2023 IEEE International Conference on
Autonomous Robot Systems and Competitions (ICARSC)
Tomar, Portugal – April 26-27, 2023

2023 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC) | 979-8-3503-0121-2/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICARSC58346.2023.10129639

Navigation and docking maneuvers control of an autonomous omnidirectional platform in a dynamic internal logistics environment

João Correia
centro ALGORITMI
University of Minho
Portugal
a83621@alunos.uminho.pt

Luís Louro
centro ALGORITMI
University of Minho
Portugal
llouro@dei.uminho.pt

Sérgio Monteiro
centro ALGORITMI
University of Minho
Portugal
sergio@dei.uminho.pt

Wolfram Erlhagen
dept. Mathematics and Applications
University of Minho
Portugal
Wolfram.erlhagen@math.uminho.pt

Estela Bicho
centro ALGORITMI
University of Minho
Portugal
estela.bicho@dei.uminho.pt

***Abstract* — This paper presents a method to control and generate motion for omnidirectional mobile manipulators that operate in internal logistics scenarios. It proposes transport and maneuvering operations where human operators may coexist. Thus, to ensure that human operators can easily infer the vehicle's movements, the vehicle preferentially performs non-holonomic (i.e., like a differential drive vehicle), and only takes advantage of holonomic movements in narrow spaces and when maneuvering. On the other hand, the docking maneuvers method enables the system to perform the approach and departure processes to the workstation or parking slot. These behaviors are orchestrated by a dedicated management controller, designing a unified solution. Simulation results demonstrate the system's performance and robustness to perform a complex industrial service in a dynamic environment.**

Keywords—docking maneuvers, omnidirectional platform, autonomous navigation, holonomic movements.

I. INTRODUCTION

The globalization of the economic market and the instability established in the current economy, are some of the external factors that shaped the *modus operandi* of enterprises, driving the research and implementation of new methods to improve productivity and profitability [1]. Typically, the strategy focuses on the introduction of automated mechanisms, mainly on the production side, increasing manufacturing capacity and reducing labor costs. However, transport and handling operations are still mostly manual processes [2], making the process asynchronous, and impairing operational performance as a result of sub-optimal utilization of available resources.

This paper proposes a solution that allows the optimized use of the available industrial resources, such as machine tools, providing seamless operation and synchronism between the overall processes, by developing an intelligent mobile solution, capable of attending nowadays industrial requirements, such as efficiency, flexibility, and optimization of industrial processes. Given the objectives and requirements, this paper presents the solution adopted for the omnidirectional mobile platform, which is part of a mobile manipulator, capable of performing manipulation operations, such as machine tending and pick-and-place tasks, and transport operations, including navigation and docking tasks. By definition, a mobile manipulator is a flexible robotic system composed of a robotic arm and a mobile platform, which enables manipulation and navigation operations, correspondingly [3]. Moreover, this combined with the collaborative robotics concept makes it possible to share the working environment with human operator's safety and unlock a wide range of possible cooperative tasks with them.

This paper focuses on transport operations, which means, the control and generation of mobile manipulator's movement in a dynamic industrial environment, considering the presence of human operators and the possibility of operating in confined spaces (e.g., access to workstations). Thus, while the transport efficiency is usually related to the ground floor characteristics, the influence could be minimized by using omnidirectional mobile platforms, allowing the definition of complex trajectories, such as lateral and diagonal movements. These types of movements, also known as holonomic movements, have special relevance in docking operations and navigation in narrow spaces, considerably reducing the number of maneuvers required and the total operation time. For this purpose, the main contribution of this paper is to present a low-level motion control planner, capable to deal with unstructured and dynamic environments.

In transport operations, the vehicle navigates on a dynamic factory floor, driven by target-following and obstacle-avoidance behaviors [4]. In this operation, despite the holonomic capacities, it was required, whenever possible, that the established movements be human-readable, i.e., that human operators intuitively understand the vehicle's maneuvering

---

This work is supported by: European Structural and Investment Funds in the FEDER component, through the Operational Competitiveness and Internationalization Programme (COMPETE 2020) [Project nº 42778; Funding Reference: POCI-01-0247-FEDER-042778]; and by FCT – Fundação para a Ciência e Tecnologia within the R&D Units Project Scope: UIDB/00319/2020.

intentions [5]. However, this driving requirement is not always possible, especially in operations in narrow environments. Therefore, holonomic navigation behavior was also considered, allowing the system to operate in tight spaces due to increased maneuverability. In the manipulation operations, the system performs the handling object tasks in pre-defined workspaces, such as on machine tools or assembly lines, performed by a collaborative robotic arm. For this purpose, the mobile platform should dock in a pose that maximizes the range of operation of the robotic arms, making these processes more efficient. To achieve the project's main goals, it was essential to automate the approach and departure procedures and establish them through holonomic movements. To orchestrate and articulate the described behaviors to establish a complete task, it was necessary to develop a mechanism to manage the execution of tasks depending on the state of the system. The proposed solution was implemented and verified in a dynamic simulation scenario, replicating a complex industrial environment.

The remainder of this paper is structured as follows: section II presents some existing projects in the literature related to mobile robotics topics. Then, the robotic system configuration and task constraints are described in section III; The overall system architecture is presented in section IV; Section V presents the dynamic behavior behind the navigation control methods. Further, in section VI the docking maneuvers control methods are described; Section VII presents the system's operational management and behavior orchestration. Section VIII presents the tests and validations performed to verify the system's performance. Lastly, in section IX the conclusion and future work are presented.

