An Autonomous Mobile Robot Navigation Architecture for Dynamic Intralogistics

David Taranta[1,2], Francisco Marques[1,2], André Lourenço[1,2], Pedro Alexandre Prates[1], Alexandre Souto[1,2], Eduardo Pinto[1,2], José Barata[1,2]
[1]CTS-UNINOVA Instituto Desenvolvimento de Novas Tecnologias, 2829-516 Caparica, Portugal
[2]School of Science and Technology, NOVA University of Lisbon, 2829-516 Caparica, Portugal
Email: d.taranta@campus.fct.unl.pt, fam@uninova.pt, andre.lourenco@uninova.pt, pap@uninova.pt, a.souto@campus.fct.unl.pt, emp@uninova.pt, jab@uninova.pt

***Abstract*—This paper presents a platform-agnostic distributed navigation architecture for autonomous mobile robots operating in intra-factory logistics. Communication, control, navigation, diagnosis and hardware are layered in a hierarchical approach increasing robustness, modularity and flexibility. This architecture promotes several key features, such as dynamic selection of navigation profiles, semantic mapping and human-aware navigation. The approach allowed multiple autonomous mobile robots, cooperating through a fleet management system, to adapt to a wide range of situations, alternating their path planning between high-speed free-space strategies, and high precision low-speed for tight passageways and docking to assembly stations. The benefits of the proposed architecture were validated through a set of experiments in a mockup shopfloor environment. During these tests 3 robots operated continuously for several hours, self-charging without any human intervention.**

Fig. 1. AMRs integrated in the ATLAS project during the experiments.

I. Introduction

To achieve a reconfigurable production system, the subsystems and their elements must be defined accordingly during design phases [1], [2]. Several methodologies describing production system design address specifically this issue [3], linking it to the level of automation and modularization. The literature describes technologies enabling reconfigurability [1], [2], layouts in the form of determination and classification of various configurations [4], and ways of considering future product types [5], [6].

In the current fast-paced manufacturing scenarios, the ability to react to the introduction of new products, changes to the shop-floor, and recovery from unforeseen technical issues is paramount. Using fixed or semi-reconfigurable logistics units increases significantly the cost and time of reconfiguration. Recently, the European Research Agenda has pushed for safe coexistence and collaboration between humans, cooperative robots, and smart Autonomous Mobile Robots (AMRs) as a driver to improve the agility and reconfigurability of the shop-floor process.

However, while in other areas human-aware navigation is growing a trend [7], current applications are still mostly focused on safety, and less on operator acceptance, comfort or collaboration. One approach in industrial settings is to represent workstations, human workers, and obstacles in different costmap layers and fuse them with obstacle data [8]. While this strategy can improve the overall comfort of the operators it does not account for the time constrained specificity of assembly environments. A way to mitigate this problem is to include a prediction of the operators' locomotion to execute efficient and safe motions as [9] proposes in automotive assembly.

Intralogistic environments are by nature predictable and structured, and the presented model takes advantage of it by employing a deliberative control architecture [10]. Specifically, a platform-agnostic distributed hierarchical solution to empower AMRs with all the skills needed for a seamless and effective integration into a working shop-floor, predominantly designed for human labour.

The most noteworthy feature of this architecture is its navigation module. The goal is the development of a highly adaptable system capable of working in dynamic environments populated by other AMRs and human workers. The system not only tries to improve workers' safety but also their comfort while working alongside the AMRs. This is achieved by designing different navigation modes that adapt, in real time, to workers' behaviour taking into account their position, movement, among other cues.

Such is the target scenario in the ATLAS project, pairing the UNINOVA research institute and INCM [11] to improve the efficiency, security and traceability of logistics on the Portuguese Electronic Passport (PEP) and Citizen Card (CC) production processes. The project uses multiple fully-autonomous mobile robots, designed with different capabilities, coupled with collaborative stations to prepare and transport the products inside the factory. Both isolated module-specific, and whole-process continuous tests have shown the ability of the model to scale up to 3 different robots (see Fig. 1) coping safely with all allocated tasks, traversing an environment mirroring the final INCM processes and spatial constraints.

