ISSN 1590-8844
International Journal of Mechanics and Control, Vol. 26, No. 01, 2025
https://doi.org/10.69076/jomac.2025.0003

OBSTACLE AVOIDANCE METHODS FOR DIFFERENTIAL DRIVE ROBOTS

Federico Neri* Giacomo Palmieri* Daniele Costa* Massimo Callegari*

* Department of Industrial Engineering and Mathematical Sciences (DIISM), Università Politecnica delle Marche, Ancona, Italy.

ABSTRACT

This paper presents an extended obstacle avoidance approach for non-holonomic mobile robots with differential drive transmission, focusing on collaborative environments where human operators and robots share the workspace. Two distinct avoidance strategies are evaluated: one combining translational and rotational movements for greater flexibility, and the other relying solely on rotational maneuvers to ensure smoother trajectory adjustments. The performance of these strategies is assessed through simulations in scenarios involving both static and dynamic obstacles. The first test shows a mobile robot navigating along a path where an obstacle is located, while in the second the mobile robot is fixed at a target position and an obstacle interferes. The robot's direction only changes when the obstacle enters the zone of influence. The possibility of using a rectangular kinematic structure with a tip at the front of the robot is introduced. Future developments will focus on implementing these strategies within a mobile manipulator system, comprising a differential drive base and a 6-DOF robotic arm, to expand the algorithm's applicability to advanced robotic configurations.

Keywords: obstacle avoidance; differential drive; non-holonomic; motion planning

1 INTRODUCTION

The increasing adoption of autonomous mobile robots (AMRs) in industrial environments has necessitated the development of advanced navigation strategies that can ensure safety and maintain efficiency without reducing cycle times. Unlike automated guided vehicles (AGVs), which rely on fixed routes and external infrastructure [1], AMRs use perception systems to dynamically adapt to their surroundings. This capability is particularly important in shared workspaces between humans and robots, where obstacle avoidance helps prevent collisions [2]. In recent years, significant advancements have been made in the development of mobile robots capable of operating in various environments [3]. These systems have been widely adopted not only in industrial and healthcare settings, where they navigate flat and structured surfaces [4], but also in more challenging terrains such as agricultural fields, where rough and uneven surfaces require advanced locomotion strategies [5]. The ability to adapt to different working conditions is important for expanding the applicability of mobile robots beyond controlled environments, enabling their deployment in outdoor and unstructured scenarios.

Another critical aspect of mobile robotics research is localization, which ensures precise positioning within a given workspace. The most commonly employed method relies on an initial environmental mapping performed at the time of installation. While this approach is widely adopted, it has notable limitations, particularly in modular or frequently changing environments such as warehouses or dynamic industrial settings. Structural modifications or temporary obstacles can compromise the accuracy of the map-matching algorithm, leading to localization failures and reduced navigation reliability [6].

To address these challenges, alternative localization strategies have been explored. One promising solution involves Ultra-Wideband (UWB) positioning systems, which provide

E-mail: federico.neri@staff.univpm.it, g.palmieri@univpm.it, d.costa@staff.univpm.it, m.callegari@staff.univpm.it

high-accuracy localization even in dynamic environments by leveraging time-of-flight measurements [7]. Another effective approach is the use of motion capture systems based on marker detection, such as OptiTrack, which enables real-time tracking with sub-millimeter precision [8]. These technologies, when integrated with onboard sensor fusion techniques, can significantly enhance the robustness and adaptability of mobile robot navigation, ensuring reliable performance even in highly dynamic and unstructured environments.

The reliance on predefined paths in mobile robot navigation presents significant challenges in dynamic environments. When obstacles suddenly appear or the workspace layout changes, a precomputed trajectory may become invalid, requiring the robot to stop and recalculate a new path. This process not only introduces inefficiencies but also increases the risk of navigation failures if real-time updates are not feasible. To address these issues, it is crucial to develop advanced algorithms capable of real-time obstacle avoidance [9], allowing robots to dynamically adjust their trajectories and ensure continuous, collision-free operation without the constraints of static path planning [10, 11].

Obstacle avoidance strategies in mobile robotics have been extensively studied [12, 13], with numerous approaches attempting to balance computational efficiency and real-time responsiveness. Global planning methods, such as A* and Dijkstra's algorithms, are widely used for generating optimal trajectories in static environments by computing the shortest path from a given start point to a goal while accounting for known obstacles [14]. However, their effectiveness is significantly reduced in dynamic environments, where obstacles can change over time, requiring more adaptive strategies to ensure safe and efficient navigation. In contrast, reactive methods, such as artificial potential fields [15] and vector field histogram, offer an immediate response to obstacles, but may suffer from problems such as local minima or oscillatory behaviour. Hybrid approaches attempt to combine the advantages of both solutions, providing a balance between long-term planning and immediate responsiveness. However, their application to non-holonomic robots with differential transmission presents a challenge, as these systems impose kinematic constraints that limit their manoeuvrability compared to omnidirectional platforms [16]. In current studies of obstacle avoidance techniques for mobile robots, it is often assumed that the robotic platform has almost unlimited mobility [17], neglecting the kinematic restrictions imposed by non-holonomic systems, such as those with differential drive. This leads to solutions that are effective in ideal simulations or on omnidirectional platforms, but may be difficult to apply to differential mobile robots in the real world. This research addresses this limitation in more detail than the existing literature by developing obstacle avoidance strategies that incorporate the robot's actual kinematic constraints. This approach allows for more effective and applicable solutions in real-world scenarios, improving navigation and safety in complex operational environments.

Regarding the kinematics of mobile robots, two main macrocategories can be identified: omnidirectional wheel systems and differential drive systems. Omnidirectional robots are equipped with specialized wheels, such as Mecanum or omnidirectional rollers, which allow movement in any direction without requiring rotation of the robot's body. This capability provides high maneuverability, making these robots particularly suitable for constrained environments where precise positioning is required, such as warehouses, medical facilities, and assembly lines [18, 19].

