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	**:**	**Perception enabled navigation and docking for mobile robots: Initial Prototype**
Reference	**:**	**D3.4**
Availability	**:**	**Public**
Date	**:**	**17/10/18**
Author/s	**:**	**TECNALIA**
Circulation	**:**	**EU, consortium**

Summary:

This document provides a description of the first prototypes developed for navigation for the dual arm robot including visual docking modules. It describes the software developed for the prototypes for Standard SLAM Based navigation, 3D navigation, Static Docking and Mobile Docking.

Table of Contents

1. LIST OF FIGURES

2. LIST OF TABLES

3. EXECUTIVE SUMMARY

The content of this document is the description of the Initial prototype of the navigation software for the dual arm robot including the visual docking modules. The main purpose of this document is to:

- Describe the status of the current navigation system of THOMAS
- Describe the algorithms selected to be implemented in both cell to cell and in-cell navigation of THOMAS

The main technologies/practises identified include:

- Laser based navigation.
- Low level wheel control on an Omnidirectional Mobile Platform of type Swerve Drive.
- Dynamic robot foot-print calculation.
- 3D localization.
- 3D obstacle detection.
- 3D Semantic Mapping.
- Visual servoing for static and mobile docking.

Further information is provided in the following:

- Section 5.1 provides a description of cell to cell navigation, addressing standard slam navigation and 3D perception-based localization and obstacle detection.
- Section 5.2 presents the approach adopted for in-cell navigation, addressing static docking and mobile docking.
- Section 5.3 explains the main tests done for the navigation system during the project THOMAS first review meeting and the BIEMH 18 fair.

Primary conclusions/results include the following:

- Cell to cell navigation system is in a good shape with demonstrated robustness tests, such as the BIEMH18 fair. The integration of 3D perception is expected to bring more robustness and reliability to the navigation system.
- In cell navigation, proved to be robust for static docking. The big challenge in this area is the mobile docking. Achieving a reasonably robust mobile docking is envisioned to be possible, but it is depending on the nature of operation. If the MRP should perform actions synchronized with the MPP, the docking tolerance needed might be difficult to achieve.

4. INTRODUCTION

This document explains the initial prototype of the navigation software for THOMAS dual arm robot including the visual docking modules.

Section 5.1 explains the state of the initial prototype for cell to cell navigation. Laser based navigation and localization has been implemented. Besides the navigation and localization methods implemented, some other actions were necessary to improve navigation efficiency and safety. Some of these actions were the improvement of the wheel low level management and the use of dynamically adaptable robot's footprint to have into account the robot's arm configuration while navigating. This is explained in detail in section 5.1.1.

Even with the previous improvements, laser based navigation algorithm has its limitations. The main limitations are, robot re-localization problems (robots must start navigation actions always at a known point in their map) and the limited 2D information about obstacles received from lasers. To address these problems, 3D perception is being integrated to THOMAS navigation algorithm. In addition, the viability of using 3D semantic maps will be studied in next months of the project. Section 5.1.2 explains in details the 3D navigation approach being followed at THOMAS.

Section 5.2 explains the state of the initial prototype for in-cell navigation. This section focuses in the two types of docking mechanisms being developed in the project. Section 5.2.1 explains the first prototype developed in THOMAS project for making the MRP able to dock into a static plug, for example for power or air supply. Section 5.2.2 Mobile Docking extends the static docking paradigm to make the MRP able to follow a mobile object and perform an operation synchronized with the mobile object. This kind of navigation will be demonstrated at the PSA use case of the project.

Section 5.3 presents the two main demonstration points for the navigation system during the initial period of the project. They have been presented during the first THOMAS review meeting and the BIEMH18 fair where THOMAS Mobile Robot Platform was presented for 5 days in continuous operation of 10 hours per day.

5. PERCEPTION ENABLED NAVIGATION AND DOCKING FOR MOBILE ROBOTS: INITIAL PROTOTYPE

5.1. CELL TO CELL NAVIGATION

5.1.1. STANDARD SLAM NAVIGATION READY FOR THE MRP

As detailed in the design, the navigation development plan stablished a common base for the MRP's cell-to-cell and in-cell navigation. Both navigation steps use as base navigation standard laser based techniques that will be detailed below.

Besides the navigation and localization methods implemented, some other actions were required in order to improve navigation efficiency and safety. Some of them were the improvement of the wheel low level management and the use of dynamically adaptable robot's footprint to have into consideration robot's arm configuration while navigating.

Except from the dynamic footprint, all described developments where implemented and successfully tested in real-life demos, such as AERNNOVA plant during project's first review meeting and in TECNALIA's stand in the *BIEnal de la Maquina Herramienta 2018* (BIEMH18) fair. The dynamic footprint has been tested in simulation and it is expected to be integrated to THOMAS navigation in the following period.