II. BACKGROUND

Mobile robots are being applied as internal logistics solutions in today's industries [6], providing flexibility and efficiency across industrial processes. From these, AMRs (Autonomous Mobile Robots) are state-of-art in the development and research of internal logistics solutions [7] due to their real-time adaptive capacities [8]. These concepts converge to current industrial requirements, driving research and development of new motion control methods, with a focus on dynamical systems [9].

There are several strategies and methods for controlling vehicle movements in dynamic environments. One such possibility is the fuzzy control method, showing its performance in establishing an optimal and smoother route during navigation, as Basheer Essa *et al.* claim [10]. These control methods qualitatively characterize the input into a fuzzy output according to predefined rules. On the other hand, empirical control methods are methodologies based on empirical information that allows simplifying complex considerations compared to traditional control methods. In this technique, the system's behavior is trained by machine learning techniques during the learning process. Considering the research field of autonomous navigation, Choi *et al.* approach is emphasized [11], applying the reinforcement training technique in the system's behavior definition. Another possible control strategy is presented by Louro *et al.*, by formulating the navigation dynamics based on the target following and obstacle avoidance behaviors [12]. Both behaviors are specified and parametrized as individual dynamical systems and integrated into a single dynamical system, which means that all components are active during the motion generation. Currently, most articles make use of existing state-of-the-art global path planners, complemented by local path planning algorithms, for dynamic obstacle avoidance and vehicle speed control [13].

Docking operations are considered a delicate and precise process, highly dependent on the vehicle's maneuverability capacities. Although tricycle vehicles are more usual in industrial environments, the complexity of trajectory formulation and orchestration during docking operations is notable. These docking operations can be formulated in two phases (approach and pose correction), and thus it is necessary to determine an auxiliary maneuver point, as proposed by Fan Guangrui & Wang Geng [14]. This type of constraint is not verified in holonomic vehicles, which due to their high maneuverability can move directly to the docking site.

Regarding navigation control and docking methods for omnidirectional vehicles, the proposal by Bavelos *et al.* stands out, presenting a solution with similar objectives to those intended to be achieved in this article. The navigation process is defined in a previously known environment, based on the SLAM (Simultaneous Localization And Mapping) algorithm. The mapping process generates a cost map, being then possible to calculate the "best path" [15], based on the sensorial information of the laser scanners. During navigation, the presence of dynamic obstacles is also considered, and the planned trajectory is subsequently redefined. Navigation accuracy and velocity are configured by parameterization of the SLAM algorithm and cost maps. In the docking process, the system compensates for the positioning error allowed in navigation. The control method is defined by a PID (Proportional–Integral–Derivative) controller, with feedback on the positioning error. The error is calculated by the difference between the desired final position (obtained via visual information from a fiducial marker) and the vehicle's frame. In turn, Xiuzhi Li *et al.* present a method focused on docking maneuvers for omnidirectional vehicles. The solution is based on a visual servo control method relative to the position error [16]. Based on the vehicle's inverse kinematics, the angular velocity to be applied to each wheel is calculated to transit the desired velocity vector for the motion.

The literature review shows that most navigation strategies rely on prior information about the operating environment (e.g., depending on SLAM algorithms). This condition imposes limitations in dynamic applications, concerning layout redefinition (as may occur in industrial environments). For this reason, we intend to minimize this dependency by proposing a navigation strategy depending on the information obtained from the space surrounding the platform, as presented in Louro et al. but for omnidirectional vehicles.

III. VEHICLE AND TASK CONSTRAINTS

A mobile manipulator is a robotic system consisting of a robotic arm and a mobile platform. Such a configuration allows the system to integrate various internal logistics operations in different workplaces, extending the work scope. An omnidirectional mobile manipulator was chosen as a use case, to minimize the required maneuvers to establish the docking operations, as well as allow the operation in tight workspaces.

The mobile platform is equipped with two laser scanners arranged diagonally in opposite corners, covering the platform's entire surroundings. Each sensor has a maximum range of 30 m with 0.5 ° resolution coverage, ensuring safe operation in dynamic environments. Four independent omnidirectional wheels (*mecanum* wheels type) drive the platform. Each wheel can be described as a free shaft bearing assembly, arranged transversally to each other on a rim, replicating a wheel. These wheels allow holonomic movement, i.e., they increase the maneuverability and flexibility of the mobile platform. The motion of the vehicle is established by defining the velocity vector, composed of the longitudinal and lateral velocities and angular velocity, v_x , v_y and ω_{vehi} respectively. Individual wheel motion control is linked to the vehicle's intrinsic characteristics with the desired velocity vector. This linking is established as in (1), where r_{wheel} is the wheel radius, $\alpha_i$and l_i the orientation and linear distance to the i wheel, relative to the vehicle's center, Υ the wheel's cylinders rotation orientation and β_i the rotation transforms between vehicle and wheel referential.

$$\omega_i = v_x \frac{-\cos(\beta_i - \Upsilon)}{r_{wheel}\sin(\Upsilon)} + v_y \cdot \frac{-\sin(\beta_i - \Upsilon)}{r_{wheel}\sin(\Upsilon)} + \omega_{vehi} \cdot \frac{-l_i \sin(-\alpha_i + \beta_i - \Upsilon)}{r_{wheel}\sin(\Upsilon)} \tag{1}$$

The mobile manipulator control system is structured in the ROS environment, networking the overall system's architecture.