II. Proposed Architecture

The architecture is designed to be platform-agnostic without establishing requirements or assumptions about the locomotion type, sensory payload or any other base feature. It is layered to separate the different functionalities the system integrates: External Control; External Perception; Communications; Control; Navigation; Diagnostics and Hardware. Each layer exchanges data with subsequent layers to achieve the desired behaviour. The current implementation exploits Robot Operating System [12] distributed architecture. However, its very basic navigation functionality is not suited for the current challenges of dynamic shopfloors. Several new components were developed providing the required adaptability, and scalability for a seamless integration with multiple robots. The following subsections will detail the core Control and Navigation and briefly overview the complementing layers.

A. External Control

This layer includes any system or platform of high-level control a factory may have installed to manage the AMRs. For instance, web-based dashboards to monitor robot systems and the environment, or an interface with a Manufacturing Execution System, decomposing objectives to transportation orders. With multiple AMRs, as the target application requires, a Fleet Manager system is added to evaluate coordination strategies between concurrent resources and space, and put in practice complex decision making on managing common skills among the robots.

B. External Perception

External Perception includes all external software modules that can provide information to improve the robot's behaviour. Externally obtained information resulting from human detection and tracking or obstacle mapping, even from other AMRs, is integrated by the Navigation's External People Perception and Map and Obstacle Sharing modules, respectively. Currently, the solution for human-tracking exploits the Marvelmind Robotics Starter Set IA-01-2D [13] wearable sensor. It uses an Inverse Architecture, where stationary beacons emit ultrasound and mobile beacons receive it to calculate the position in the map. The Navigation Layer uses the detected people poses to change navigation behaviour unlike the current implementations in ROS and ROS2 navigation stacks.

C. Communications

Internal and external components to the AMR exchange data through a single point of communication to reduce their complexity. Separating the core components from it mitigates possible critical errors when communication blackouts occur. The module was implemented through refactoring the multi-master Rosbridge server-client solution in [14] allowing all types of messages and services to be interchangeably used among multiple independent systems. Fleet Manager, other robots, stations and other devices may have their independent ROS cores and still communicate through pairing their Rosbridge Servers and Clients using UDP, TCP or Websocket protocols. However, only a subset of all ROS services and topics are advertised to the AMR's remaining internal layers to avoid overloading the remote link.

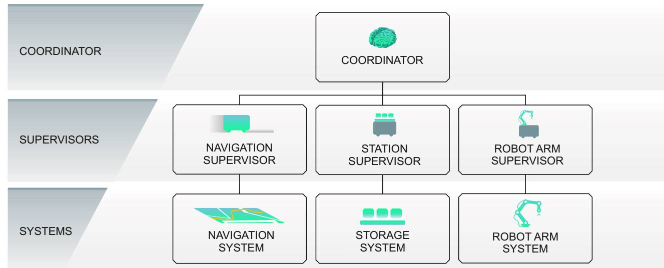

Fig. 2. Control Layer components example.

D. Control

The Control Layer contains the decision making and control logic necessary to implement all the required capabilities and skills. It provides a set of resources to the External Control Layer, enabling the system to have a higher level control awareness of the environment as well as of other robots. The components are semantically separated into Coordinator, Supervisor(s) and the System(s), as Fig. 2 shows, from top to bottom, respectively. The Coordinator is the high-level interface that exposes the skills, characteristics and state of the robot to External systems (within the External Layer). Whenever it receives a task, it breaks it down into function-atomic sub-tasks, and assigns them to the according Supervisors. It is also responsible for planning and controlling the state of execution, which may involve coordination between multiple Supervisors. In turn, the Supervisors manage all Systems to perform actions that amount to a said sub-task, each controlling locally their respective actuating and sensing devices.