In contrast, differential drive robots rely on two independently controlled wheels to achieve movement [20, 21, 22]. By varying the relative speed of each wheel, the robot can translate forwards, backwards and rotate around an axis. However, due to their non-holonomic constraints, these robots cannot move laterally, which can limit their maneuverability in tight spaces [23]. Despite this, differential drive systems remain widely used due to their mechanical simplicity, cost-effectiveness, and robustness, making them ideal for applications in structured environments such as industrial automation and service robotics. The absence of lateral movements imposes constraints on path planning and requires advanced strategies to ensure efficient and smooth obstacle avoidance. To overcome these challenges, this study presents an advanced obstacle avoidance algorithm tailored for differential wheeled mobile robots operating in dynamic environments. The proposed approach effectively manages both static and moving obstacles, with its performance evaluated through simulations that incorporate various motion strategies. A strategy is developed that allows the robot to temporarily reposition itself to avoid interference, ensuring smooth collaboration without operational interruptions. This approach improves the flexibility of the system, allowing the robot to resume its position autonomously once the work area is free, without the need for external intervention.

The paper is structured as follows: Section 2 provides a detailed description of the proposed algorithm, outlining the developed avoidance strategies and the criteria employed for their evaluation. Section 3 presents the simulation results, comparing the effectiveness of the different strategies and analyzing their performance in terms of efficiency, safety, and adaptability to dynamic environments. In Section 4, the findings are critically examined, highlighting the strengths and limitations of the proposed approach and discussing its applicability in real-world scenarios. Finally, Section 5 summarizes the key contributions of this work and outlines potential future developments.

2 OBSTACLE AVOIDANCE ALGORITHM

In a planar representation, the mobile robot can be modeled as a rectangular chassis with two wheels symmetrically

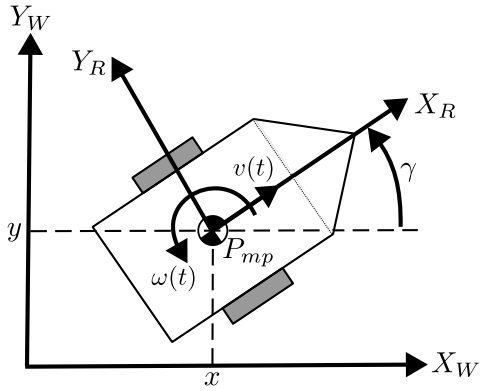


Figure 1 World and local reference frames.

positioned on either side. To simplify the analysis of the system, a local reference frame is defined, with the origin located at the center of the chassis, denoted as $\mathbf{P}_{mp}$. The axes X_R and Y_R are oriented as shown in Fig. 1. In relation to the world reference frame, the pose of the mobile platform is represented by the vector $\mathbf{x} = \begin{bmatrix} x & y & \gamma \end{bmatrix}^T$, where (x, y) denote the coordinates of the robot's position in the world frame, and γ represents the orientation of the robot, measured as the angle between the X_R axis and the global X_W axis.

Given the non-holonomic nature of the robot's locomotion, a constraint arises that prohibits motion along the Y_R direction. Consequently, the velocity of the mobile robot can only be expressed as a function of two variables: the linear velocity v along the X_R axis, and the angular velocity ω. Therefore, the robot's velocity vector in the local frame of reference is given by $\dot{\mathbf{q}} = \begin{bmatrix} v & \omega \end{bmatrix}^T$.

The velocity vector in the world reference frame can be obtained from the local velocity vector using a transformation matrix, known as the Jacobian matrix $\mathbf{J}$. This 3×2 matrix converts the local velocity components to the global frame and is given by

$$\mathbf{J} = \begin{bmatrix} \cos\gamma & 0 \\ \sin\gamma & 0 \\ 0 & 1 \end{bmatrix} \tag{1}$$

The velocity vector in the world reference frame can thus be obtained as $\dot{\mathbf{x}} = \mathbf{J}\dot{\mathbf{q}}$.

In order to generate a smooth and feasible point-to-point trajectory for a mobile robot with differential drive, a parametric cubic polynomial approach is utilized. Let the initial pose of the robot be $\mathbf{x}_i = \begin{bmatrix} x_i & y_i & \gamma_i \end{bmatrix}^T$ and the final pose be $\mathbf{x}_f = \begin{bmatrix} x_f & y_f & \gamma_f \end{bmatrix}^T$. The path between these two points is then expressed as:

$$\begin{bmatrix} x(s) \\ y(s) \end{bmatrix} = \\ = \begin{bmatrix} -(s-1)^3 x_i + s^3 x_f + \alpha_x s^2 (s-1) + \beta_x s (s-1)^2 \\ -(s-1)^3 y_i + s^3 y_f + \alpha_y s^2 (s-1) + \beta_y s (s-1)^2 \end{bmatrix} \tag{2}$$

where the parameter $s \in [0, 1]$ represents the normalized progression along the trajectory, and is defined as a function of time using a 5^{th} polynomial. The coefficients $\alpha_x, \alpha_y, \beta_x, \beta_y$ are introduced to further refine the shape of the trajectory, and are calculated as:

$$\begin{bmatrix} \alpha_x \\ \alpha_y \end{bmatrix} = \begin{bmatrix} k\cos\gamma_f - 3x_f \\ k\sin\gamma_f - 3y_f \end{bmatrix} \quad \begin{bmatrix} \beta_x \\ \beta_y \end{bmatrix} = \begin{bmatrix} k\cos\gamma_i + 3x_i \\ k\sin\gamma_i + 3y_i \end{bmatrix} \tag{3}$$

In these equations, the parameter k is a tuning variable that allows for adjusting the curvature radius of the path. The orientation along the trajectory, denoted by $\gamma(s)$, is computed by enforcing the condition that the local X_R axis remains tangent to the path at each point. Once the trajectory has been established, the reference velocity profile is derived from the velocity components in the world frame. The forward velocity v_r is given by the magnitude of the velocity vector $\sqrt{\dot{x}^2 + \dot{y}^2}$, while the angular velocity ω_r is simply the time derivative of the orientation $\dot{\gamma}$.

The proposed motion control system is designed to dynamically adjust the robot's trajectory in real time in response to obstacles that obstruct the platform's movement. A repulsive velocity vector is activated when the distance between the robot and the obstacle falls below a predefined threshold, represented by a safety radius r. This safety radius surrounds the robot and forms a protection zone on each side of the platform. This zone is modelled as a rectangular area, with rounded corners, in which the two semicircles share a common radius r.