The operation of the omnidirectional platform in dynamic environments, shared with human operators and dynamic objects, requires special safety considerations: i) the system should move only on its lane, especially in transportation tasks; ii) in navigation operations, non-holonomic movements should be prioritized as they are predictable to humans in the surroundings; iii) keeping a dynamic safety area around the mobile manipulator (dependent on the operation type and minimum safety distances imposed); iv) docking areas should be kept clear (due to the reduced safety margins). Docking maneuvers are considered critical processes, ensuring an accurate and adequate final docking posture for subsequent handling operations. The existing space next to the docking area is so small that the presence of human operators can make it difficult to maneuver the vehicle.

IV. SYSTEM ARCHITECTURE

The overall architecture is formulated in a four-modular hierarchy, which specifies the system's required characteristics (see Fig. 1), i.e., the external communication module, the motion controller module, the service manager module, and the vehicle perception and location module. The system is designed to interpret several predefined operations, received from an external server as operation requests or *services* (e.g., "Load milling machine"). The *Service Manager* receives and interprets the *services* and splits them into a set of successive tasks. These tasks are transmitted to the management and control component (the *Task Manager* component), which formulates the received tasks into primary operations. According to the operation purpose and system state, the *Task Manager* orchestrates the motion controller components: *Navigation*, which implements autonomous navigation behavior between two different locations in a dynamic environment; *Omnidirectional Navigation*, similar to the *Navigation* component, but uses the holonomic capacities of the platform (to operate in narrow environments); *Go to Dock*, which defines the docking approach maneuvers (when close to the workspace); *Return from Dock*, which establishes the exit maneuver from the workspace; *Go to Park*, similar to *Go to Dock,* but this component corrects the approach maneuver so the final movement is parallel and oriented to the battery charge connectors; *Return from Park*, establishes a safe exit movement, considering the charger connectors and the park spot layout. The handling operations are performed by the robotic arm once the mobile platform is docked at the requested workstation, i.e., at the end of the *Go to Dock*.

To ensure a safe and accurate operation in industrial environments, i.e., to establish a collision-free trajectory to accomplish a desired goal, the *Movement Controller* components depend on the mobile manipulator's location and environmental surroundings. The *Environment Perception* module components provide such information after processing raw data from sensors.

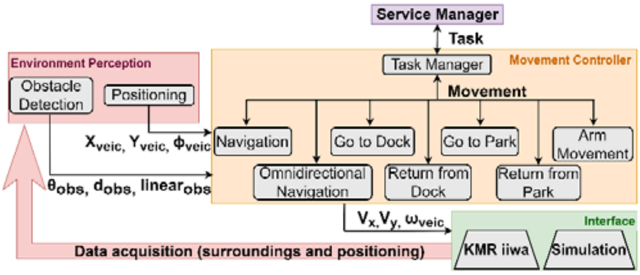


Fig. 1: Overall system's architecture

V. AUTONOMOUS NAVIGATION DYNAMICS

According to the operation workspace dimension and the type of maneuvers to be performed, the movements can be defined either by the *Navigation* or by the *Omnidirectional Navigation* components, differing in the movements' definition. Both navigation strategies are based on nonlinear dynamical systems theory, with motion defined by the behaviors of following a target and collision avoidance. These behaviors are formulated by the time course of the vehicle's navigation orientation, $\Phi_j(t)$ $(j = vehi, nav)$, and the motion velocity, $v(t)$, as stated in (2) and (3), correspondingly:

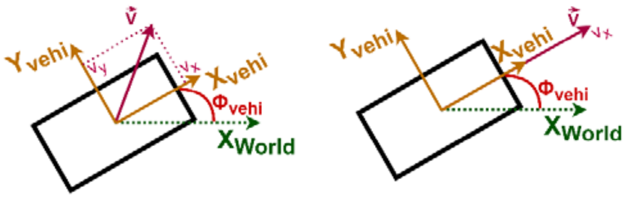
$$\frac{d\Phi_i}{dt} = \left(f_i(\Phi_j) + f_i(\Phi_j)\right) \tag{2}$$

$$\frac{dv}{dt} = c_i g_i(v) + c_i g_i(v) \tag{3}$$

where each individual contribution, f_i and g_i $(i = tar, obs)$, raises an attractor point at the desired value, the latter being controlled by c_i component (activation component).

A. Navigation established by legible movements

The *Navigation* component was designed to implement transport operations in dynamic environments by establishing human-legible movements. This was achieved by ensuring that the velocity vector always has the direction of the vehicle's physical orientation Φ_{vehi} (see Fig. 2), this being one of the control variable, and then the modulus of the velocity vector (v_x in this case). The motion results from the integration of two behaviors: moving toward a desired target and avoiding obstacles.



