allows full mobility of the robot but increases latency and reduces available bandwidth. Because of that, some methods to manage the point clouds and reduce overall data transfers has been studied.

4.2. Testing

During this period the testing has been focused on to determine where are the best position for the cameras in the work area to get as much as possible view of the workspace. For that we have use the same approach that in the laboratory: we have tested using one, two and three cameras in different positions. With these tests we have been able to set them where get the better combined view of the workspace (Figure 4). Also, related with the latency, multiple tests and modifications have been performed like the down sampling of the point clouds or the reduction in the frame rates of the cameras (Figure 3).

There is still future work to increase the reliability, reduce even more the latency and to enhance the robustness of the detection.

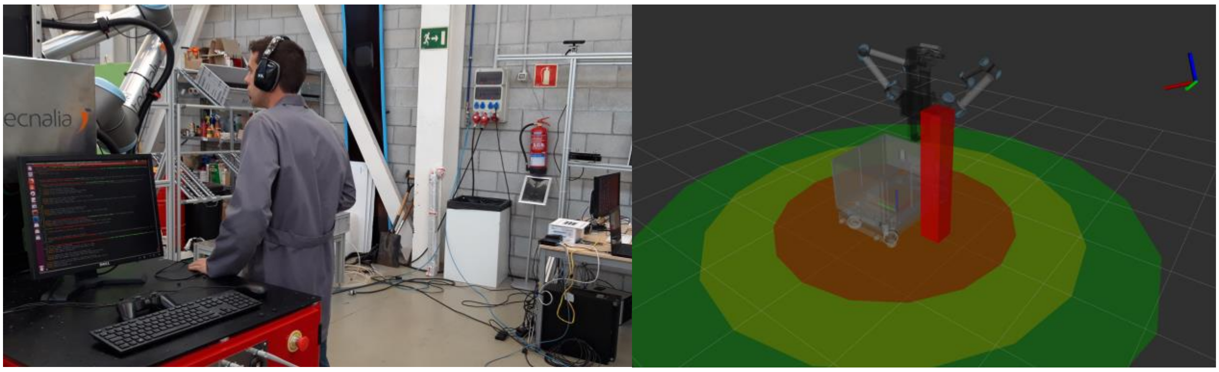

Figure 3: Test setup in TECNALIA's workshop. Human detected in the safety zone by the robot.

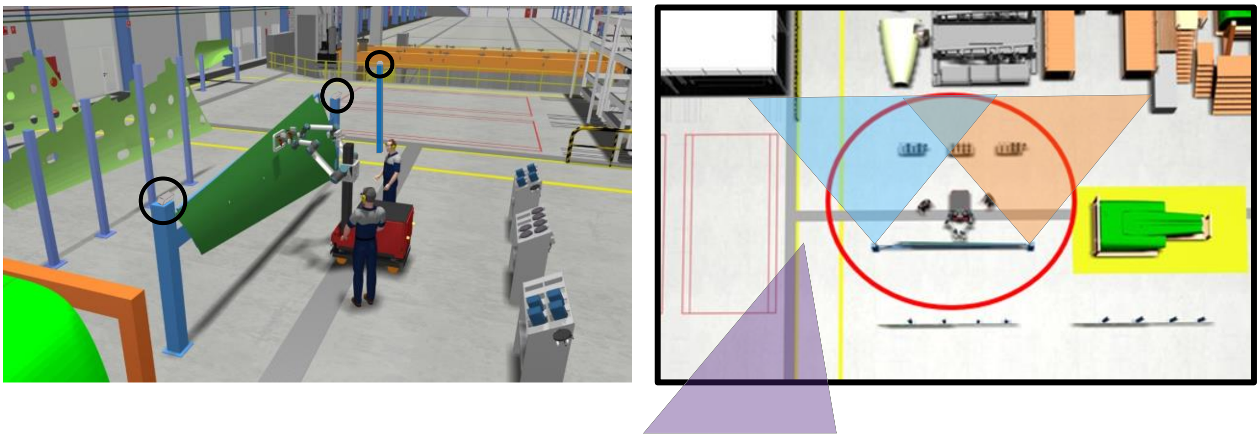

Figure 4: Proposed configuration and fields of view in AERNNOVA's scenario.

5. HUMAN – ROBOT INTERACTION

5.1. Robot control through gestures

Human Robot Interaction Mechanisms will be used to facilitate the "communication" of the human operator with the MRP during the task execution. Using either human posture or direct physical interaction, the system will receive input from the operator, through a set of sensors, and translate this information to specific commands for the MRP. In the following subsections, the main mechanisms are explained in detail.