The repulsive velocity vector $\dot{\mathbf{x}}_0$ is applied at the point $\mathbf{P}_r = \begin{bmatrix} p_x & p_y \end{bmatrix}^T$, which corresponds to a location on the segments of the robot's footprint that is closest to the obstacle. The position of the obstacle is denoted by $\mathbf{P}_o$, as shown in Fig. 2. The magnitude of the repulsive velocity, denoted by $\dot{x}_0$, is considered a control parameter and can be adjusted to modulate the strength of the repulsion. On the other hand, the direction of the repulsive velocity vector is determined by the unit vector along the distance vector $\mathbf{d}_o = \mathbf{P}_r - \mathbf{P}_o$, such that the repulsive velocity is expressed as $\dot{\mathbf{x}}_0 = \dot{x}_0 \hat{\mathbf{d}}_o$.

To describe the kinematics of the repulsive velocity vector, the Jacobian matrix $\mathbf{J}_0$ related to the point $\mathbf{P}_r$ is defined. This matrix relates the velocity components of the robot in the local frame of reference to the required velocity to avoid the obstacle. The kinematic expression of the repulsive velocity vector is derived as:

$$\dot{\mathbf{x}}_0 = \begin{bmatrix} \cos\gamma & -p_y\cos\gamma - p_x\sin\gamma \\ \sin\gamma & p_x\cos\gamma - p_y\sin\gamma \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} = \mathbf{J}_0 \begin{bmatrix} v \\ \omega \end{bmatrix} \tag{4}$$

In this expression, the Jacobian matrix $\mathbf{J}_0$ is used to transform the velocity components (v, ω) in the robot's local reference frame to the appropriate repulsive velocity $\dot{\mathbf{x}}_0$. The Jacobian depends on the configuration of the robot and the point of application of the repulsive force, allowing the robot to adjust its motion direction and speed in response to the obstacle's position and distance.

The strategy is designed to identify the link with the highest risk of collision by calculating the distance of each obstacle from each segment of the robot. As a result, the point $\mathbf{P}_r$ of minimum distance from the obstacle and the distance vector $\mathbf{d}_o$ can be obtained [24].

As illustrated in Fig. 2a, a significant problem can emerge when an obstacle interacts with the front edge of the moving platform. In such cases, the repulsive velocity generated may result solely from the translational movement of the platform without any rotational adjustment, causing the robot to move backwards along a straight path. Once the obstacle leaves the platform's area of influence, the control system orders the platform to return to its previous position, resulting in a repetitive stall scenario.

To overcome this limitation, a new strategy is proposed whereby a safety region is defined by dividing the frontal area of the platform into five segments that converge to form a tip, as illustrated in Fig. 2b. This configuration facilitates the simultaneous generation of rotational and translational velocities, allowing the robot to circumvent obstacles more effectively.

The proposed control strategy integrates two primary objectives: the primary task of following a predefined trajectory and a secondary task that is activated when the obstacle avoidance mechanism is triggered. The control law governing this behaviour can be expressed as follows:

$$\dot{\mathbf{q}} = \mathbf{v}_f + \mathbf{J}_0^{-1}\dot{\mathbf{x}}_0 \tag{5}$$

In this formulation, $\mathbf{v}_f$ represents the planned velocity profile with error compensation, as described by Maalouf et al. [25] and Kanayama et al. [26]. The expression for $\mathbf{v}_f$ is given by:

$$\mathbf{v}_f = \begin{bmatrix} v_f \\ \omega_f \end{bmatrix} = \begin{bmatrix} v_r\cos e_\gamma + k_x e_x \\ \omega_r + v_r k_y e_y + k_\gamma \sin e_\gamma \end{bmatrix} \tag{6}$$

Here, e_x, e_y, and e_γ denote the positional and orientation errors between the planned and actual poses in the robot's local reference frame. The terms k_x, k_y and k_γ are control gain parameters.

The parameters used in the proposed control strategy, such as the repulsive velocity $\dot{\mathbf{x}}_0$ and the recovery parameters k_x, k_y and k_γ, are essential for balancing obstacle avoidance and point-to-point trajectory tracking. The repulsive velocity $\dot{\mathbf{x}}_0$ directly influences the robot's responsiveness when an obstacle is detected, determining how quickly the robot will alter its path to avoid a collision. In contrast, the recovery parameters k_x, k_y and k_γ control how effectively the robot returns to its planned point-to-point trajectory after successfully avoiding the obstacle. A balance must be found between more aggressive avoidance behaviour, which increases safety but may compromise trajectory accuracy, and smoother trajectory tracking, which may reduce the robot's ability to react quickly to obstacles. Furthermore, these parameters depend on the maximum speeds that the robot can reach, which influence its ability to adapt to dynamic changes in the environment.

An alternative method of generating repulsive motion involves the imposition of purely rotational adjustments. In this scenario, only the component of the repulsive velocity perpendicular to the front edge of the platform is defined, with a fixed magnitude $v_{0n} = \dot{x}_0$, while the tangential component v_{0t} remains free. By imposing a zero velocity at the centre point of the platform $\mathbf{P}_{mp}$, the angular velocity required to produce the desired repulsive effect can be found as $\omega = \dot{x}_0/(p_y\cos\alpha_{mp} - p_x\sin\alpha_{mp})$, where the geometric variables are defined in Fig. 3. Thus the control law becomes:

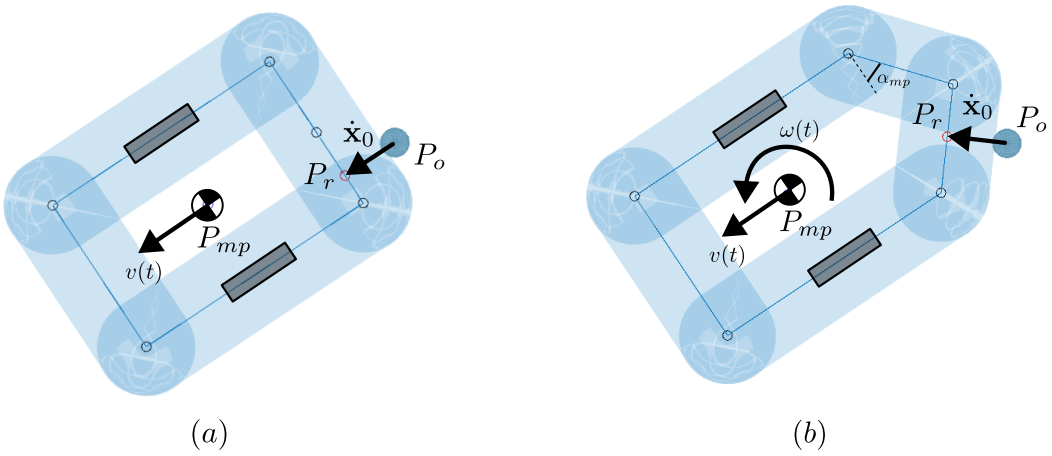
$$\dot{\mathbf{q}} = \mathbf{v}_f + \begin{bmatrix} 0 \\ \dfrac{\dot{x}_0}{p_y\cos\alpha_{mp} - p_x\sin\alpha_{mp}} \end{bmatrix} \tag{7}$$