(a) (b)

Fig. 2: Movement definition: a) By holonomic movements; b) By legible movements. (like a differential-drive vehicle)

1) Heading direction control: Steering the system to the desired route is accomplished by controlling Φ_{vehi} along the time (assuming it is moving, i.e., $v_x > 0$), replicating the steering of a differential drive vehicle. Considering an obstacle-free environment, the mobile platform follows the predefined route as a sequence of via points. Subsequently, the control system converges the platform´s heading to the calculated via-point relative orientation, i.e., Ψ_{tar}, (both calculated in external world coordinates) (see [4],[12]). This behavior is shaped through the following dynamical system:

$$f_{tar}(\Phi_{vehi}) = -\lambda_{tar}\sin(\Phi_{vehi} - \Psi_{tar}) \tag{4}$$

by establishing a fixed attractor point at $\Phi_{vehi} = \Psi_{tar}$, with strength λ_{tar} (> 0). Similarly, the obstacle avoidance behavior is modeled as a sum of individual dynamic fields by erecting a fixed repulsive point at the orientation of each sector i of the laser sensor. Therefore, the overall behavior is modeled by:

$$f_{obs}(\Phi_{vehi}) = \sum_{i=0}^{n} f_{obs,i}(\Phi_{vehi}) \tag{5}$$

where $f_{obs,i}(\Phi_{vehi})$ specifies the individual repulsive force-let, defined by the laser beam i of n considered. The individual repulsive force-let is modeled by:

$$f_{obs,i}(\Phi_{vehi}) = \lambda_{obs,i}(\Phi_{vehi} - \Psi_{obs,i})e^{-\frac{(\Phi_{vehi}-\Psi_{obs,i})^2}{2.(\sigma_{obs,i})^2}} \tag{6}$$

which establishes a fixed stable point at $\Psi_{obs,i}$, with a repulsive strength defined by $\lambda_{obs,i}$ (> 0). The repulsive effect is bounded by an angular range according to the vehicle dimensions, sensor characteristics, and obstacle distances, established by $\sigma_{obs,i}$ (see [4] for more details).

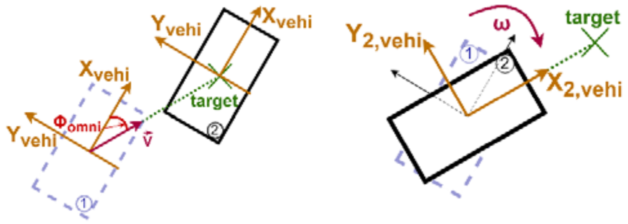
2) Motion velocity control: Similarly, to the heading direction control, linear speed control is also the result of two behaviors: target following and obstacle avoidance. The fundamental control theory is the same as for the previous control, i.e., establishes an attractor fixed point at the desired velocity:

$$G(v_x) = -c_{tar}(v_x - V_{tar}) - c_{obs}(v_x - V_{obs}) \tag{7}$$

which converges the system's linear velocity v_x, to V_{tar} or V_{obs}, depending on whether the path is free or obstructed (see [4] for more details). The behavior selection is performed by the single activation of c_{tar} (cleared route) or c_{obs} (obstructed route), which further specifies the respective relaxation rate.

B. Navigation established by holonomic movements

The *Omnidirectional Navigation* component generates holonomic robot movement, which allows it to maneuver in narrow spaces. It does so by adding the control of lateral movement, v_y, to the previously presented components. Thus, in this component, in addition to maintaining control of the direction of the vehicle (in which the direction of navigation may not be equal to the direction of the front of the vehicle) and the modulus of the velocity vector, it becomes necessary to control the direction of the velocity vector.



(a) (b)

Fig. 3: Holonomic motion control strategy divided into two complementary methods: a) Velocity vector definition; b) Vehicle space orientation. (Perspective of movement from 1 to 2)

The control strategy is divided into two complementary control methods (see Fig. 3), these referring to the definition of the linear velocity vector, performed by v_x and v_y, and to the vehicle orientation, performed by Φ_{vehi}.

1) Linear motion direction control: Considering the holonomic characteristics, the method must consider the hypothesis that the orientation of the movement (linear movement) does not coincide with the orientation of the vehicle's front (space orientation), i.e., $\Phi_{vehi} \neq \Phi_{omni}$, where Φ_{omni} is the linear motion orientation. This particularity redefines the notation for the vehicle's navigation orientation concept:

$$\Phi_{nav} = \Phi_{vehi} + \Phi_{\mathrm{omni}} \tag{8}$$

where Φ_{nav} specified the actual navigation orientation. The follow to a target behavior is modeled using a dynamical system identical to that in (4), but with respect to the vehicle´s navigation orientation, Φ_{nav}:

$$f_{tar_{omni}}(\Phi_{nav}) = -\lambda_{tar_{omni}}\sin(\Phi_{nav} - \Psi_{tar}) \tag{9}$$

erecting an attractor fixed point at the target orientation, Ψ_{tar}, with attraction strength $\lambda_{tar_{omni}}$ (> 0). The collision avoidance behavior has an identical definition as in (5), i.e., is modeled as a sum of individual repulsive force-let, each one modeled by:

$$f_{obs_{omni},i}(\Phi_{nav}) = \lambda_{obs_{omni},i}(\Phi_{nav} - \Psi_{obs,i}).\,e^{-\frac{(\Phi_{nav}-\Psi_{obs,i})^2}{2.(\sigma_{obs,i})^2}} \tag{10}$$

which erects a repulsive fixed point at $\Psi_{obs,i}$, with a repulsive strenght $\lambda_{obs_{omni},i}$ (> 0). As stated before, $\sigma_{obs,i}$ corresponds to the angular range over which the repulsive force is exerted by the i sector of the laser sensor considered. The overall system behavior is defined as the influence of both behavior:

$$\frac{d\Phi_{nav}}{dt} = f_{tar_{omni}}(\Phi_{nav}) + \sum_{i=0}^{n} f_{obs_{omni},i}(\Phi_{nav}) \quad (11)$$

As matter of principle, this control method is capable of compensating and correcting possible intrinsic and extrinsic influences on the system (such as the dynamics of the vehicle itself). However, the main goal is to extract the desired navigation orientation, Φ_{nav}, from the differential equation (11), to subsequently determine Φ_{omni}. This was achieved through the progressive Euler method strategy.