5.1.1. General approach for Gestures Control

Gesture recognition approach initially has been designed and reported in D2.1. Various updates have been established, following the requirements of THOMAS pilot cases. However, the process pipeline maintained the same (Figure 5). The main input that is used for Human Robot Interaction is the human body itself. Multiple image sensors (2D and 3D) positioned on the robot or on the shopfloor are used, in order to achieve the maximum coverage. Afterwards, this human intention will be translated into robot command, sending the appropriate instructions to the MRP controller. More specifically, the steps that will be followed by this module are as follows:

- Receive the coordinates of each skeleton joint from the human detection module as input
- The Gesture Recognizer, will translate the gesture/posture/intention into predefined instructions
- These instructions will be published to the next component as input
- The Gesture Controller will receive this input and translate it to actual robot commands.

A general diagram of the above sequence is presented in the following figure (Figure 5). Its component of this pipeline will be analysed in the following sections.

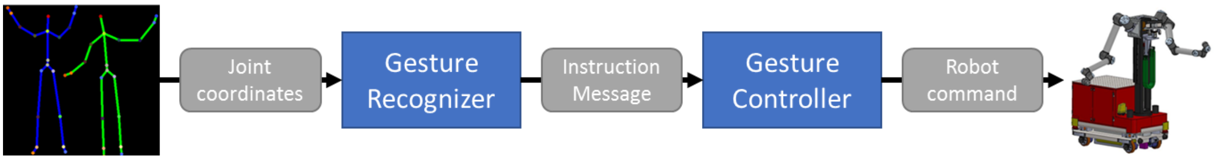

Figure 5: Human Gesture recognition pipeline diagram

5.1.2. Gesture Recognition

As it is presented in Figure 5, Gesture Recognizer is the first component of the MRP gesture control application. This component encapsulates two subcomponents that are work in sequence in order to derive the gesture instruction message from an image. In particular, Figure 6 presents the whole machine learning architecture of Gesture Recognizer. At the begging of the project, the approach for the Gesture control application was designed to work solely with Microsoft Kinect [1] as the main data acquisition sensor and to use the OpenNI library for human skeleton detection. This approach proved to be very restricted both in sensor selection and also in future maintainability because the OpenNI stopped to be updated. THOMAS solution foresees to provide universal solutions for many mobile robots with various sensors. A machine learning approach that works with every kind of sensors that provide a 2D image or a Pointcloud set in case of 3D information, is a universal and ease to integrate solution.

As presented in the architecture diagram below, OpenPose Deep Neural Network [2] is the first subcomponent of Gesture Recognizer. This part is responsible for extracting the human skeleton information out of an image. OpenPose library works with the principles of Machine

Learning. It serves a pre-trained deep neural network with COCO and MPI datasets, which are focused on human body parts. The results are extremely accurate even with a low-quality image as input. However, the greatest advantage that leads us towards this direction is the tolerance in various environmental conditions (lightning, position of sensor, etc.) and human characteristics (height, body shape, gender, etc.). In other words, OpenPose is an unbiased approach for computing the human body joint coordinates.