1. **Laser based navigation**

The navigation of the first prototype of the MRP is composed of standard 2D laser-based navigation. Several implementations available as packages of ROS have been configured, tested and fine-tuned for the MRP. This 2D navigation is used as base system for global and local navigation.

Standard laser-based navigation is typically composed of a two-step approach:

- In a first learning step, a SLAM (*Self Localization And Mapping*) is used to generate a 2D occupancy map. In the THOMAS project, the implementation tested and used is the SLAM approach from [10], available in ROS as the package *hector_mapping*. The main advantage of this approach is that it offers great robustness without depending on additional ego-motion estimation sources (e.g. odometry). This algorithm had to be tuned in case of the MRP in order to cope with platform's dynamics and laser scanners' refresh rate.
- In the second step, the existing map used for navigation. This navigation is also composed of two levels:
    - Localization: This module provides a position based on the already recorded map. The localization algorithm used in THOMAS is the well-known *Augmented Monte-Carlo Localization* (AMCL) from [9], available as the popular *AMCL* ROS package[1]. AMCL is considered as the *de-facto* standard for laser based localization and it is widely used in many applications.
    - Planning: This module is for computing safe trajectories for robot's navigation and send direct speed commands to the drive controller allowing the robot to follow one trajectory. This planning is also made at two levels:
        - Global planner: This module is responsible to compute a complete path from the starting to the goal point. This computation is purely geometric, based on the well-known Dijkstra and A* algorithms. ROS provides implementations for both algorithms in the packages *global_planner*[2] and *navfn*[3], available as plugins of the ROS navigation stack. Both implementations have been tested and *global_planner* was preferred.
        - Local planner: This module is responsible for (1) compute actual speed command to follow the planned path and (2) adapt the previous path to avoid

[1] http://wiki.ros.org/amcl
[2] http://wiki.ros.org/global_planner
[3] http://wiki.ros.org/navfn

obstacles in collision course, not detected when the original path computed. There are several packages available in ROS that implement different algorithms and approaches for this task. Three of them have been tested with the MRP, presenting different outcomes:

- *dwa_local_planner*: This ROS package[4] implements the classic *Dynamic Window Approach* (DWA) from [8]. It is a robust and well proven algorithm. However, the generated trajectories for an omnidirectional robot are somehow very unintuitive for a non-knowing human observer. Since the indented use of the MRP is in collaboration or close by with humans, this approach was abandoned in favor of more modern approaches with better behavior adjustment.
- *eband_local_planner*: This module[5] uses an elastic band approach, generating bands (curves) that link consecutive points in the path, as described in [6]. While obtaining good paths, its current implementation is focused on differential drive robots. However, the generated trajectories do not take advantage of the omnidirectional capabilities of the platform.
- *teb_local_planner*: This package[6] implements the Timed Elastic Band approach [7], which tries to optimize the original trajectory with bands that minimize the trajectory execution time, separation from obstacles and compliance with the Kinodynamic constraints. Compared to the other two choices, *teb_local_planner* is the newest approach and allows greater configuration and fine tuning of the final behavior. It implements support for omnidirectional robots in ROS Kinetic version (later backported to ROS Indigo version). A great effort has been put to tune the planner for the MRP and it has been the planner of choice for the general assembly and BIEMH18 demos. While its current state allows for safe and robust navigation (as it was showed during the BIEMH18), it still presents some issues in very cluttered environments, in which proximity from obstacles and noise in the sensors cause instability in the generated optimal paths. Combined with the slow reaction to orientation changes of the MRP due its swerve drive, this algorithm results require a big amount of time.

2. **Low level wheel management**

The MRP is a four wheeled, omnidirectional platform in the configuration usually referred as “Swerve Drive”. The Swerve Drive is composed by several (usually four) wheels that can be controlled both in orientation and speed. Each configuration of wheels with their orientation and speed provides a specific linear and angular speed to the center of the platform (Figure 1), providing three degrees of freedom.

[4] http://wiki.ros.org/dwa_local_planner
[5] http://wiki.ros.org/eband_local_planner
[6] http://wiki.ros.org/teb_local_planner

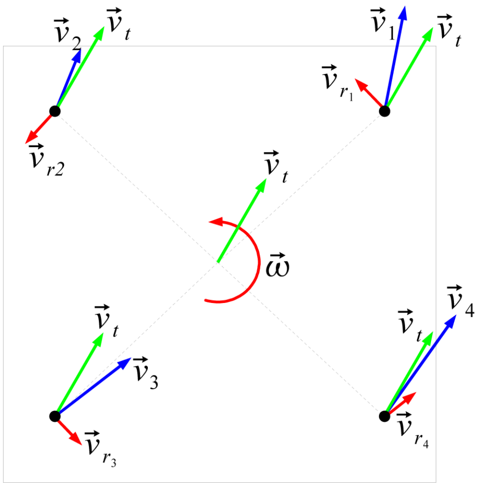

Figure 1: Relationship between individual wheels and common wheels in Swerve Drive configuration

Swerve Drive is frequent in omnidirectional platforms since it has several important advantages, like that it uses simple wheels (that provide better stability) and has greater pushing force (since all wheels provide traction).