The velocity in modulus is determined identically (7) to the one of the navigation component, taking into account both behaviors. Therefore, the linear motion control variables are calculated as the vector projection on a two-dimensional plane, where v_{omni} is the calculated motion velocity:

$$v_x = v_{omni}\cos(\Phi_{\mathrm{omni}}) \quad (12)$$

$$v_y = v_{omni}\sin(\Phi_{\mathrm{omni}}) \quad (13)$$

2) Orientation Control: The orientation control strategy is similar to the one presented for legible movements, i.e., modeling both behaviors (follow to a target and obstacle avoidance) in dynamical systems. Therefore, the overall system behavior is shaped by integrating of (4) and (6), specifying however different attractive and repulsive strengths, $\lambda_{tar,ori}$ (> 0) and $\lambda_{obs,ori}$ (> 0) respectively.

The emphasis on the holonomic nature translates into the prevalence of the vehicle motion orientation control method (Fig. 3 a)) over the vehicle orientation control (Fig. 3 b)). As an example, when the orientation of the target to the vehicle's trajectory is changed (Ψ_{tar}), it converges the direction of the velocity vector to the desired orientation faster than the orientation of the front of the vehicle to the desired orientation. This is achieved by ensuring a $\lambda_{\mathrm{tar}_{\mathrm{omni}}} > \lambda_{\mathrm{tar,ori}}$. The same is true for obstacle behavior.

VI. DOCKING MANEUVERS CONTROL

The docking operations must ensure that the system performs the approach and departure procedures safely and securely. These maneuvers are performed by exploring the vehicle's holonomic characteristics, thus reducing the number of movements required as well as the maneuvering space to perform them. Motion control is formulated in two complementary methods: spatial orientation control of the vehicle and the vehicle's linear motion control. Both strategies are based on a proportional controller.

A. Space orientation control

The heading correction is established by controlling ω_{vehi} as a function of the orientation error, $\Phi_{dock_{error}}$. This error is determined depending on the desired docking orientation, Ψ_{dock}, and the vehicle orientation, Φ_{vehi} (see Fig. 4):

$$\Phi_{dock_{error}} = \Psi_{dock} - \Phi_{vehi} \quad (14)$$

once the orientation error is determined, the control action is generated by a proportional controller, imposing a $k_{p,w}$ (> 0) gain:

$$\omega_{vehi} = k_{p,w}\Phi_{dock_{error}} \quad (15)$$

This control strategy is applied for both the approach and departure processes.

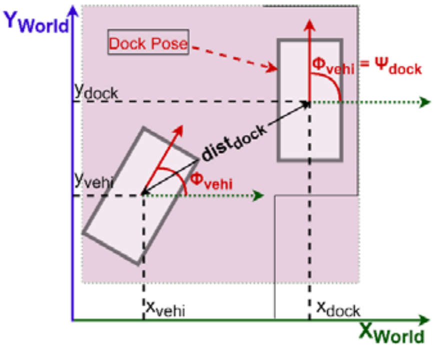


Fig. 4: Docking maneuver considering the desired dock pose.

B. Linear motion control

Linear motion control is formulated as a projection of the velocity vector onto a plane. Therefore, the control strategy is structured on two levels: linear motion direction control and linear motion velocity control.

1) Linear motion direction control: Given an obstacle-free environment, the system's linear movement must coincide with the desired dock location. Otherwise, the motion compensation mechanism is expected to act (c.f. in Section VI-C). Thus, the movement direction, θ_{dock}, is calculated as a function of the vehicle location (x_{vehi}, y_{vehi}) and the desired docking location (x_{dock}, y_{dock}). However, considering the assumption that the linear movement orientation does not coincide with the vehicle orientation (due to the holonomic characteristics) ($\theta_{dock} \neq \Phi_{vehi}$), the latter term is discounted in the calculation:

$$\theta_{dock} = \arctan^{-1}\left(\frac{y_{dock} - y_{vehi}}{x_{dock} - x_{vehi}}\right) - \Phi_{vehi} \quad (16)$$

where Φ_{vehi} specifies the vehicle orientation. This control strategy is applied for both the docking approach and departure processes.

2) Linear motion velocity control: Velocity is adjusted considering the positioning error (linear distance to the desired dock location), $dist_{dock}$:

$$dist_{dock} = \sqrt{(x_{dock} - x_{vehi})^2 + (y_{dock} - y_{vehi})^2} \quad (17)$$

depending on the docking operation, approaching, or departing, the path velocity should decrease or increase, respectively, as the vehicle reaches the desired docking location. Therefore, in approaching maneuvers, the system's linear motion velocity, v_{dock}, is modeled by a linear proportional controller:

$$v_{dock} = k_{p,appr} dist_{dock} \tag{18}$$

imposing a $k_{p,appr}$ (> 0) gain. On the other hand, in departing maneuvers, the system´s linear velocity control is shaped by a quadratic proportional controller so that the vehicle gradually increases the modulus of the linear speed as a function of proximity to the imposed extraction location (decreasing of $dist_{dock}$):

$$v_{dock} = -k_{p,depar}(dist_{dock})^2 + V_{max} \tag{19}$$

where $k_{p,depar}$ (> 0) gain is defined considering that the maximum velocity, V_{max}, occurs when $dist_{dock} \approx 0$, considering the initial dock distance, $dist_{initial,dock}$:

$$k_{p,depar} = \frac{V_{max}}{dist_{initial,dock}} \tag{20}$$

The linear motion control variables (v_x, v_y) are calculated as in (12) and (13).