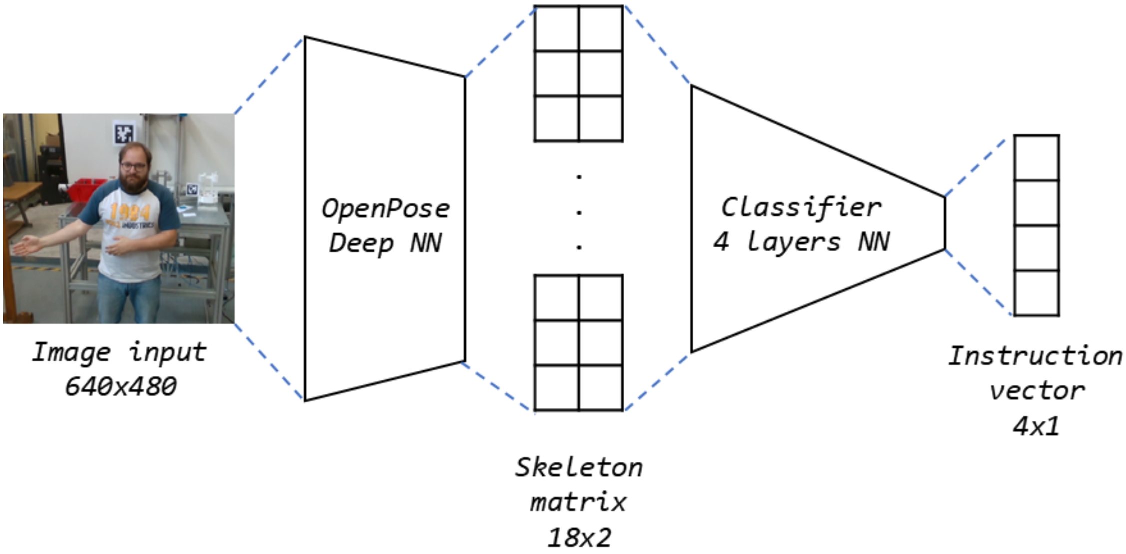

Figure 6: Gesture Recognition machine learning architecture

The outcome of OpenPose NN is a 18x2 matrix containing the positions of body joints, each row defines one body part as is listed in the Table 2. This body parts classification is established by the COCO dataset. After this classification and position tracking of human joints, a second smaller Neural Network with is responsible for classifying the gesture instruction.

Table 2: Body parts categorization based on COCO

Row number	Body parts
1	Nose
2	Neck
3	Right Shoulder
4	Right Elbow
5	Right Wrist
6	Left Shoulder
7	Left Elbow
8	Left Wrist
9	Right Hip
10	Right Knee
11	Right Ankle
12	Left Hip
13	Left Knee
14	Left Ankle
15	Right Eye
16	Left Eye
17	Right Ear
18	Left Ear

This NN takes the skeleton matrix with size 18x2 and provides as an output a 4x1 vector (Figure 6). This is the instruction vector and includes all the gestures sorted based on their detection likelihood values. The four gesture instructions are pre-defined and the NN is trained with a labelled dataset. Figure 7 presents four indicative images that used for the training process. Current gesture recognizer is trained to identify the following gesture instructions: 1) Front 2) Back 3) Right and 4) Left. After the accurate detection of an instruction, Gesture Controller takes over in order to convert it into an MRP command.

Figure 7: Example of gesture instructions 1. *Front* 2. *Back* 3. *Right* 4. *Left*

5.1.3. Gesture Controller

The final subcomponent of this application for controlling through gestures the MRP, is the Gesture Controller. As it has been reported in D2.1, initially gesture control was designed and tested for moving the robotic arms on MRP. Gesture Controller developed further in order to be able to control also the mobile platform. The aforementioned gesture instructions are mapped with MPR's platform relative motion and further converted in velocity commands as showing the Table 3. Based on testing of the system, it has been identified the need of maintaining constantly a relative position between the detected person and MRP's platform. Figure 8 presents the distance (d) and theta (θ) variables that gesture controller trying to maintain constant. In particular, the desired angle theta has been pre-defined and should be close to zero with 5^o tolerance. From the other side the distance between MRP and human, is defined according to the initial detection at the moment that the Gesture application is enabled.

Table 3: Gesture mapping with MRP's platform velocity

Gesture Instructions	Velocity command
Front	x axis: 0 m/s y axis: 1 m/s
Back	x axis: 0 m/s y axis: -1 m/s
Right	x axis: 1 m/s y axis: 0 m/s
Left	x axis: -1 m/s y axis: 0 m/s

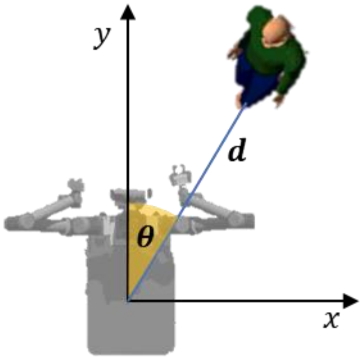

Figure 8: Relative position of Human with MRP's base link

5.2. AR Operator's Support Application

Another important part of the Human Robot interaction is the visualization of information to the human operators in the most intuitive and direct way possible. In order to achieve that, AR technology has been selected due to its immersive characteristic to blend to the real-world virtual data. In particular, Microsoft's HoloLens AR glasses have been used implementing a markerless based object visualization approach that may increase the application's stability and user experience.

5.2.1. AR Application Functionalities

With this application for the HoloLens glasses, we try to exploit the latest advancements in AR technology for implementing novel interfaces for human robot interaction while closing the communication loop between human operators and the robot resources. In this context the suggested application, and the framework around it, enhances human robot interaction by allowing human workers:

- To directly instruct the MRP robot: a) during execution in cases of unexpected / unplanned events, b) for short term re-programming requirements when changes occur in the production environment,
- To receive real time information: a) on robot active tasks, b) about his / her assigned tasks.
- To provide feedback on the real time execution status in the central execution control system.