However, it also has some important drawbacks, coming mainly from the complexity of its control scheme and build. For instance, Swerve Drive control strategy requires very precise wheel synchronization to guarantee constant coherent wheel configuration. Otherwise, the platforms must either stop or drag one or more of the wheels, increasing the stress and wear they suffer.

In the case of THOMAS project, smooth platform movement is a requirement since it combines platform movement with manipulation (e.g. dumper screwing in the PSA use case). The control provided out of the box with the platform was found to have sub-optimal performance, causing that the MRP must constantly stop its movement to reconfigure the wheels. Several modifications to the control must be performed:

- Reimplementation of the angle and speed computation for each wheel, following the swerve drive formulas. Given a platform's speed command $\vec{v}_t$, $\vec{\omega}$; each wheel's speed is computed as:

$$\vec{v} = \vec{v}_t + \vec{\omega} \times \vec{r}$$

The direction of this speed vector provides wheel's orientation, while its magnitude provides its speed.

- Trace and fix several bugs that caused unnecessary wheel reconfigurations.
- Improved reconfiguration and limit check strategy, giving priority to constant movement against permanent wheel coherence. Limit checks where relaxed depending on current speed, allowing for incoherent states in wheels of up to 15º of orientation, depending on current speed. Criteria for inverting speed when opposite angle was closer was also updated.
- Fine tuning of the driver parameters, adapting them to the specifics of the MRP and its whole control system.

3. Dynamic robot footprint

To safely travel the environment, the robot not only needs to know the available space and the position of the obstacles surrounding it. It also needs to be aware of what amount of space itself occupies to know in which positions it is able stand without colliding with any other object in the environment. Traditionally, this space that the robot occupies is known as the robot's "footprint".

In 2D navigation approaches, the footprint is estimated as the projection (the plant) of the robot in the map's plane. A conventional robot usually has a fixed footprint and current available approaches do not consider the possibility of a robot with a changing form.

This is not the case in a mobile manipulator. This kind of robots are equipped with robotic arms that can project over the base footprint and whose configuration changes over time. Since the navigation system is not aware of this, it creates a situation with high collision risk (e.g. the navigation system tries to go through a door which the base can traverse, but the arms not).

Traditionally this problem is coped by defining a safe travel arm configuration in which arms do not project over the mobile base limits. Every time that the robot needs to move, the arms must be put in the travel configuration. This notably increases cycle times and prevents any simultaneous navigation and manipulation.

To avoid that, in THOMAS project a new module for dynamic footprint adaptation has been developed. This module has three functions:

- Monitor the arm's joints to obtain their position with respect to the robot's base.
- Compute the bounding polygon of the most external joints and base borders.
- Replace the current footprint with the computed bounding polygon.

Additionally, the standard ROS planner used for navigation has been modified to work with changing footprints, instead of using a fixed one from the start of the execution.

An example of the use of this module can be seen in Figure 2, where in the left there is the "standard" footprint covering the mobile base, while in the right the footprint has been extended to cover also the stretched arms.

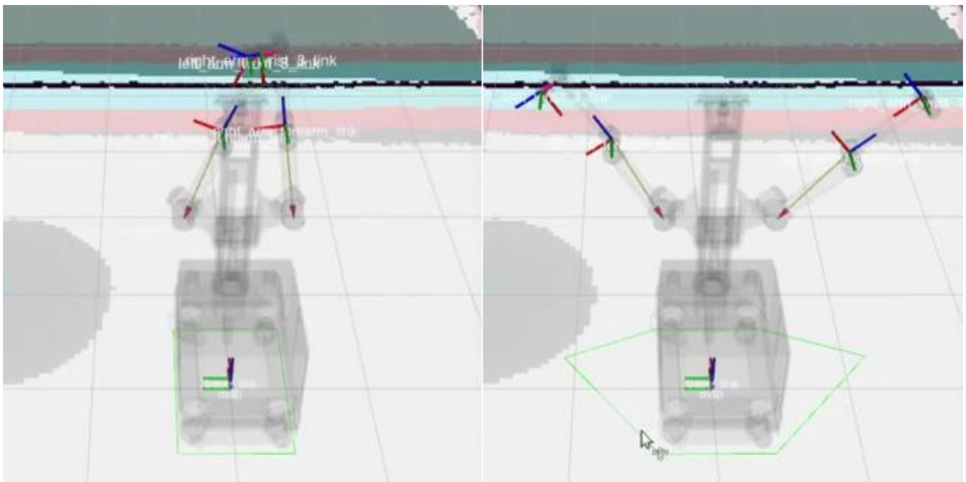

Figure 2: (Left) Default footprint covering the MRP's base. (Right) Updated footprint covering the stretched arms.