The model proposes a single Coordinator for the system, but multiple Supervisors and Systems can be used, depending on the robot's capabilities. The hierarchy is also designed to promote different levels of execution monitoring, diagnosis and error recovery. Fig. 2 shows only the connections between modules omitting internal connections between their software blocks. Yet it portrays an example of a fully-fledged AMR that besides navigation also features an internal storage compartment and a robotic arm to handle materials.

The implemented system has the Coordinator relay and overview incoming tasks from the External Control to the Navigation and Station Supervisors. To control the flow of the received tasks, the Coordinator, makes use of a state machine set by default to a *Waiting* state, and switches to other states such as, *Charging*, *Navigating* or *Docked*, depending on the task currently being executed. Finally, the Coordinator also

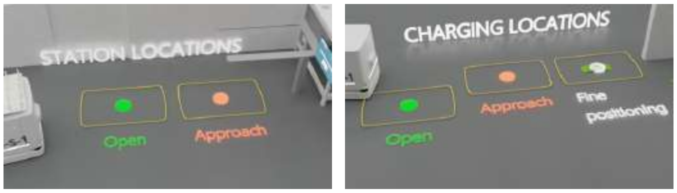

Fig. 3. Location Types and Navigation Profiles. On the left, the Profile will transition from *Free* to *Precise*, approaching the station from location *Open* to *Approach*. On the right, during the Charging approach, it will also transition from *Free* to *Precise*, and then shift to *Strict* crossing location *Approach* towards *Fine-Positioning*.

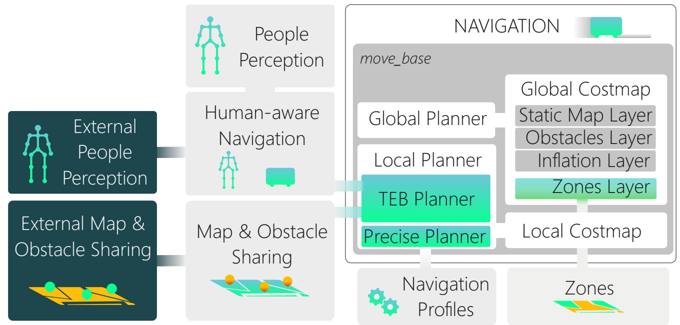

Fig. 4. Navigation Layer module set.

handles Charging tasks and exposes services to monitor the robot status or its storage content, and access and control Navigation parameters such as Get Plan, Set/Cancel Goal, Clear Costmaps or Set Zones (refer to II-E for more detail).

The Station Supervisor handles container storage, and their contents, in the robot's storage area. During any Loading or Unloading process, triggered by *Dock-Load/Unload-Undock* task sequence, the Station Supervisor allows the robot to share its resources with a shop-floor station. This way, the Station Supervisor's *Pick/Place* and *Set/Update* system operations are directly integrated into the external station's process to transfer the contents of the Storage System to and from the robot. The execution control of these interactions is implemented through a behaviour tree based on the BehaviorTree.ROS package [15].

The Navigation Supervisor manages all the System modules related to the robot's autonomous navigation capabilities. It interfaces the high-level ontology with the Navigation System data, and vice-versa. A goal location, for instance, relayed by the Coordinator, through a *Move* task or a *SetGoal* request, is translated to a map-referenced pose before it is sent to the Navigation System. It is then responsible for requesting and tracking the execution of navigation by the Navigation layer. This Supervisor includes a novel mechanism to load and change the navigation system's map at runtime, based on the ROS Navigation stack's *map_server* coupled with a set of locations that once crossed trigger the map change. Thus, the robot can have separate, smaller maps instantiated on a by-need basis and also support multi-floor environments. Another key feature is the ability to change Navigation Profiles based on the types of goal locations, currently varying between *Open*, *Approach* and *Fine-Positioning*. This feature is based on the ROS Dynamic Reconfigure component, but implemented in a way that enables the parameters to be changed in groups, depending on the situation. The implemented Supervisor triggers path planning and/or localisation methods, and their parameters accordingly: *Free* profile when both current and goal locations are *Open*; the *Precise* profile in between *Open* and *Approach* locations; and the fine-positioning *Strict* profile whenever a *Fine-Positioning* location is either start or goal.