The following sub-sections provide a deep insight in the developed functionalities and their added value towards supporting human operators during the assembly process. For the navigation in the AR environment, the AirTap gesture has been defined as user input for all the virtual buttons included in the application.

5.2.1.1. User initialization phase

Given that in the THOMAS use cases each operator needs to work in several workstations, the discussed framework has been designed so as not to relate to the particularities of a specific workstation. In addition, variations in the different operators' characteristics needed to be considered in order to make sure that all the digital objects are superimposed in the field of view of the operator in the correct scale and position with respect to the physical world. After several experiments, the height of the operator who wears the AR glasses has been identified as a variable that needs to be initialized for each different user. Thus, for every new user, an initialization phase is required as visualized in Figure 9. The user is instructed to spot a marker placed in a specific location in the assembly area and based on the distance of the camera from this marker the required height is calculated. Then, the user is ready to start using the AR tools deployed in the AR glasses.

Figure 9: Human operators' field of view - User initialization phase

5.2.1.2. Robot Instructing phase – Direct robot control

One important limitation in existing robotic applications is that the robots need to be offline programmed by robot experts with high accuracy based on the specific layout and the parts involved in the assembly. If changes occur either in the assembly layout or in the product variants, the production needs to stop until a robot expert may manually re-program the robot. This creates losses in terms of cost and time that have a great impact in productivity especially in the cases of SMEs. The discussed flexible production paradigm aims to overcome the limitation providing the human operator the mechanisms that will allow him / her to directly instruct the robot in an easy and fast way when needed, without having any expertise in robotics. Two different functionalities have been implemented comprising this robot instructing phase.

Direct robot navigation instructions

The first functionality allows the user to give new navigation goals to the MRP which they were not initially programmed. In that way, the MRP may be online allocated to new workstations when this is needed based on the production requirements. As visualized in Figure 10, the user may simply AirTap in the desired location for the MRP. This user input is transferred in real time in the MRP path planner which generates the optimized path for achieving this new navigation goal. Then, the planner sends this path to MRP controller for the final execution.

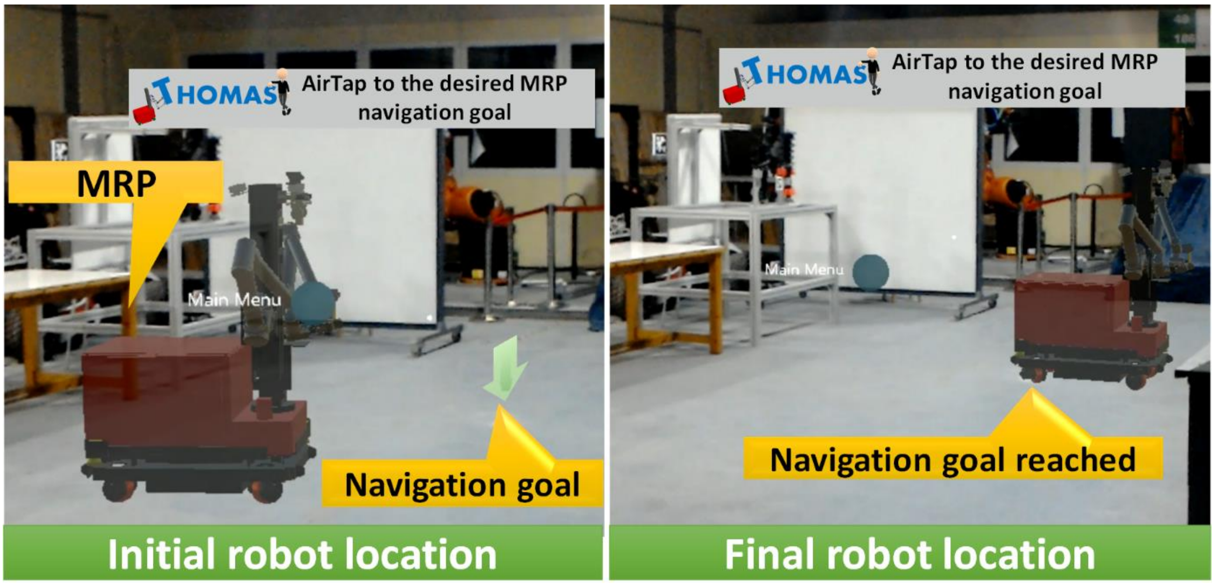

Figure 10: Human operators' field of view - MRP direct navigation instructions

Robot position corrections – Teleoperation

The second functionality aims to allow the user to make small adjustments and real time corrections in the mobile robot's location. This may be useful considering the dynamically changing environment and the non-static positioning of the resources. This teleoperation is implemented by visualizing to the user a cross pad composed of four virtual buttons. When the user AirTaps on one of these buttons, the MRP moves in the respective direction based on a pre-defined offset. As the mobile robots moves and possible rotates, the pad is rotated as well to have always the correct orientation with respect to robot's platform orientation. Through the available buttons the user may request robot position correction by: a) moving forward, b) moving back, c) rotating to the left, d) rotating to the right. For instance, the user in Figure 11 instructs the robot to rotate in the left so to ensure better reachability of the robot arms in the workbench in front of the platform.