In this equation, the geometric parameters p_x, p_y, and α_{mp} are illustrated in Fig. 3. To ensure that the platform rotates in the correct direction, the following geometric condition must be satisfied:

$$\alpha_{mp} > \tan^{-1}\left(\frac{D}{L}\right) \tag{8}$$

where D represents the distance between the wheels, L is the length of the robot, and α_{mp} is the angle of the tip. For the specific case study presented in this paper, $L = 890\,\mathrm{mm}$ and $D = 580\,\mathrm{mm}$, yielding a geometric constraint of $\alpha_{mp} > 33^\circ$.

A larger value of the angle α_{mp} results in a greater contribution from the angular velocity, leading to more pronounced rotational movement. However, if the angle becomes too large, the tip of the robot may become excessively large, potentially affecting the robot's ability to navigate efficiently. Consequently, an angle of $\alpha_{mp} = 40^\circ$ was selected for the control model, as it represents a suitable compromise, slightly above the minimum value considered optimal.



(a) (b)

Figure 2 Safety region delimiting the robot's footprint, with 4 segments (**a**) and 5 segments (**b**).

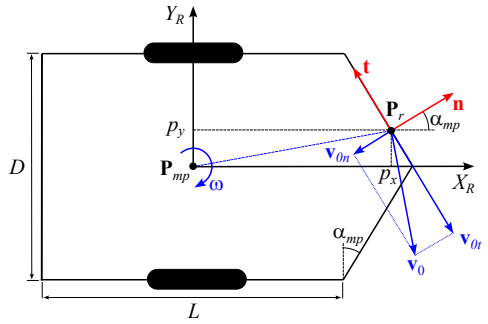


Figure 3 Geometric parameters for Eq. 7

3 RESULTS

Through Matlab simulations, the proposed algorithm was rigorously tested, demonstrating the response of the mobile robot when it encounters an obstacle along its planned trajectory.

To define the safety area around the segments of the robot, the radius r is increased compared to the r_{min} taken as reference. In particular, $r = \frac{4}{3} r_{min}$ and the radius r_{min} is set as a linear function of the velocity v_o of the obstacle, with lower and upper limits ($r_{min} = r_{inf} = 0.17\,\mathrm{m}$ for $v_o \leq v_{inf} = 0.1\,\mathrm{m/s}$, $r_{min} = r_{sup} = 0.23\,\mathrm{m}$ for $v_o \geq v_{sup} = 0.5\,\mathrm{m/s}$). For the examples shown the speed of the obstacle is zero or lower than the imposed limit, so only the minimum safety value will be taken into consideration.

In the experimental tests presented in the next sections, the following two methods will be used for obstacle avoidance:

- **Method 1** (Eq. 5): combines translational and rotational repulsive velocities. This method is suitable for static or semi-static environments, where the robot may need to maintain a specific position while adjusting its pose to avoid obstacles.
- **Method 2** (Eq. 7): applies only a repulsive rotational velocity to avoid obstacles. This approach ensures stable and smooth maneuvers, making it ideal for dynamic environments where the platform is continuously moving.

3.1 TEST CASE 1

In the simulated scenario, a stationary obstacle is positioned along the robot's trajectory. The point-to-point motion planning method facilitates the generation of a trajectory that respects the constraints inherent to a mobile robot with differential wheels, allowing the robot to reach the target position with the desired orientation. For this specific simulation, a curvature parameter of $k = 5\,\mathrm{m}$ was selected to ensure an adequate radius of curvature.

The figure 4 illustrates six frames ranging from $t = 0$ to $t = 15\,\mathrm{s}$, depicting the movement of the robot. The obstacle is represented as a circle. When the robot's influence region intersects with the obstacle, the algorithm kicks in, imposing a repulsive velocity term $\dot{x}_0 = 10\,\mathrm{m/s}$. Each frame includes the planned trajectory and the actual trajectory derived from the collision check based on Eq. 5, while the solid line represents the trajectory obtained using the control strategy defined by Eq. 7.

Once the robot overcomes the obstacle, the positioning error is incrementally corrected through a proportional correction term. The recovery parameters utilized in the simulation are $k_x = 9\,\mathrm{s}^{-1}$, $k_y = 9\,\mathrm{rad/m^2}$ and $k_\gamma = 5\,\mathrm{rad/s}$.

In Fig. 4, the point $\mathbf{P}_r$ closest to the obstacle is marked with a small red circle on the respective segment. In the images, a small blue circle indicates the point on the planned trajectory that the mobile robot must reach.

In the frames shown, it is possible to observe how, in the sections where the obstacle is not present or does not interfere with the robot's safety zone, the robot manages