Fig. 3 gives two specific applications. These profiles are explained further in II-E. Finally, besides relaying the Coordinator's Navigation parameterization services, it also features a Dynamic Reconfigure client module to update all those navigation configurations at runtime.

E. Navigation

The Navigation layer ensures that the AMR moves safely through the environment. It is within this layer that all the mechanisms and methods are included for the robot to harmoniously share the space with humans, employing zone and context-related queues into its navigation profile. The system also features the ability to integrate external queues in the form of maps, specific dynamic obstacle information, as well other devices informing the status of humans in the surrounding area. As depicted in Fig. 4, there is a main Navigation component, the centre box, that is directly responsible for the autonomous navigation. This main component represents the ROS Navigation Stack, which is enhanced by the modules proposed in this architecture such as Human-aware Navigation, Navigation Profiles, Zones, and Map and Obstacles Sharing modules. The actual architecture is not bound to these modules, as they are optional but they greatly increase the robustness of the whole system. They can be removed, replaced or further supplemented by new modules as other capabilities are required and made available.

The developed Zones component is responsible for restricting areas to the autonomous navigation core component. A polygon is marked on the map and assigned a restriction level depending on the situation. Fig. 5 depicts the implemented two levels of restriction: completely *Blocked*, with red boxes, and zones that should be *Avoided*, if possible, in yellow boxes. Each of the three situations depict a robot that wants to pass through a corridor that is split at the middle by a wall. As a default, when no restricted zones affect its path, the robot should take the shortest, least narrow option in its map. But as the figure shows, the zones with the least restriction level should be taken over the others. Though it is not a novel concept, this module allows a fast adaptation of the AMRs to a specific environment improving its overall performance.

This architecture can be used in a single robot system but can also benefit from the information gathered by the collective of a multi-robot system. Fig. 6 shows an example of the developed Obstacle Sharing module, having one robot detecting a blockage, and communicating it to its peers, may prevent them to waste time and energy attempting those

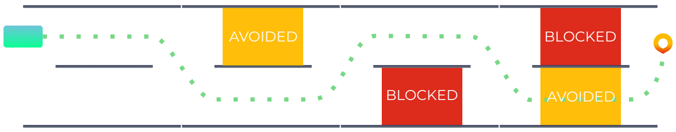

Fig. 5. Zones module behaviour. The AMR, blue box on the left, planning a path (dotted green line) to reach the goal, marker on the right, will prefer from left-to-right the shortest way, no-zone over *Avoided* or *Blocked*, and cross *Avoided* when no other option is possible.

temporarily obstructed passageways. Only individual obstacles are shared and are temporally added to the map so that if the obstruction is cleared in the meantime the passageway is attempted once again. Besides the obstacles, each robot shares its pose and current path. Hence, a robot's own planning can account for others' motion, avoiding deadlocks and time-wasting manoeuvres crossing each other in tight spaces.

The Navigation Profiles module is designed to adapt the navigation behaviour at runtime. For example, in a tight corridor the robot should navigate more carefully and slower than in an open space, or, given the nature of the obstacles, maintain a longer distance. Hence the system uses Profiles to switch the parameters and methods accordingly to a context input. This is done by implementing a ROS Dynamic Reconfigure Server in each node that needs its parameters changed at runtime, which will then be automatically changed by the Control Layer depending on the context.

With operators as integral parts of the shop-floor, the Human-aware Navigation module will adapt the robot's behaviour whenever it interacts or shares their work-space. It receives estimates of the human's position, both from internal and external People Perception modules (see Fig. 4) and changes the navigation to not disturb the ones in the robot's vicinity. The Human Aware Navigation is responsible for implementing the necessary behaviours that, for example, based on proxemics, keeps the robot at a comfortable distance from people, defining spaces around them. That distance is not fixed, but rather changes depending on the robot velocity, whether the person is moving or not, and others factors. Also, the robot should not exceed a maximum velocity near people so humans feel safe around it.