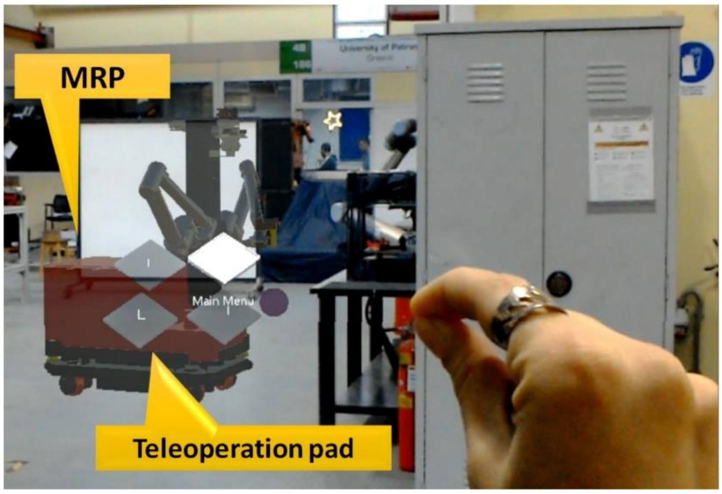

Figure 11: Human operators' field of view - MRP position corrections

5.2.1.3. Execution phase – Assembly status information exchange

In conventional fully automated robot-based assembly systems the process execution control and coordination may be realized through various approaches such as Programmable Logic Controller (PLC) based or service oriented based control architectures. Nevertheless, when human workers are also part of the assembly, two important requirements occur:

- Provide them information on their assigned task as well as provide them interfaces for reporting back in the execution system the execution status
- Inform them on the robot active task in order to be alert with respect to robot real time behaviour and thus increasing their safety awareness.

This information exchange is achieved through the integration of the AR based framework with the central execution system, namely the Station Controller.

Robot active task execution information

Once the assembly tasks have been dispatched to the resources and the execution has been started, the human operator may request to receive information on active tasks in each workstation. Figure 12 presents the field of view of the operator enhanced with information on the current active task, the assigned resource as well as the task execution status.

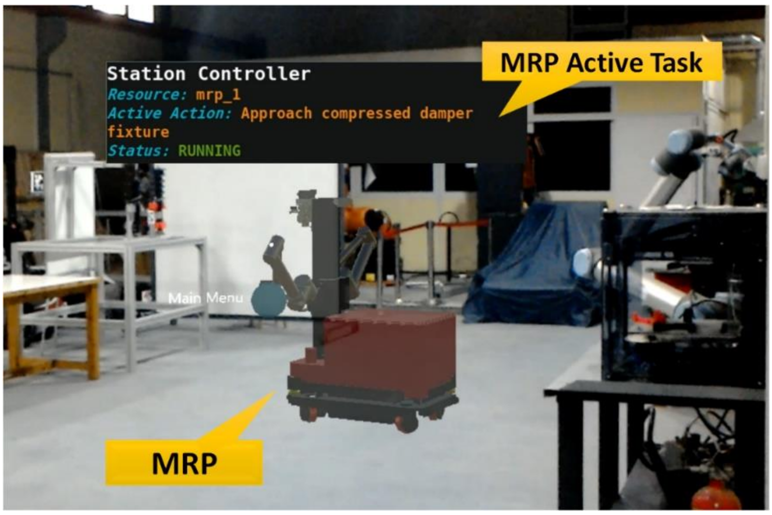

Figure 12: Human operators' field of view - MRP active task information

Human operator's assigned tasks information

Respectively, when a task is dispatched to the specific human operator, he / she receives a notification along with a textual description on the assembly task that needs to be performed as shown in Figure 13. In complementarity, a virtual button, namely the "Task Completed" button, is superimposed in his field of view allowing him to notify the Station Controller when he / she has completed the assigned task. In that way, the Station Controller may efficiently coordinate the entire assembly processing execution respecting the precedence relation among the different tasks to be performed by the different humans and robot resources.