5.1.2. 3D PERCEPTION BASED NAVIGATION

Most commonly used navigation systems relay on laser sensors (e.g. lidar) to calculate the distance from obstacles. The array of distances obtained from such sensors can be used for several tasks that are cornerstones in a navigation pipeline:

- Mapping: the mobile robot will construct a representation of the environment (i.e. a map) that will be used as a model in which the navigation will be planned.
- Localization: in order to plan and execute the planned paths, the mobile robot must know and update at a relatively high frequency its position in the map previously built.

- Obstacle detection: while executing the planned paths, the sensors are used to detect obstacles that were not previously inserted in the map (e.g. dynamic obstacles) and use that information in an obstacle avoidance behavior.

As opposed to the classical approach that uses laser sensors as the main input to tackle the tasks avoid, Visual SLAM algorithms use vision sensors to solve the same problems. There are some advantages in using vision sensors in relation to solve the same task with laser sensors:

- Sensor consumption/cost: vision sensors are typically cheaper than lidar sensors, and consume less power, which has an impact on the autonomy of battery-powered mobile robots.
- Re-localization: this relates to both, the problem of initialization and the problem of recovering the robot's localization once the tracking was lost. Vision sensors provide richer information and allow to solve this problem in a more efficient way.
- 3D obstacles: lidar sensors typically provide distance readings in a scanning plane (there are exceptions that provide scanning movements in two axis). This can potentially lead to collisions in case some obstacles cannot be correctly detected.
- Map information: the amount and quality of information provided by a sensor system is much richer, which can be used to produce richer maps.

However, it is important to note that we do not propose to substitute laser sensors with vision. Laser provide very robust readings and allow for safety certification to be achieved, while vision sensors are much harder to certify. We propose to combine both sensor modalities, as it is a good way of improving a system's robustness and resilience to have redundant information coming from different sources with different parameters of performance and failure modes.

To perform the initial tests, we have reviewed the state of the art of Visual SLAM methods, focusing on those with open implementations provided, in order to minimize the integration time required. In the Table 1 below, we show the different algorithms analyzed, along with some parameters that help characterizing their suitability.

Table 1: Visual SLAM algorithms

Package	**Sensor**	**Map type**	**Relocalization**	**CPU/GPU**	**ROS**	**Notes**
FAB-MAP	2D Camera	Appearance-based navigation	BoW	CPU	Fuerte	Useful for relocalization
ORB-SLAM	2D Camera	Sparse	DBoW2	CPU	Yes	Robust
ORB-SLAM2	2D Camera RGB-D	Sparse Densified over keyframes No fusion	DBoW2	CPU	Yes	*No Comments*
LSD-SLAM	2D Camera	Semi-dense	openFabMap	CPU	Yes	*No Comments*

Package	Sensor	Map type	Relocalization	CPU/GPU	ROS	Notes
RGBDSLAMv2	RGB-D	Dense	No	GPU Only for SiftGPU	Yes	Very good integration in ROS
Kintinuous	RGB-D	Dense	DBoW	GPU	No	Issues with locally loopy trajectories
ElasticFusion	RGB-D	Dense	Random Ferns	GPU	No	Limitations on the map size
RTAB-Map	RGB-D	Dense (Densified , no fusion?)	BoW	CPU	Yes	Very good integration in ROS
DPPTAM Dense Piecewise Planar Tracking and Mapping	2D Camera	Dense Assume piecewise planar	No	CPU	Yes	Looks good for reconstruction, not so much for navigation
BundleFusion	RGB-D	Dense	Yes (global alg.)	GPU	No	Evolution of ElasticFusion Visual Studio project
SVO	2D Camera	Sparse	No	CPU	Wrapper	Fast, embedded systems
DSO	2D Camera	Semi-dense	No	CPU	Wrapper	Odometry
OKVIS	2D Camera + IMU	Sparse	No	CPU	Wrapper	Focus on IMU integration

Package	Sensor	Map type	Relocalization	CPU/GPU	ROS	Notes
ROVIO Robust Visual Inertial Odometry	2D Camera + IMU	Sparse	No	CPU	Yes	Focus on IMU integration
maplab / ROVIOLI Visual-inertial mapping framework	2D Camera + IMU	Sparse/Dense	Yes	CPU	Yes	Generic framework
DVO Dense Visual Odometry and SLAM	RGB-D	Dense	No	CPU	Wrapper Fuerte	*No Comments*
MRSMap	RGB-D	Dense Surface elements	No	CPU	Wrapper	Looks good for reconstruction, not so much for navigation
RGBDTAM	RGB-D	Dense	DBoW2	CPU	Yes	*No comments*

Out of the list, the RGBDSLAMv2 algorithm was selected for the initial tests, as it provides very good integration with ROS, and has a clean code structure that can be modified.