The main Navigation module is based upon Navigation Stack's *move_base* which has two main stages of operation: Global and Local planning (see Fig. 4). While the Local Costmap resorts to sensor data directly, the Global uses a compilation of layers, which features the novel Zones configuration proposed in this paper. The Navigation Profiles are set up to shift the Local Planner between the TEB [16], for the *Free* profile, and two settings for a Precise Local Planner, to increase motion precision for *Precise* and *Strict* profiles. This last planner relies on simple vector velocity commands to perform finer adjustments for docking procedures. Besides lower planner velocity, among other settings, the *Strict* profile also affects the Localisation type of the robot to aid the planner with more-precise pose estimations.

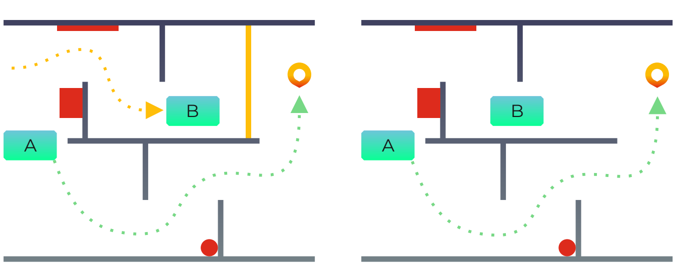

Fig. 6. Obstacles sharing module. Red represent obstacles not present in the static map. In orange, the obstacles detected by the other robot but not in line of sight.

In *Free* the Human-Aware Navigation and Shared Obstacles output translates operator's estimated positions and velocities from People Perception modules to TEB Dynamic Obstacle, with their weight based on proxemics theory.

The developed Obstacle Sharing module subscribes to other AMRs (and publishes its own) pose and planned paths and transforms them into TEB Obstacles as well. Zones are also translated to TEB Obstacles to integrate the pre-defined space restrictions in the planning step.

F. Diagnostics and Hardware

The Diagnostics Layer main objectives, are to make sure the system is fully working at startup and to inform the existence of problems at runtime. Hardware and software components report to it any issues that can be recovered or require human intervention.

The Hardware Layer represents the software modules, drivers and interfaces that interact directly with the hardware. The implemented solution is based on KUKA ROS stack nodes and a WebSocket interface with KUKA's proprietary Sunrise software. The data acquired by this layer is available for other components to use, such as the Navigation System from the Control Layer.

III. Experimental Setup - RICS Shop-floor

The goal of this experiment is to evaluate the developed system's behaviour emulating a secure working environment where human operators and mobile robots cooperate in the process of producing and dispatching PEPs and CCs (see Fig. 7). The entire process is divided into stations, some with collaborative arms responsible for executing tasks autonomously, while others require human operators. The mobile robots are responsible for securely transporting the products and miscellaneous parts between stations. A fleet management system sends all navigation goals to the robots while monitoring their progress. The environment also includes two automatic charging stations for the mobile robots with a floor contact system.

A. Autonomous Mobile Robots

The two mobile robot models used in this experiment are the holonomic KUKA KMR and KMP 200 (see Fig. 1). Both models have a length of 1.080 m, a width of 0.640 m and a height of 0.700 m without the robotic arm. Their

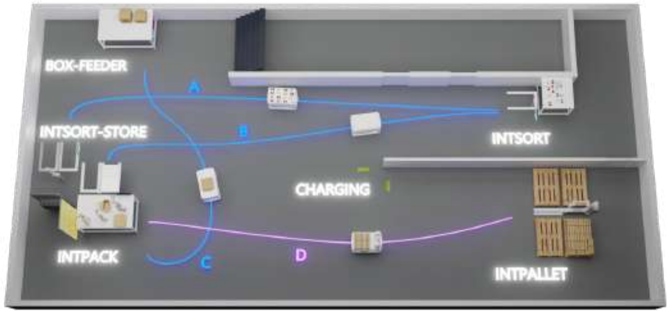

Fig. 7. Render of RICS-INCM LAB Shop-floor. In blue and purple are depicted the paths for the *INTMOB-S* and *INTMOB-L* robots respectively.