Figure 13: Human operators' field of view - Human operator's active task information

5.2.2. Integration in the THOMAS overall system architecture

A vital aspect of the AR application is its integration to the THOMAS system architecture. For the support of this application two additional components have been used:

- A Digital Twin of the production environment involving: a) the scene reconstruction based on the CAD models of the layouts and the involved components as well as the real time data of the sensors placed on the MRP, b) the interfaces to MRP's path planner for requesting optimized paths giving as input the real time sensor data.
- The Station Controller responsible for dispatching the scheduled tasks and monitoring the execution status through the robot side and human side interfaces.

Considering the above, with respect to the end-to-end system integration, the main challenge that had to be met was the diversity of these systems in terms of software programming and communication channels compatibility. For encountering this complexity, the discussed solution deployed a networking architecture based on Robot Operating System (ROS). The ROS framework, running in Linux environment provides a standard communication infrastructure based on topic publisher / subscribed paradigm customized for robotic systems. The Digital Twin and Station control are developed in C++ directly compatible with ROS system and may be deployed in any Linux PC. The AR based functionalities were created in Unity 3D game engine using Microsoft Mixed Reality Toolkit for UWP applications running in Windows Operating System.

All the data that are exchanged between the a) AR application, the b) Digital Twin, c) Robot's planner interface and d) Station controller are in form of ROS messages enabling the use of topics and services. For this communication to be realized, ROS# library was used establishing a ROSBridge server. This server allows the communication of non *nix system such as the AR tools with a ROS based environment. In that way, the AR tools have direct access to the following information:

- robot's Universal Robot Description File (URDF) file that enables the robot's visualization as well as manipulation based on the robot kinematics,
- robot's base position in the global map for the accurate superimpose in the physical world,
- robot's status information for robot behaviour awareness,
- execution status and status feedback provision.

The AR based tools were deployed in Microsoft HoloLens glasses used as the human side hardware interface. The Digital Twin, the Station Controller and the robot side software interfaces were deployed in a Linux PC running Ubuntu 16.04 and ROS Kinetic version. The connection between the hardware components involved was established through a local Wi-Fi network.
Figure 14 visualizes the exchange of information for the realization of the Direct navigation instructing functionality.