C. Safety mechanism

The docking processes are complemented by a safety mechanism, to avoid collisions with possible obstacles in the maneuvering area. However, its use is limited due to the very small safety margins. This mechanism can override the previously generated control actions if the safety distance to obstacles is not respected. First, the presence of obstacles closes to the vehicle that infringe on the safety margin considered is verified. In this case, if the direction of the linear motion coincides with obstacle's location, but only in one control variable (e.g.: v_y), the mechanism cancels it while maintaining the other. Moreover, if the distance to the nearest obstacle infringes on the minimum safety distance to continue the operation, the mechanism immobilizes the vehicle. Docking processes are designed to operate in critical proximity to obstacles. For this reason, the margin considered is variable, being less sensitive the closer to the dock location (if approaching) (21). In the extraction operation, the opposite is true (22), where c parameterizes the sensitivity of the maneuver.

$$safety_{margin} = c.dist_{dock} \tag{21}$$

$$safety_{margin} = c.\frac{1}{dist_{dock}}, dist_{dock} > 0 \tag{22}$$

VII. BEHAVIOR ORCHESTRATION

As stated in section III, the system operation is established by the management and activation of primary operation types, performed by the *Task Manager* component. The orchestration of the behavior depends on the assigned task (e.g.: navigating to a workstation, or returning to the parking slot, among other possibilities) as well as on the system state (i.e., the current task status, e.g.: approaching the docking zone, navigating in a narrow area). Therefore, in the first instance, the last executed operation sub-task is verified. Considering the beginning of the action, these can be the *Go to Dock* or the *Go to Park* operation (only these behaviors are valid as first sub-task operation behavior), selecting the complementary behavior maneuver operation (*Return from Dock* or *Return from Park*, accordingly).

Once the procedure is successfully completed, the *Task Manager* selects the navigation component according to the characteristics of the navigation space. Two navigation options have been considered: navigating by holonomic movements (*Omnidirectional Navigation* component) or as a differential-drive vehicle (*Navigation* component). If the navigation space is considered narrow (demanding in terms of maneuvering space), the *Omnidirectional Navigation* behavior is selected. Otherwise, the *Navigation* component is activated (wide maneuvering space). The space classification is contained in the narrow area flag, received in the service, signaling narrow spaces operation.

When the omnidirectional platform is in the vicinity of the dock location (the entire planned route has already been traveled), the *Task Manager* selects the approach maneuver (depending on the assigned task). Thus, if the goal is to dock at a work area, the *Go to Dock* maneuver behavior is selected. On the other hand, if the goal is to return to the parking slot, the *Go to Park* is activated.

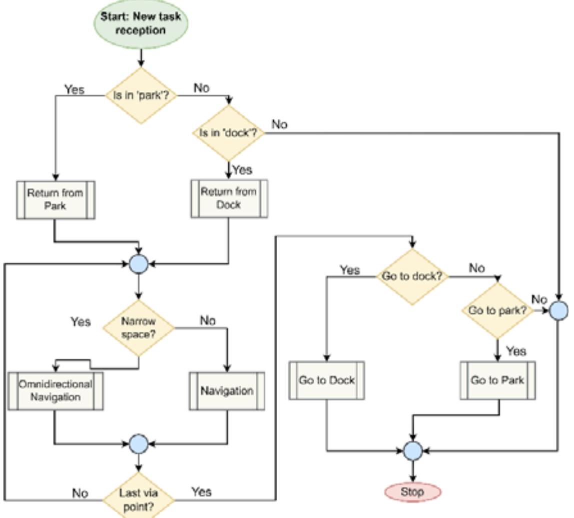


Fig. 5: Illustration of the operation orchestration in a flowchart.

The *Task Manager* continuously monitors the execution of the various components' tasks, and in case of any failure, it aborts the task execution, informing the *Service Manager* about the error. Fig. 5 illustrates the operation orchestration performed by the *Task Manager* according to the received task.

VIII. SIMULATION VALIDATION

This section concerns the tests and validations performed to verify the system's overall behavior in a simulated realistic industrial environment. For this purpose, a simulation scenario was developed in *CoppeliaSim*, considering a complex industrial floor as well as the dynamic model of the mobile platform. This validation method allowed us to analyze the system behavior in the execution of a typical internal logistic task and verify the behavior in a dynamic environment (environment composed of dynamic simulation models: mobile platform and human operators' models).

A. *Methodology and test design*

The method validation was specified by integrating the system control processes with the *CoppeliaSim* simulation scenario. These were interfaced through mediation by ROS (Robotic Operation System) communication schemes: ROS *Topic* (for data stream purposes) and ROS *ActionLib* (for high-level communication). A complex industrial service, that requires all the proposed motion controllers, was requested to be executed. Then, the *Service Manager* divided it into the following three actions: *Unload Production* (to navigate and dock to the pickup point); *Load Milling* (to navigate and dock to the machine tool feeding zone); *Go Park* (to return to parking zone).