During the setup of RGBDSLAMv2, many conflicts with the current versions of the dependencies were presented. In order to avoid these conflicts, another algorithm was selected. This algorithm was the Docker-based solution. Docker provide isolation from the dependencies in the system, while avoiding any performance penalty as happens in the case of virtual machines.

Out of the initial tests performed, Figure 3 shows the initialization of the system. The top image shows the created map, which is represented as a pointcloud. The library offers functions to export the pointcloud to an octomap, which can be used also by other ROS packages, for example, to perform a path planning action. The bottom images present the output from the sensor, with the features located in the 2D image and used to calculate the camera tracking.

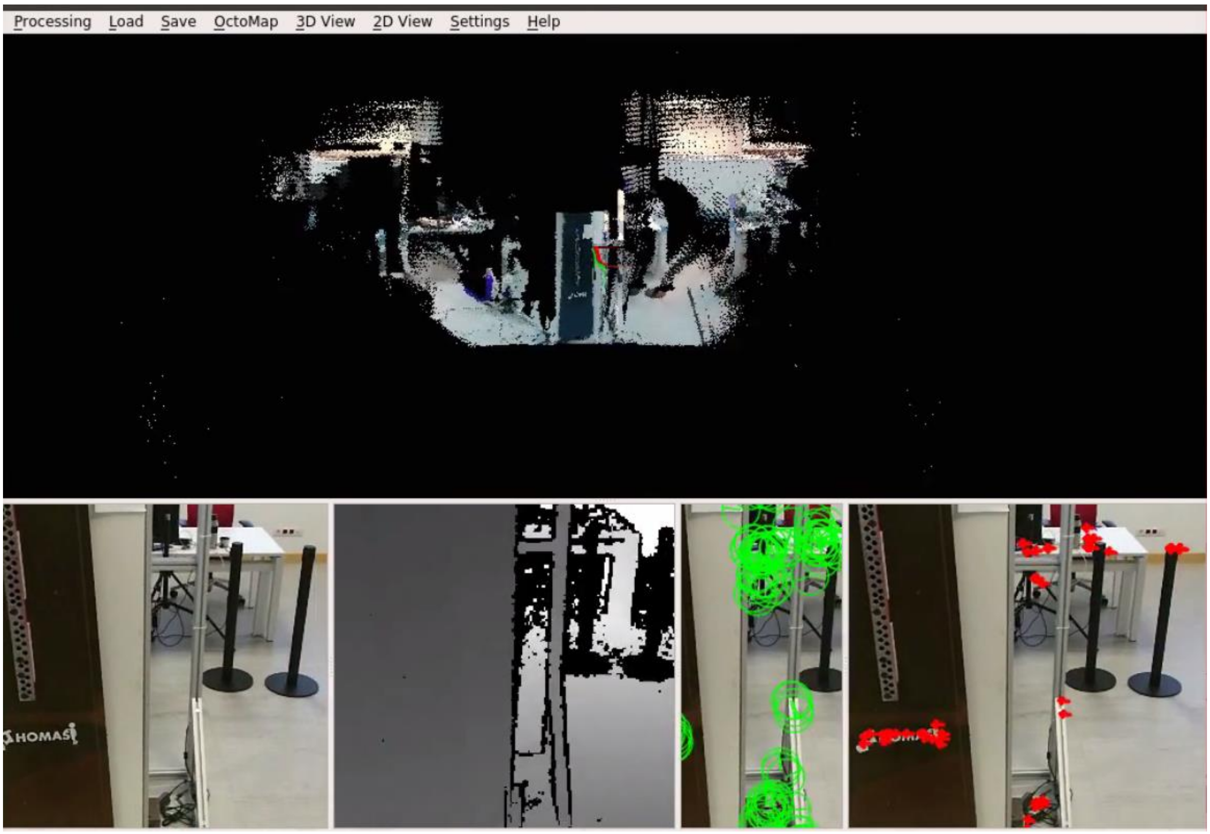

Figure 3: vSLAM initialization view

Figure 4 shows the map produced after a run in the test environment, with the camera/robot trajectory.