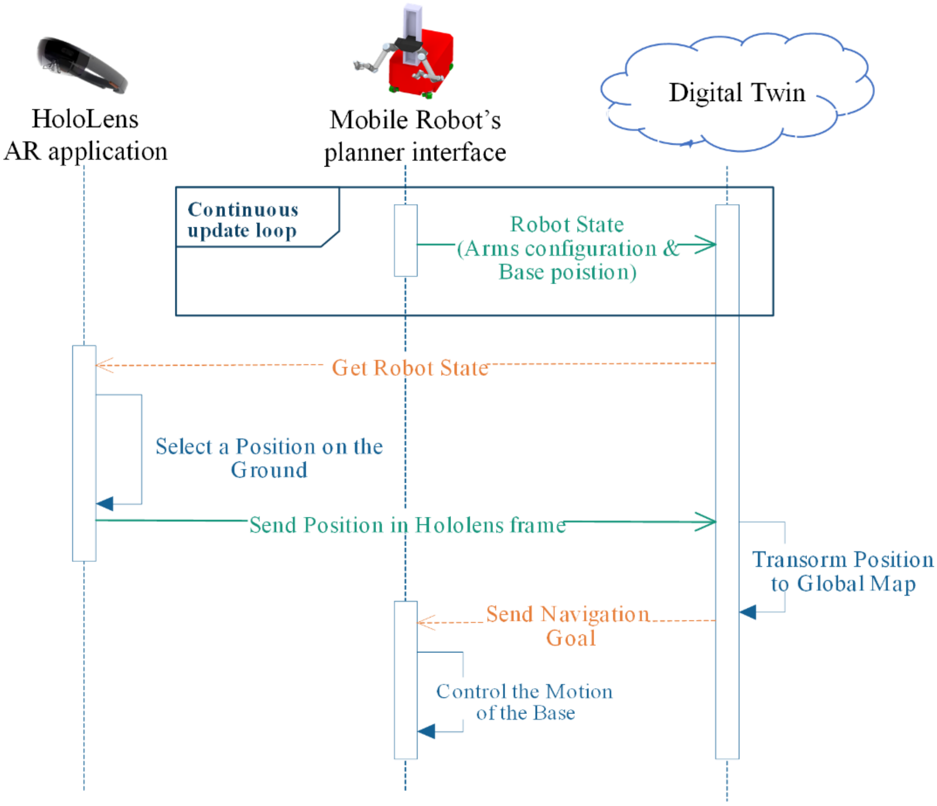

Figure 14: Sequence diagram for robot direct navigation instructing

5.2.3. Prototype Implementation

The discussed AR based framework has been tested and validated in the Laboratory for Manufacturing Systems (LMS) premises. To be able test the application in a realistic robotics set up in terms of 3D layout, a GAZEBO – ROS based - simulation was set up replicating the assembly environment. The digital models of the MRP (URDF) and human (CAD) where added in the simulation integrating the human side interface and robot controller in the backend.

Figure 15 visualizes the re-construction of the assembly environment based on the sensor data of the two laser scanners located in the mobile platform of the MRP and the Kinect located on its torso. In this way, any instruction send to the robot from the human through the AR tools, is transferred to the robots through the Digital Twin for ensure collision free paths for the new robot actions.

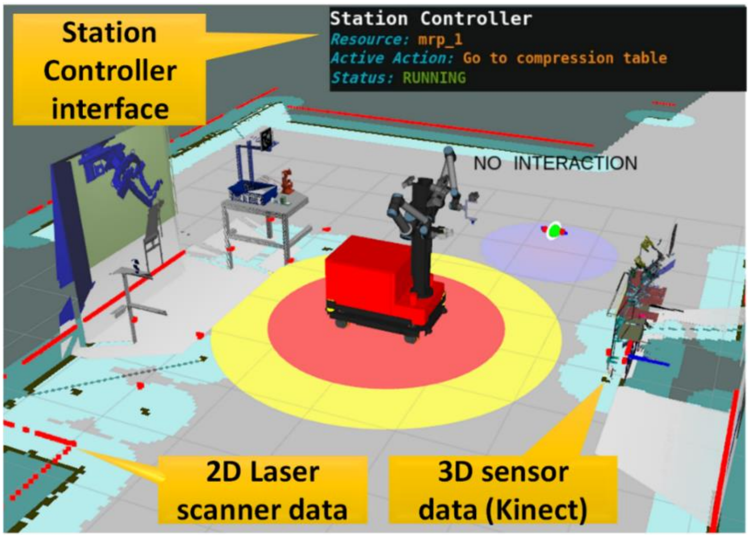

Figure 15: Digital Twin scene re-construction

The developed AR tools, which run on the HoloLens AR glasses allow the operator to visualize the MRP digital model in the physical world and directly interact with it by: a) teaching it how to reach unknown workstations and b) be notified on robot and human operators assigned and active tasks. Figure 16 presents the visualization of the a) Physical world – field of view of the operator, b) Digital Twin and c) Simulation environment during the different stages occurred in the environment while the human instructs the mobile robot on how to reach the final assembly area.