linear velocity is limited to 1.0 m/s while the maximum rotational speed is 0.56 rad/s. The robot's perception package is comprised of two 270 º field-of-view SICK SAFE S300 laser scanners, placed on opposing corners to 360 º field-of-view at a 15 cm height. An AMD Ryzen 7 3700X Octa-Core 3.6 GHz based processing unit, running Ubuntu 20.04 LTS and Robot Operating System [12], was added for the high-level navigation control. The low-level control is ensured by the two industrial computers running KUKA's proprietary software.

The KMR and KMP are equipped with autonomous charging for a battery capacity of 106 Ah allowing approximately 8 hours of continuous operation. The KMP 200, from now on referred to as *INTMOB-S*, is responsible for transporting simulated goods between various stations (see paths A, B, and C in Fig. 7). The KMR 200, or *INTMOB-L*, transports only cardboard boxes between two stations (see path D in Fig. 7). Three *INTMOB*s were available for the experimental testing, two *INTMOB-S* and one *INTMOB-L*.

B. Assembly Stations

The process simulated in the RICS shop-floor is composed by: two assembly stations *INTSORT* that produce batches of PEPs and CCs and *BOX-FEEDER* that provides empty flat cardboard boxes; one storaging and packaging station *INTPACK*; one palletizing station *INTPALLET*. Each INTMOB-S can transport products or raw materials between *INTSORT*, *BOX-FEEDER* and *INTPACK* while the only *INTMOB-L* connects the *INTPACK* station to the *INTPALLET*. With the normal production process starting with PEPs and CCs batches at *INTSORT* that are then packaged at the *INTPACK* station to then be palletized at *INTPALLET* (see Fig. 7).

IV. Experimental Results

This section presents the results obtained during the experiments where the proposed model was instantiated with the exact same configuration in three *INTMOB* mobile platforms. These field trials were designed to simulate a real intralogistics environment, with the mobile units accumulating more than 1100 meters without any issue (see Fig. 8).

Table I summarises the quantitative results obtained during the experiments. Performance was evaluated by recording possible safety events, measuring navigation robustness and

TABLE I
Inter Stations travel time with approaches

	Time (s) +- SD			
Path	A	B	C	D
Minimum	24	22	11	22
Maximum	36	33	13	40
Average	28.11±3.52	26.38±3.70	12.00±0.71	28.38±6.28
Average + 2 * Approach	56.49	54.76	40.38	56.76

accuracy and time taken in each route, and finally docking reliability. Overall, the proposed model showed the ability to navigate through the environment with all AMRs without any safety events or collisions. Navigation continuous performance and reliability is apparent by the small variance in the time travelled in each logistics route despite the encounters. Furthermore, the robots travelled safely with an average speed around 0.5 m/s in a very constricted space shared with human operators. However, the precision needed during each docking procedure (see Fig. 9a) increased significantly the overall time between stations. For instance, in the case of path C the total time more tripling the normal navigation time. In the paths A, B and D the time taken in the approach manoeuvre is not as significant as the paths themselves are longer. In a real shop-floor this will be also the tendency as the expected distance between stations will be greater than in this constricted scenario.