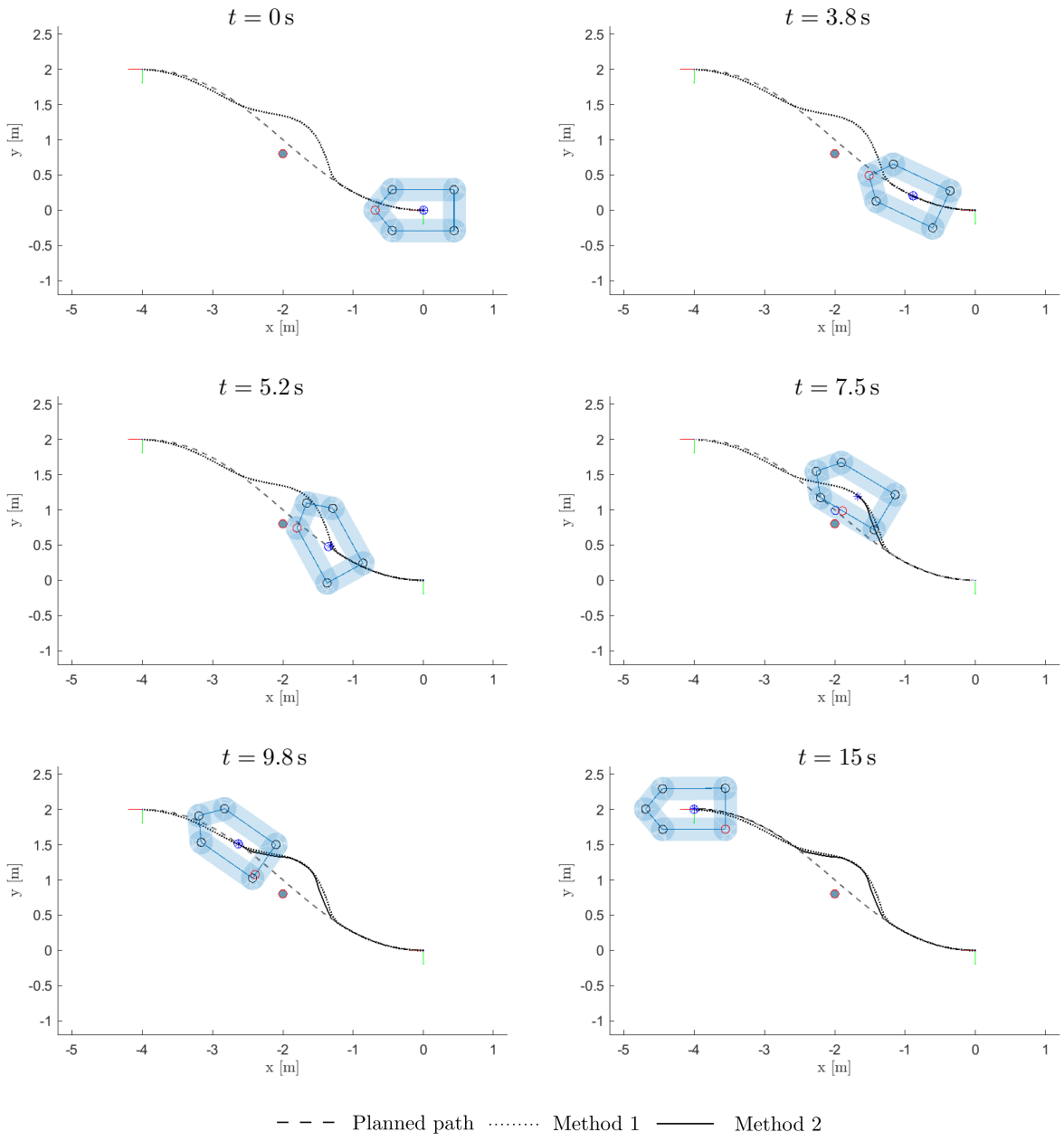


Figure 4 Avoidance of a fixed obstacle while the mobile robot moves along a path.

to faithfully follow the desired trajectory. Under these conditions, the mobile robot maintains an orientation tangent to the trajectory at every point, ensuring smooth movement in accordance with the initial planning.
However, when the distance between the obstacle and the robot's closest point reaches a predetermined limit value, the control system activates a repulsive velocity term that induces a deviation from the planned trajectory. This behaviour is essential to avoid a collision, allowing the robot to dynamically modify its path in response to the presence of the obstacle. It is important to emphasise that the robot's speed is generated moment by moment based on the information available in real time, and not through a predefined alternative path. This allows the system to continuously adapt to its surroundings, reacting immediately to any changes in the position of obstacles.
The effect of repulsive velocity manifests itself in a temporary alteration of the robot's motion, which deviates from its original trajectory in order to safely circumvent the obstacle. Once the obstacle has been overcome, the control system intervenes again to progressively correct the positioning error accumulated during the avoidance manoeuvre. This correction takes place through a proportional action, which gradually returns the robot to its initially planned trajectory, ensuring that the destination is reached with the correct final orientation.
The analysis of the frames makes it possible to appreciate the robot's behaviour in the different phases of the trajectory: initially, the movement occurs without interference, then the deviation induced by the repulsive velocity is observed, and finally the recovery phase that brings the robot back to the desired trajectory.
The comparative analysis of the planned and updated trajectories is presented in Fig. 5, while Fig. 6 illustrates the velocity components v and ω in the local reference frame of the robot.
Between $t = 5\,\mathrm{s}$ to $t = 10\,\mathrm{s}$, near the obstacle, a deviation of the trajectory from the planned path is observed.
Furthermore, Fig. 6a illustrates the velocity components generated by Eq. 5. In this context, a negative translational component emerges for the repulsive segment, resulting in a total velocity v that occasionally becomes negative. In contrast, Figure 6b shows the velocity derived from Eq. 7, in which only the rotational velocity is used for the repulsive action.
In both cases, the trends of the velocity components v and ω aim to closely follow the expected trajectory, indicated with a dotted line in the absence of obstacles.
From the analysis of the two methods, it is observed that method 2, based exclusively on rotation, allows to bypass and subsequently recover the trajectory more quickly than method 1. This behavior is due to the fact that, in method 1, when the robot reaches the minimum safety distance from the obstacle, a repulsive velocity is generated which, in some

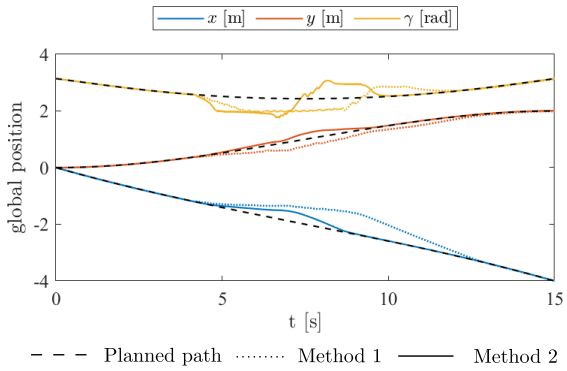


Figure 5 Test case 1: comparison between the planned path and the trajectories obtained using Eq.5 and Eq.7.