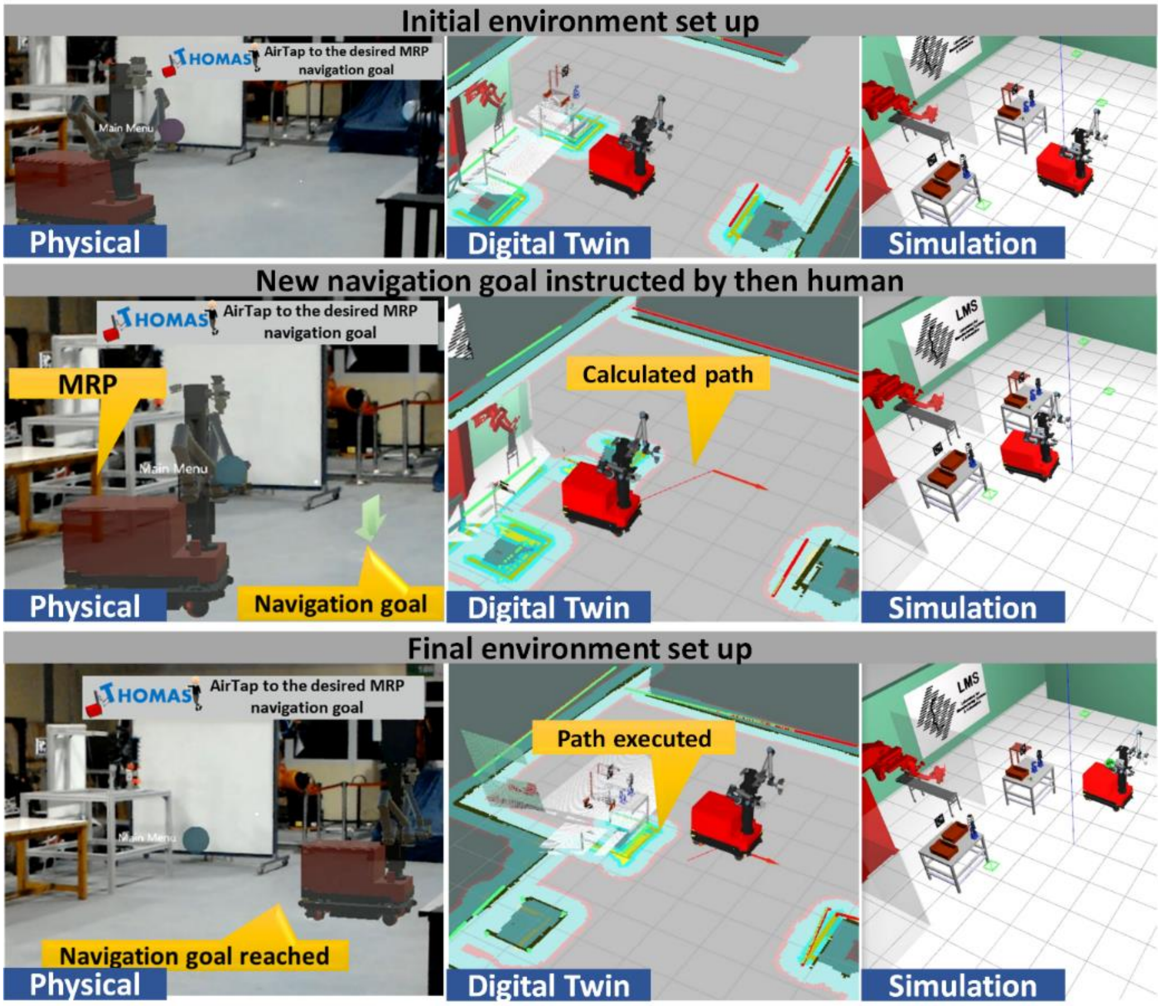

Figure 16: Direct navigation instructions functionality workflow

6. CONCLUSIONS

Workplace sharing and collaboration between robots and humans requires both human presence awareness and appropriate means of communication between operators and robots. These two technologies are also a key element in the THOMAS project.

The report shows the last implementations of several of such systems in the final THOMAS prototypes. In the case of human awareness, a double system of 2D and 3D based detections is being tested. HRI, on the other hand, is explored in different ways, like wearable apps, human gesture recognition and AR based support applications.

Current testing shows a good performance of the prototypes. Further testing and development will be done during the preparation of the use case demos.

7. GLOSSARY

2D	Two-Dimensional
3D	Three-Dimensional
AR	Augmented Reality
CAD	Computer Aided Design
HRC	Human Robot Collaboration
HRI	Human Robot Interaction
MPP	Mobile Product Platform
MRP	Mobile Robot Platform
PCI	Peripheral Component Interconnect
PLC	Programmable Logic Controller
RGB-D	Red-Green-Blue-Depth
URDF	Universal Robot Description File
USB	Universal Serial Bus
UWP	Universal Windows Platform
ROS	Robot Operating System
NN	Neural Network

8. REFERENCES

[1] "Microsoft Kinect," [Online]. Available: https://developer.microsoft.com/el-gr/windows/kinect.

[2] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei and Y. Sheikh, "OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields," *arXiv p,* no. 1812.08008, 2019.

[3] K. C. Fuerstenberg, "Fahrzeugumfelderfassung und Fußgängerschutz unter Nutzung mehrzeiliger Laserscanner," *Open Access Repositorium der Universität Ulm,* 2009.

[4] A. Elfes, "Using Occupancy Grids for Mobile Robot Perception and Navigation," *Computer,* pp. 46-57, June 1989.

[5] M. Kohler, "Using the Kalman Filter to track Human Interactive Motion -- Modelling and Initialization of the Kalman Filter for Translational Motion," 1997.

[6] D. Arbuckle, A. Howard and M. Maja, "Temporal occupancy grids: a method for classifying the spatio-temporal properties of the environment," in *IEEE/RSJ International Conference on Intelligent Robots and Systems,* 2002, pp. 409-414 vol.1.

[7] A. Leigh, J. Pineau, N. Olmedo and H. Zhang, "Person tracking and following with 2D laser scanners," *2015 IEEE International Conference on Robotics and Automation (ICRA),* pp. 726-733, 2015.