Figure 4: Final 3D map of the environment and robot trajectory

5.2. IN-CELL NAVIGATION

5.2.1. STATIC DOCKING: ACCURATE POSSITIONING WITH RESPECT TO A STATIC REFERENCE (TECNALIA)

To perform the first technology validation tests, a Microsoft Kinect camera and a large size marker were used since the resolution of the camera was not good enough for using a smaller marker. The Kinect is installed in the torso of the robot at 1.85m from the ground, which is why the movements of the platform and arms affect the stability of the image due to the propagation of vibration by the torso. The Kinect is located in a motorized pan tilt system that gives the camera a large field of view. However, if the pan tilt system together with the camera are not properly calibrated they can produce noise and imprecations in the measures of the docking system produced when moving the tilt pan.

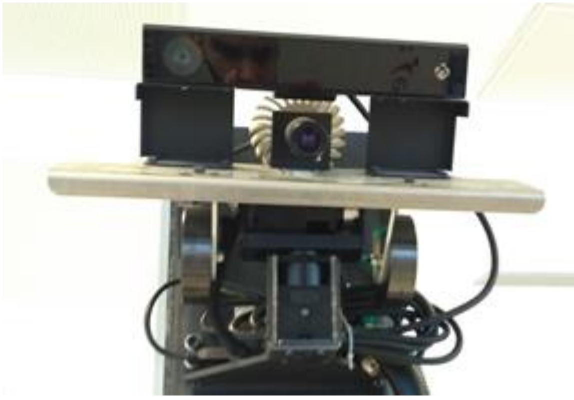

Figure 5: PanTilt system with the Kinect camera

In spite of this, and in order to validate the concept, a system of visual servoing of the robot was developed with respect to the marker taken as a reference. This system is based on a proportional control that maintains and ensures the position of the robot with respect to the marker. To obtain a good response of fluid movements of the robot it has been decided to use a closed loop system with a sampling frequency of 20 Hz. A frame transformation is performed to validate the position and orientation from the image of the marker seen from the camera to the base of the robot. The position error between both is translated, within a closed loop, as setpoint speed that is sent directly to the robot's traction system. The system accepts various configuration parameters to achieve precise docking. The reference parameter establishes the relationship between the marker and the final position of the robot. A pre-calibration process is necessary to establish that position. Tolerance is the margin of precision with which the robot is intended to reach the goal. The dock variable allows whether the robot is performing the dock or undock.

After validating the pilot test with the described system, a more robust method is sought to perform the static docking of the robot. For this reason, an IDS camera has is installed on robot’s base. This gives the system a higher image quality that translates into precision. Being in the bottom of the robot do not feel so many vibrations. All these improvements facilitate the installation of a smaller marker and therefore less invasive with the working area.

Figure 6: Docking camera installed on MRP's base

Due to the need to have a large flow of compressed air for the activation of the pneumatic drill, a docking station (images) has been manufactured and the marker is attached there in order achieve the static docking using the charging station. [Video of the process]

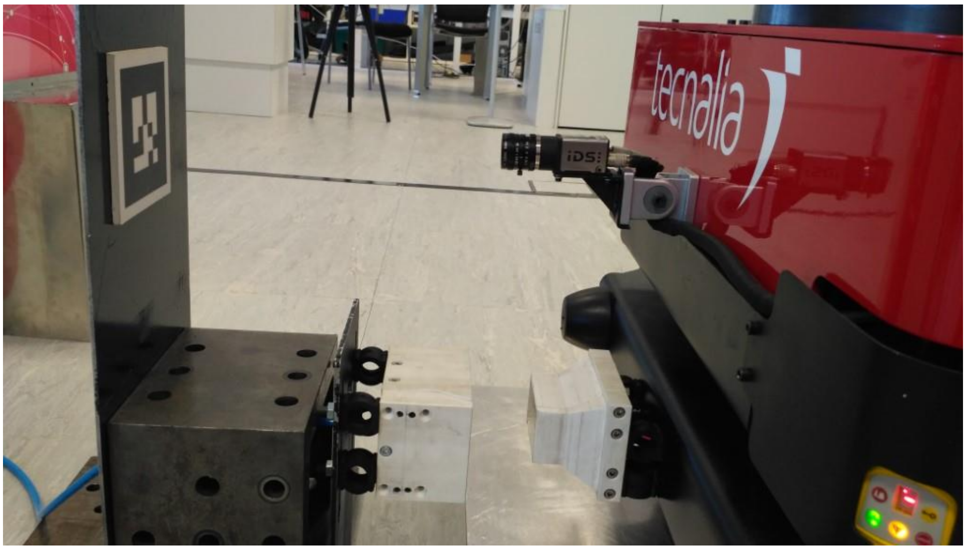

Figure 7: Final docking system with charge station and marker installed

5.2.2. MOBILE DOCKING: MOBILE REFERENCE ACCURATE POSSITIONING AND TRAJECTORY FOLLOWING (TECNALIA)

To carry out the dynamic docking process, the actual literature (SoA) has been used to know more about the current state and possible solutions to this problem.