The experiments showed that the proposed system in this paper is apparently robust and reliable when performing intralogistics tasks. During simulated production the robots faced several times humans crossing their path and managed to avoid them correctly (see Fig. 9d). The mobile units also faced each other in constricted and narrow spaces but were always able to perform correctly without any deadlocks, bottlenecks, or significant delay. Furthermore, all station docking, undocking, and charging procedures were executed successfully. In sum, the system showed no major issues during operation when instantiated in a scale of a real production environment with

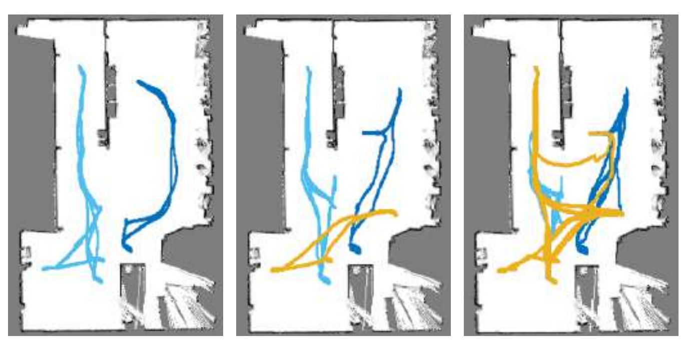

Fig. 8. LAB data from experimental tests. Light and dark blue lines represent the paths taken by the INTMOB-S-1 and INTMOB-L-1 respectively. Yellow lines represent the paths from the INTMOB-S-2.

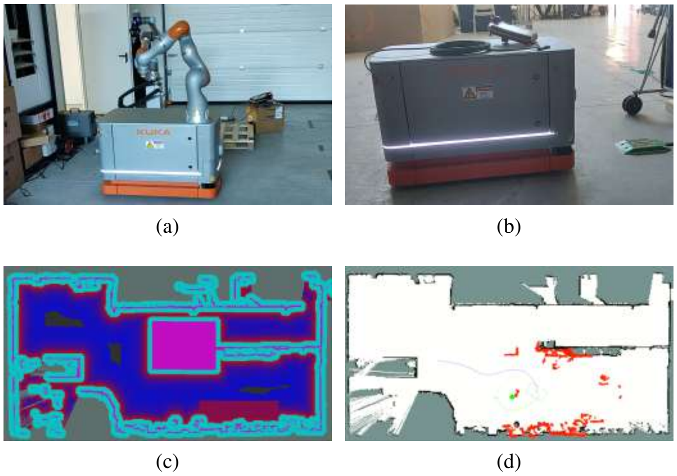

Fig. 9. Snapshots of the experiments. (a) *INTMOB-L* during docking to the *INTPALLET* station. (b) *INTMOB-S* using the precise navigation profile to dock into the automatic charging pads. (c) The global costmap of the RICS shop-floor. In pink and red are depicted *Blocked* and *Avoided* zones respectively. Other colours adjacent to pink are the result of inflation. (d) Using external perception for navigation.

three mobile units and several stations. The length, variability, and extension of the current tests indicate the robustness of the system. However, the system will undergo more extensive stress testing with longer production times to validate its performance in a twenty-four seven production environment.

V. Conclusion

A platform-agnostic distributed software architecture for autonomous mobile robots for intralogistics operations was presented, showing the ability to perform logistics assignments in a robust and safe way. The distributed architecture was developed using the ROS framework following an hierarchical layer approach to increase robustness, modularity, resilience, and flexibility.

The navigation layer is one of its major contributions even though based on the ROS Navigation Stack, it integrates other components developed and integrated adding, for instance, the ability to dynamically change the navigation behaviour based on semantic assigned zones, multi-floor support, and docking to stations. A planner capable of handling dynamic obstacles [16] was integrated, adapted, and showed to be able to perform human-aware motion. To increase the robustness of the docking process, a especially developed planner also showed to improve the overall behaviour. Besides the navigation, the control layer was also proven capable of executing tasks from a higher level manager.

The experimental results have confirmed the applicability and the added benefit of the presented multi-layered model. Particularly, portraying the navigation system's versatility as it switched between high-speed free-space path planning strategies, and high-precision, low-speed, manoeuvres facing tight passageways and docking to stations (charging and work stations). The tests were performed with three different robots in a real mockup environment based on INCM's production scenario. For several hours, the absence of collisions or any permanent blockages with obstacles either static, previously mapped and not, or dynamic, people and other AMRs, gives evidence of the proposed navigation system's ability to keep its safety requirements while optimising its trajectories while pursuing goals from a fleet manager.