B. *Unload Production: Navigation and docking to the collecting workspace*

As illustrated in Fig. 6, this action is split into four distinct behaviors: *Return from Park*, *Navigation*, *Omnidirectional Navigation*, and *Go to Dock.*

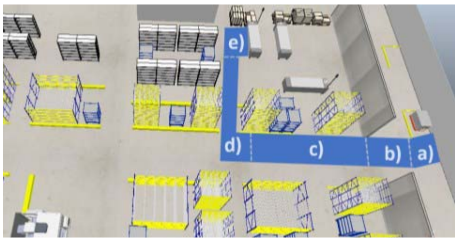


Fig. 6: Behavior representation that defines the *Unload Production* action. a) *Return from Park*; b) *Omnidirectional Navigation*; c) *Navigation*; d) *Omnidirectional Navigation*; e) *Go to Dock*.

As priorly stated, behavior orchestration is performed by the *Task Manager*. Since the initial system location coincides with a parking zone, the *Return from Park* component was selected (Fig. 7 a)). Once completed, the system is governed by the navigation components. Given the limited workspace in the environment, the mobile robot started by using the *Omnidirectional Navigation* mode (Fig. 7 b)). Then, in the main corridor, the system motion became defined by readable movements (*Navigation* component) (Fig. 7 c)). Due to the limited room to maneuver at the workstation access, the system behavior was once again defined by holonomic movements (Fig. 7 d)), and subsequently established by the *Go to Dock* component (for pose correction) (Fig. 7 e)). The final dock pose was specified to maximize the working space of the robotic arm (future intention of conjugation with motion generation of a robotic arm attached to the mobile platform).

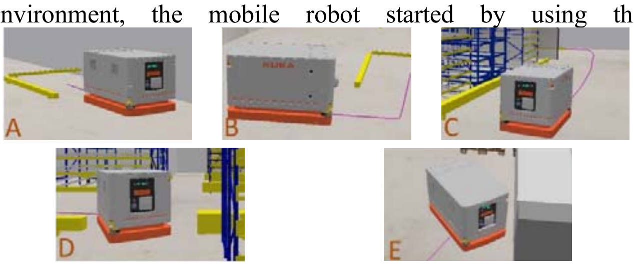


Fig. 7: Snapshots sequence of the *Unload Production* action. (https://youtu.be/BYTV8281Dfc)

C. *Load Milling: Navigation and docking to machine tool feeding zone*

This action is split similarly to *Unload Milling*, except for the starting operation, which is a *Return from Dock*, as shown in Fig. 8, being the complementary operation to the dock state (last active state).

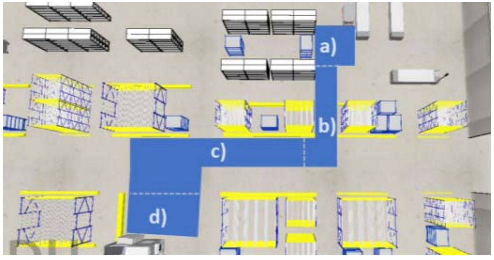


Fig. 8: Behavior representation that defines the *Load Milling* action. a) *Return from Dock*; b) *Omnidirectional Navigation*; c) *Navigation*; d) *Go to Dock*.

After *Return from Dock* (Fig. 9 a)), and once cleared the workspace zone, the goal is to access the main corridor. Here, due to the narrow space, the system´s motion was specified by the *Omnidirectional Navigation* component (Fig. 9 b)). After accessing the main corridor, the system is controlled by the *Navigation* components, defining readable movements (Fig. 9 c)). This behavior was active until near the machine tool feeding zone and then controlled by the *Go to Dock* component (pose adjustment) (Fig. 9 d)).

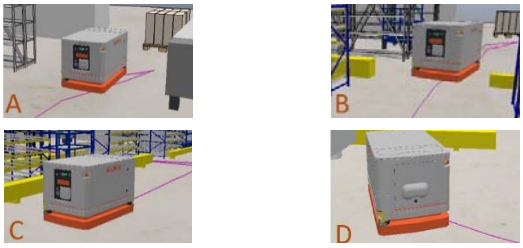


Fig. 9: Snapshots sequence of the *Load Milling* action. (https://youtu.be/BYTV8281Dfc)

D. *Go Park: Returning to the parking zone*

This action is orchestrated in a similar way to the ones before, differing in the last approach component: *Go to Park* instead of the *Go to Dock* behavior. This last component capacitates the system to perform an appropriate approach to the battery charger connectors. The system's behavior can be verified in the last action of the video: https://youtu.be/BYTV8281Dfc.

E. *Dynamic Environment Operation: Safe crossing with a human operator at the main corridor*

This experiment validates the system's capacity to share the factory floor with human operators by defining legible movements. This experiment condition occurred during the return navigation to the park site. As illustrated in the Fig. 10 sequence, the mobile platform established a safety bypass maneuver to the right, avoiding the action of the stop mechanism or potential incidents.

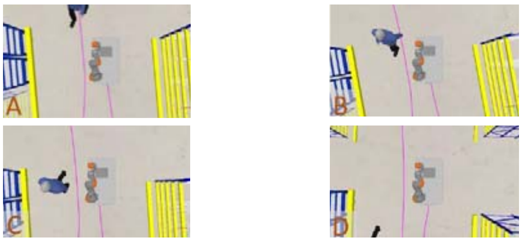


Fig. 10: Snapshots sequence of a safe crossing between the mobile manipulator and the human operator. (https://youtu.be/SUva9dcU9Rg)

F. *Dynamic Environment Operation: Congested workspace access corridor*

This experiment demonstrates the system´s ability to avoid an imminent collision during access to the workspace corridor (narrow access). As demonstrated in the Fig. 11 sequence, the system was able to compensate for the movements by the action of the safety stop mechanism (critical proximity) (Fig. 11 a)) and subsequently access trajectory correction (by holonomic movements) (Fig. 11 snapshots B and C).