instants, induces a backward motion with a negative local translational component. This temporary inversion of motion leads to an increase in the time needed to recover the planned trajectory.
The graph in Fig. 7 shows the trend of the minimum distance between the robot and the obstacle for the two methods analyzed. In the interval between 5 and 10 seconds, the robot approaches the obstacle until it reaches a minimum distance. It can be observed that method 1 allows to maintain a slightly greater distance than method 2, avoiding to significantly fall below the safety margin.
The areas highlighted in the graph indicate critical safety thresholds: the gray band represents the safety margin within which the robot can operate without high risks, while the red zone identifies the minimum limit, beyond which the probability of collision increases considerably. Compared to method 2, method 1 shows a tendency to reach lower distances, suggesting a greater difficulty in ensuring a safe trajectory.
Despite that, method 2 proves more effective in avoiding the obstacle, while still limiting excessive approach and allowing for a quicker recovery of the desired trajectory.

3.2 TEST CASE 2

In the second test case analyzed, a moving obstacle interferes with a stationary robot. The robot is initially in a fix position, but when the obstacle approaches and enters its influence region, it temporarily moves to avoid a collision. Once the obstacle has passed, the robot returns to its original position. Figures 8 and 9 illustrate three frames for each method, from $t = 0$ to $t = 15\,\mathrm{s}$, depicting the movement of the obstacle and the robot's reaction. The obstacle, represented as a circle, moves along a predefined linear path. When the obstacle enters the robot's influence region, a repulsive velocity term $\dot{x}_0 = 10\,\mathrm{m/s}$ is imposed.
Once the obstacle moves past the robot, a proportional correction term incrementally adjusts the positioning error.

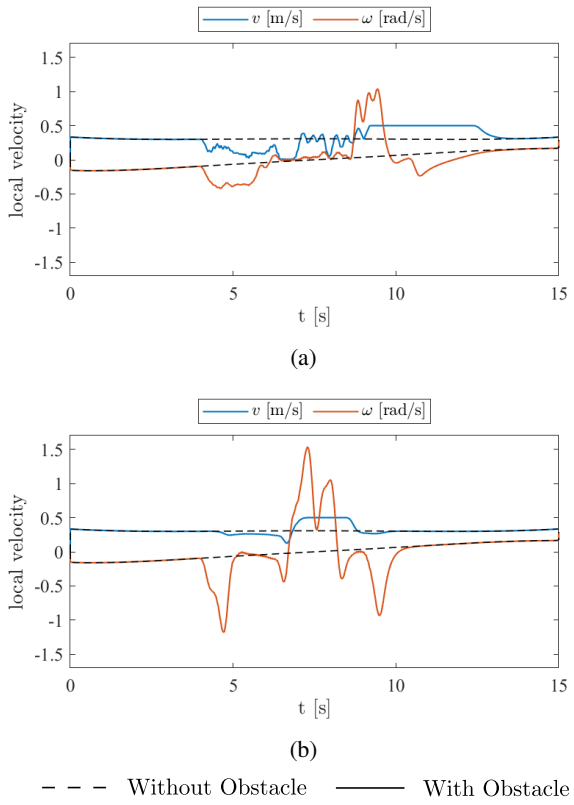


Figure 6 Test case 1: local velocities using Eq.5 (a) and Eq.7 (b).

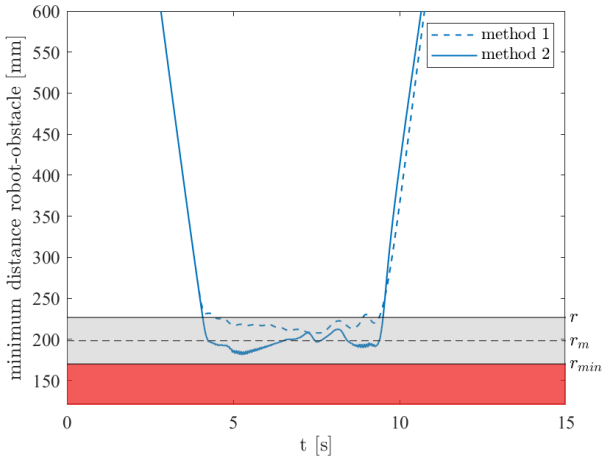


Figure 7 Test case 1: evolution of the minimum distance between the robot and the obstacle for the two methods considered, using Eq.5 and Eq.7.

The recovery parameters utilized in the simulation are $k_x = 1\,\mathrm{s}^{-1}$, $k_y = 1\,\mathrm{rad/m}^2$, and $k_\gamma = 2\,\mathrm{rad/s}$.

Figures 8 illustrate the test case frames using method 1, which avoids obstacles by allowing the robot to move backwards, exploiting the translational repulsive component of Eq.5. In contrast, Figure 9 depicts the scenario with method 2, where the mobile robot relies exclusively on rotational motion to avoid the obstacle with Eq.7. This constraint limits the effectiveness of the algorithm, making it unable to avoid obstacles that pass close to the center of the robot base. In this specific case, where the robot is in the desired position from the beginning, method 2 proves to be less effective, while method 1 provides a more efficient and reliable avoidance maneuver.

This scenario can occur when the robot is positioned in a desired work position and an operator needs to perform a task in the same area or wants to manually move the robot away. In such cases, the robot temporarily moves to avoid interference. Once the operator has left the workspace, the robot autonomously returns to its original position without requiring explicit commands or external communication. This capability enhances flexibility in collaborative environments, ensuring smooth human-robot interaction without disrupting workflow.

Figure 10 illustrates the comparison between the planned path and the actual trajectories obtained using the two different avoidance methods. The graph represents the global position of the robot over time, with the x-axis denoting time in seconds and the y-axis showing the position components x (blue), y (orange), and orientation γ (yellow). The dashed black line represents the planned trajectory, which the robot should ideally follow in the absence of obstacles. The dotted line corresponds to the trajectory obtained using method 1 (Eq.5), which allows for translational repulsion, including backward motion when necessary. Conversely, the solid black line depicts the trajectory using method 2 (Eq.7), which relies solely on rotational adjustments for avoidance.