Many articles about objects' detection and tracking by laser have been found. Mainly in automotive fields for the recognition of people and other vehicles in outdoor conditions. [3]

Looking in the literature, solutions that combine SLAM techniques with detection and tracking of moving objects (DTMO) [4] has been found. This approach is useful for the project because it is

intended to locate the AGV using the lasers installed on the robot's base. In Figure 8, it is possible to see the readings of the lasers where the contour / silhouette of the MPP is represented with a sequence of red dots that are laser beams bouncing against it. It is possible to observe the actual position of the MPP by means of a capture made from the Kinect camera located in the torso. The correspondence between the position of the real robot and the reading of the lasers is very precise.

The aim is to look for the characteristic points in a known shape that is generated by "the shadow" of the AGV [5].

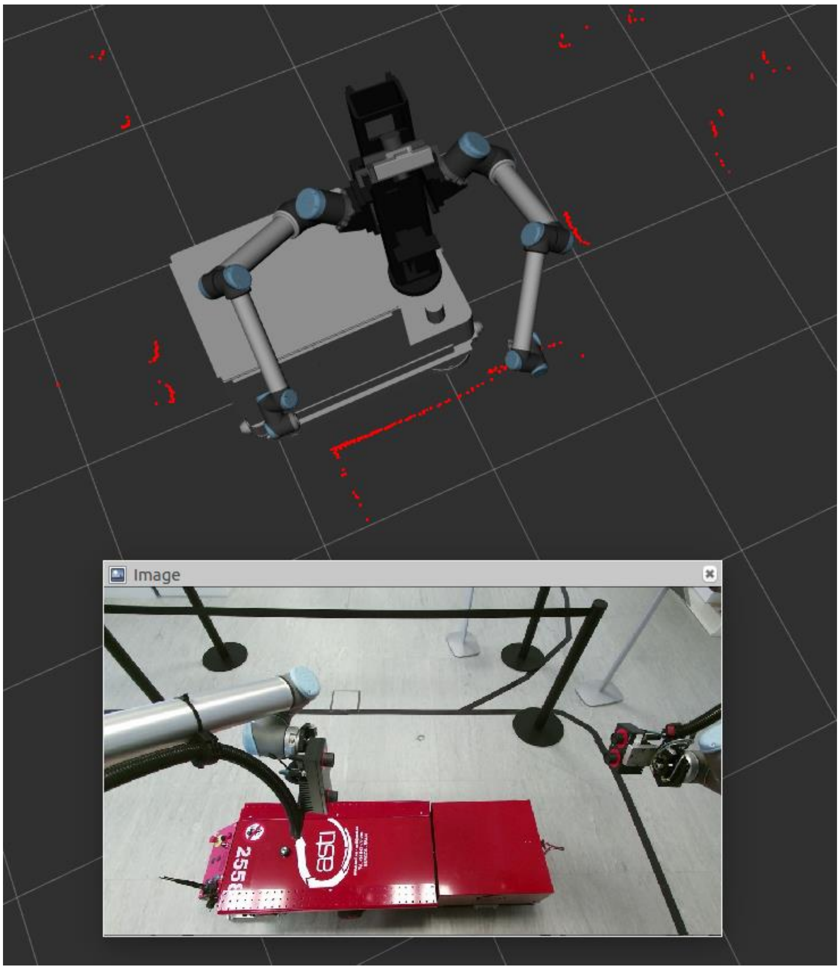

Figure 8: Silhouette of the MPP read from the robot lasers

In a first validation phase and taking advantage of the work done in the static docking, it is intended to update the code developed to adapt it to the dynamics and, therefore, be able to follow a moving marker, assuring the position and orientation with defined tolerances.

It is necessary to have a simulation very close to reality in order to perform these tests. That is why we have been working on a simulated environment in GAZEBO software using real robot's controllers and data from sensors. This is due to the danger of performing tests directly on the real system. The process of migration of the robot from the version of ROS Indigo to ROS Kinetic has involved an extra effort of what was considered at the beginning of the project and the simulation tests have been delayed.

5.3. PROTOTYPE DEMONSTRATIONS

5.3.1. REVIEW MEETING

From 12 to 13 of April 2018, the first review meeting of the THOMAS project held in TECNALIA and AERNNOVA premises. Several live demos where performed, including demos of the performance of the first prototype of the navigation and static docking systems, running in the actual TECNALIA's MRP.