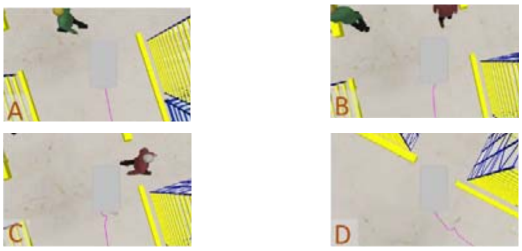


Fig. 11: Snapshots sequence of a safe crossing between the mobile manipulator and the human operator. (https://youtu.be/qrymVdjKcBA)

IX. CONCLUSIONS AND FUTURE WORK

This paper proposes an internal logistics solution based on an omnidirectional mobile manipulator. Motion control was designed considering both the environment layout and the presence of human operators in the shared workspace. This method was based on a low-level planner, using dynamical systems. As matter of principle, this approach is also able to compensate for possible slippage problems. Methods allowing docking maneuvers were developed to perform the approach and departure trajectory to a workstation or a parking slot, considering the final pose requested and trajectory restrictions. The overall system behavior results from behavior orchestration by a dedicated management controller, given the task to be performed as well as the system’s operational state.

Simulation results demonstrate the system’s performance to perform a complex industrial service, composed of several individual tasks, and its robustness to operate in dynamic environments, being able to adapt to operating circumstances. Future implementations include both the integration of this solution with the robotic arm motion controller, and the system´s validation in a real industrial context.

REFERENCES

[1] Ignatiev, K. V., Kopichev, M. M., & Putov, A. V. (2016, May). Autonomous omni-wheeled mobile robots. In 2016 2nd International Conference on Industrial Engineering, Applications and Manufacturing (ICIEAM) (pp. 1-4). IEEE.

[2] Porunov, M., & Trifonov, A. (2020, October). Development of a special module for an industrial collaborative robot for assembling standard threaded products. In 2020 International Multi-Conference on Industrial Engineering and Modern Technologies (FarEastCon) (pp. 1-5). IEEE.

[3] Mestiri, Y. (2022). Mobile manipulator robot: omni 3 wheels manipulator robot (Doctoral dissertation).

[4] Bicho, E. (1999). Dynamic Approach to Behavior-Based Robotics: design, specification, analysis, simulation and implementation. Universidade do Minho (Portugal).

[5] Faria, M., Melo, F. S., & Paiva, A. (2021, August). Understanding robots: Making robots more legible in multi-party interactions. In 2021 30th IEEE International Conference on Robot & Human Interactive Communication (RO-MAN) (pp. 1031-1036). IEEE.

[6] Schneier, M., Schneier, M., & Bostelman, R. (2015). Literature review of mobile robots for manufacturing (p. 21). Gaithersburg, MD, USA: US Department of Commerce, National Institute of Standards and Technology.

[7] Markis, A., Papa, M., Kaselautzke, D., Rathmair, M., Sattinger, V., & Brandstötter, M. (2019, May). Safety of mobile robot systems in industrial applications. In Proceedings of the ARW & OAGM Workshop (pp. 26-31).

[8] Melo, A. F., & Corneal, L. M. (2020). Case study: evaluation of the automation of material handling with mobile robots. International Journal of Quality Innovation, 6(1), 1-14.

[9] Papa, M., Kaselautzke, D., Stuja, K., & Wölfel, W. (2018). DIFFERENT SAFETY CERTIFIABLE CONCEPTS FOR MOBILE ROBOTS IN INDUSTRIAL ENVIRONMENTS. Annals of DAAAM & Proceedings, 29.

[10] Basheer Essa, A., Al-Mayyahi, A., Wang, W., Hussien, A., & Birch, P. (2017). Motion control design for an unmanned ground vehicle in a dynamic environment using an intelligent controller. In International Journal of Intelligent Computing and Cybernetics (Vol. 10, Issue 4).

[11] Choi, J., Lee, G., & Lee, C. (2021). Reinforcement learning-based dynamic obstacle avoidance and integration of path planning. Intelligent Service Robotics, 14, 663-677.

[12] Louro, L., Teixeira, D., Malheiro, T., Mesquita, L., Machado, T., Monteiro, S., ... & Bicho, E. (2020, October). A safe autonomous stacker in human shared workspaces. In IECON 2020 The 46th Annual Conference of the IEEE Industrial Electronics Society (pp. 4389-4396). IEEE.

[13] Chen, C. S., Lin, C. J., Lai, C. C., & Lin, S. Y. (2022). Velocity Estimation and Cost Map Generation for Dynamic Obstacle Avoidance of ROS Based AMR. Machines, 10(7), 501.

[14] Guangrui, F., & Geng, W. (2017, December). Vision-based autonomous docking and re-charging system for mobile robot in warehouse environment. In 2017 2nd International Conference on Robotics and Automation Engineering (ICRAE) (pp. 79-83). IEEE.

[15] Bavelos, A. C., Kousi, N., Gkournelos, C., Lotsaris, K., Aivaliotis, S., Michalos, G., & Makris, S. (2021). Enabling flexibility in manufacturing by integrating shopfloor and process perception for mobile robot workers. Applied Sciences, 11(9), 3985.

[16] Li, X., Liang, X., Fan, J., & Jia, S. (2018, August). Position-based visual servo control of intelligent wheelchair/bed docking. In 2018 IEEE International Conference on Information and Automation (ICIA) (pp. 1497-1502). IEEE.