From the graph, it is evident that during the interaction phase (between $t = 3$ s and $t = 12$ s), both methods result in deviations from the planned path due to the obstacle interference. However, the nature of these deviations differs significantly: method 1, benefiting from translational adjustments, enables the robot to maintain a trajectory closer to the planned one. method 2, limited to rotational avoidance, exhibits a larger deviation, especially when the obstacle passes near the robot's center, making it difficult to maintain a stable avoidance maneuver. After the obstacle exits the interaction region, both methods allow the robot to return to its original trajectory, with method 1 achieving a smoother and quicker recovery.

Figure 11 presents the local velocity profiles of the robot during test case 2, comparing the performance of the two obstacle avoidance methods. The plots illustrate the linear velocity v (blue) and the angular velocity ω (orange) over

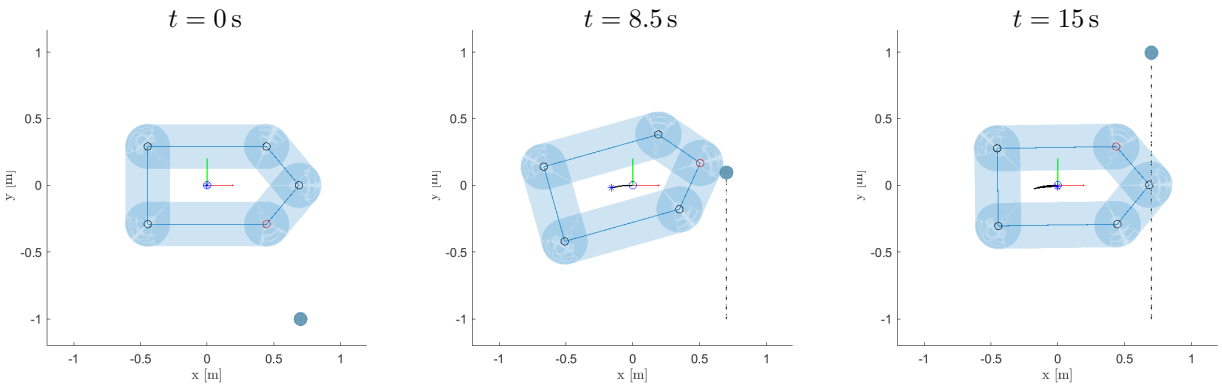


Figure 8 Avoidance of a dynamic obstacle that interferes with the mobile robot in a stationary position using Eq.5.

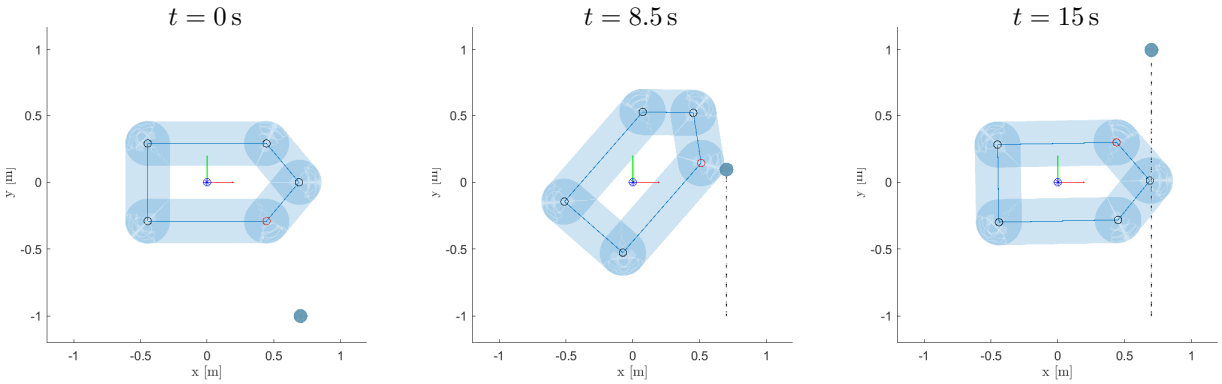


Figure 9 Avoidance of a dynamic obstacle that interferes with the mobile robot in a stationary position using Eq.7.

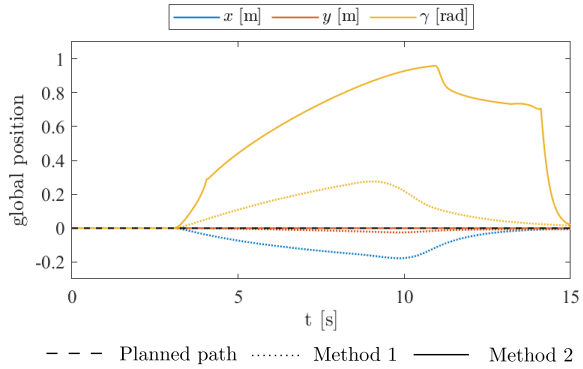


Figure 10 Test case 2: comparison between the planned path and the trajectories obtained using Eq.5 and Eq.7.

time. Subfigure 11a corresponds to the method based on Eq.5, which allows for translational repulsion, while subfigure 11b represents the method using Eq.7, which relies solely on rotational avoidance.

In subfigure (a), the velocity profile remains smooth throughout the trajectory, with minor deviations in both linear and angular velocities when the robot interacts with the obstacle (around $t = 3$ s). Since this method permits backward motion, the adjustments in velocity remain limited, ensuring a stable and efficient avoidance maneuver.

Conversely, in subfigure (b), the avoidance method relying only on rotational motion exhibits significant fluctuations in angular velocity ω. When the obstacle interferes with the robot's path, large spikes in ω indicate abrupt rotational adjustments to maintain a collision-free trajectory. This behavior results in less stable motion.

These results further highlight the advantages of incorporating translational motion for obstacle avoidance, as it allows the robot to maintain smoother and more controlled velocity profiles while ensuring effective maneuverability in dynamic environments.