In the demo at TECNALIA premises, a simulation of a drilling operation was successfully completed[7]. Besides the demos of the template detection and drilling, the process also included a simulation of cell-to-cell navigation using standard 2D Navigation and of in-cell navigation, using static docking with markers and visual servoing.

Figure 9: MRP during navigation, docking and drilling demo at TECNALIA's premises on 13th April of 2018

5.3.2. BIEMH18

The Spanish Biennial of Machine-Tool (BIEMH) is the third most important industrial fair in Europe and the first of its sector in Spain. This event holds every two years at the Bilbao Exhibition Center (BEC) in Baracaldo, in the months of May and June. It is aimed at the main manufacturers, importers and distributors of both national and foreign machinery to show their products on the site and reach commercial agreements with the more 35,000 buyers from the main industrialized countries of the world who visit this event in each edition.

TECNALIA has attended the fair of 2018 presenting a simulation of the drilling process[8] as part of aeronautics use case. This process was agreed with autonomous navigation between two parts of the cell demonstrating therefore this capacity that is also intended to validate in the THOMAS project. The ability to do static docking of the platform was also dominated when changing from one work area to

[7] Simulated drilling process of THOMAS Aeronautics pilot case during THOMAS 1st review meeting https://www.youtube.com/watch?v=56BcHbrruVQ&t=55s
[8] Drilling process of THOMAS Aeronautic use case in BIEMH18: www.youtube.com/watch?v=DXuNugUu0yI

the other. Both demos had a great acceptance among the visitors and the work done on the project was highly appreciated.

Figure 10: TECNALIA´S Stand at BIEMH18 where THOMAS AERNNOVA use-case mock-up presented

6. CONCLUSIONS

There is a running version of the cell to cell navigation algorithm with a high level or repeatability. Cell to cell navigation has been presented in both THOMAS 1st review meeting and BIEMH18 fair with positive feedbacks. However, this is expected to be even more reliable after the insertion of 3D perception system in MRP's navigation algorithm.

Despite the fact that in cell navigation proved to be robust for static docking, mobile docking seems to require a lot of effort. The docking tolerance in case of synchronized motion of MRP with the MPP is difficult to be achieved. This should be compensated with extra aids added to the operation to mitigate the tolerance effect.

7. GLOSSARY

MRP	MOBILE ROBOT PLATFORM
MPP	MOBILE PRODUCT PLATFORM
BIEMH	Bienal de Maquina Herramienta

8. REFERENCES

1. Chryssolouris, G., Manufacturing Systems: Theory and Practice, 2nd Edition, Springer-Verlag, New York, New York, (2006)

2. G. Michalos, S. Makris, N. Papakostas, D. Mourtzis, G. Chryssolouris,"Automotive assembly technologies review: challenges and outlook for a flexible and adaptive approach", CIRP Journal of Manufacturing Science and Technology, Volume 2, Issue 2, pp. 81-91 (2010)

3. MacLachlan, R., & Mertz, C. (2006, September). Tracking of moving objects from a moving vehicle using a scanning laser rangefinder. In Intelligent Transportation Systems Conference, 2006. ITSC'06. IEEE (pp. 301-306). IEEE.

4. Wang, C. C., & Thorpe, C. (2002). Simultaneous localization and mapping with detection and tracking of moving objects. In Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on (Vol. 3, pp. 2918-2924). IEEE

5. Fayad, F., & Cherfaoui, V. (2007, June). Tracking objects using a laser scanner in driving situation based on modeling target shape. In Intelligent Vehicles Symposium, 2007 IEEE (pp. 44-49). IEEE

6. Quinlan, S. and Khatib, O. "Elastic Bands: Connecting Path Planning and Robot Control." Proc. IEEE International Conference on Robotics and Automation, Atlanta, Georgia 1993, vol 2. pp. 802-807

7. C. Rösmann, W. Feiten, T. Wösch, F. Hoffmann and T. Bertram: Trajectory modification considering dynamic constraints of autonomous robots. Proc. 7th German Conference on Robotics, Germany, Munich, 2012, pp 74–79.

8. Fox, D., Burgard, W., & Thrun, S. (1997). The dynamic window approach to collision avoidance. IEEE Robotics & Automation Magazine, 4(1), 23-33.

9. Thrun, S., Burgard, W., & Fox, D. (2005). Probabilistic robotics. MIT press.

10. S. Kohlbrecher and J. Meyer and O. von Stryk and U. Klingauf, A Flexible and Scalable SLAM System with Full 3D Motion Estimation, Proc. IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR), November 2011