As future work we also intend to integrate and enhance our previous work in modelling operators and stations [8] as new cost layers. Finally, we aim to improve the human-aware behaviour of the model by exploiting motion queues produced by the AMRs and by integrating task recognition and classification modules to the external perception layer.

Acknowledgment

This work was co-funded by INCM ATLAS project and by the CTS multi-annual funding, through the PIDDAC Program funds.

References

[1] Y. Koren, U. Heisel, F. Jovane, T. Moriwaki, G. Pritschow, G. Ulsoy, and H. Van Brussel, "Reconfigurable manufacturing systems," *CIRP annals*, vol. 48, no. 2, pp. 527–540, 1999.

[2] M. G. Mehrabi, A. G. Ulsoy, and Y. Koren, "Reconfigurable manufacturing systems and their enabling technologies," *International Journal of Manufacturing Technology and Management*, vol. 1, no. 1, pp. 114–131, 2000.

[3] C. Rösiö and K. Säfsten, "Reconfigurable production system design–theoretical and practical challenges," *Journal of Manufacturing Technology Management*, 2013.

[4] Y. Koren and M. Shpitalni, "Design of reconfigurable manufacturing systems," *Journal of manufacturing systems*, vol. 29, no. 4, pp. 130–141, 2010.

[5] M. R. Abdi and A. Labib, "Grouping and selecting products: the design key of reconfigurable manufacturing systems (rmss)," *International journal of production research*, vol. 42, no. 3, pp. 521–546, 2004.

[6] T. AlGeddawy and H. A. ElMaraghy, "Changeability effect on manufacturing systems design," in *Changeable and reconfigurable manufacturing systems*. Springer, 2009, pp. 267–283.

[7] T. Kruse, A. K. Pandey, R. Alami, and A. Kirsch, "Human-aware robot navigation: A survey," *Robotics and Autonomous Systems*, vol. 61, no. 12, pp. 1726–1743, 2013.

[8] F. Marques, D. Gonçalves, J. Barata, and P. Santana, "Human-aware navigation for autonomous mobile robots for intra-factory logistics," in *International Workshop on Symbiotic Interaction*. Springer, 2017, pp. 79–85.

[9] V. V. Unhelkar, P. A. Lasota, Q. Tyroller, R. Buhai, L. Marceau, B. Deml, and J. A. Shah, "Human-aware robotic assistant for collaborative assembly: Integrating human motion prediction with planning in time," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 2394–2401, 2018.

[10] P. Pirjanian, T. L. Huntsberger, A. Trebi-Ollennu, H. Aghazarian, H. Das, S. S. Joshi, and P. S. Schenker, "Campout: a control architecture for multirobot planetary outposts," in *Sensor Fusion and Decentralized Control in Robotic Systems III*, vol. 4196. International Society for Optics and Photonics, 2000, pp. 221–230.

[11] "Prémio inovação 2017 incm, atlas – autonomous transport for logistic automatic system," last accessed 12 December 2019. [Online]. Available: https://premioin3mais.pt/edicoes-anteriores/

[12] Robot operating system. [Online]. Available: https://www.ros.org/

[13] Marvelmind robotics. [Online]. Available: https://marvelmind.com

[14] A. Lourenço, F. Marques, R. Mendonça, E. Pinto, and J. Barata, "On the design of the robo-partner intra-factory logistics autonomous robot," in *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2016, pp. 002 647–002 652.

[15] D. Faconti, "Mood2be: Models and tools to design robotic behaviors," Eurecat Centre Tecnològic, Barcelona, Spain, Tech. Rep., 4 2019.

[16] C. Rösmann, F. Hoffmann, and T. Bertram, "Kinodynamic trajectory optimization and control for car-like robots," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2017, pp. 5681–5686.