Figure 12 illustrates the evolution of the minimum distance between the robot and the obstacle, comparing the two avoidance methods. The dashed line represents the distance profile for the method 1, which allows translational motion,

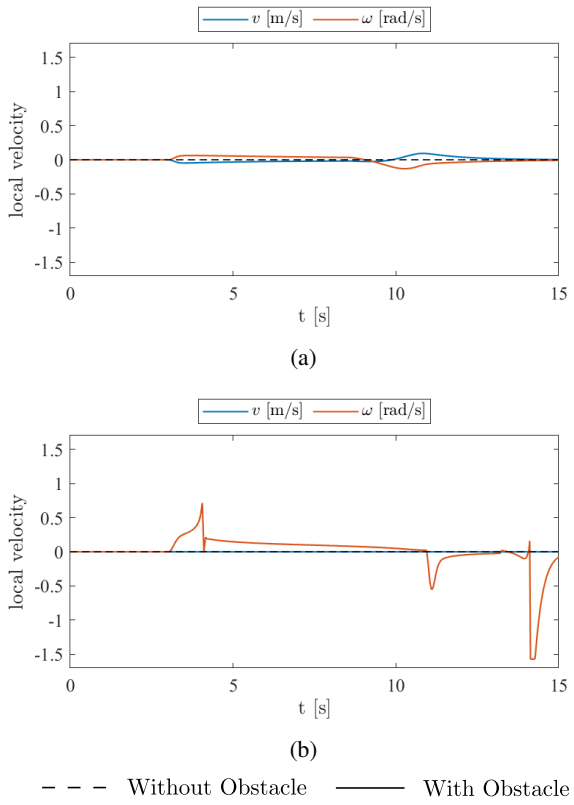


Figure 11 Test case 2: local velocities using Eq.5 (a) and Eq.7 (b).

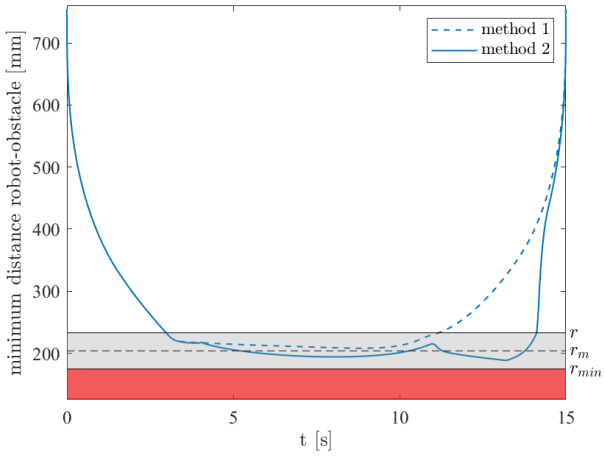


Figure 12 Test case 2: evolution of the minimum distance between the robot and the obstacle for the two methods considered, using Eq.5 and Eq.7.

while the solid line corresponds to the method 2, which relies solely on rotational avoidance.

The graph shows that, for both methods, the minimum distance between the robot and the obstacle decreases as the obstacle approaches. However, a significant difference can be observed in how each method handles proximity constraints. The method allowing translational motion (dashed line) maintains a greater separation from the obstacle throughout the interaction, ensuring a safer avoidance maneuver. Conversely, the purely rotational method (solid line) results in a much closer interaction, particularly around the minimum distance threshold. Both methods detect the obstacle at the same time, at $2.9\,\mathrm{s}$, but diverge in the times for exiting the safety beam: method 1 exits at $11.1\,\mathrm{s}$, with an overall duration of $8.2\,\mathrm{s}$, while method 2 exits at $14.1\,\mathrm{s}$, with an overall duration of $11.2\,\mathrm{s}$.

The shaded regions in the plot highlight critical safety zones: the red area represents distances below the minimum safety threshold r_{min}, where collision risk is high, while the gray region corresponds to a cautionary zone near the safety limit. The rotational method brings the robot close to the obstacle, whereas the translational method consistently maintains a safer distance.

4 DISCUSSION OF RESULTS

Two different methods are employed to enable obstacle avoidance. The first method allows the robot to move both translationally and rotationally, granting greater flexibility in avoiding obstacles, including the ability to temporarily move backward if necessary. The second method relies solely on rotational maneuvers, which can be effective in certain scenarios in which the robot has to move along a path where obstacles are present.

The obstacles considered in this study include both static objects unintentionally left within the workspace and dynamic elements, such as human operators moving within the shared area. The proposed algorithm is not constrained by the type of obstacle detection system used, as long as it can accurately provide the spatial coordinates of the detected object.

The comparative evaluation between the two avoidance strategies revealed significant performance differences. method 1, based on a classical reactive approach, showed stable behaviour, but with some limitations, especially in scenarios where speed of adaptation is crucial. In particular, its ability to avoid obstacles is less smooth, with more marked deviations from the desired path.

In contrast, method 2 demonstrated greater efficiency, allowing more progressive and smooth adjustments of the trajectory, reducing sudden deviations and improving continuity of motion. This approach proved to be more robust and flexible, being particularly suitable for contexts

in which the robot must follow a predetermined path in the presence of fixed or dynamic obstacles.
Test case 2 analysed the behaviour of the robot in a situation where it is already in the desired position and a moving obstacle is approaching. In this scenario, method 1 demonstrated superior performance compared to method 2, as it allows the robot to make a temporary deviation from its position without losing its main objective.
The analysis of local velocities shows that method 1, based on reactive control, allows the robot to move in a rapid and controlled manner to avoid interference with the obstacle, and then autonomously return to its initial position once the area is clear. This behaviour is particularly advantageous in collaborative applications, where human operators may have to work in the same area as the robot or move it temporarily without the need for explicit commands.
On the other hand, method 2, designed to generate more fluid and continuous trajectories, proved less effective in this specific context. Since the robot does not have to follow a predefined path, but simply maintain the desired position while avoiding collisions.
One limitation of the proposed approach is that it does not explicitly account for the relative velocity between the robot and the obstacle, which may lead to unnecessary deviations or delayed reactions depending on the scenario. A possible enhancement involves dynamically adjusting the safety radius based on the approaching speed: if the relative velocity is high, the safety region expands, activating the repulsive velocity earlier. This approach helps prevent excessive reactions to slow-moving obstacles while ensuring a timely response to fast-approaching ones.

5 CONCLUSIONS

This study presented a new approach for obstacle avoidance in non-holonomic mobile robots equipped with differential transmission. The proposed methodology was tested in complex dynamic environments, demonstrating its ability to improve manoeuvrability and reduce the risk of sudden stops, while maintaining real-time response. The simulations presented in the previous section are aimed at verifying the correctness of the algorithm. In fact, the final objective of this research is to transfer the control framework to a real system, currently under construction in the laboratory, where a vision system composed of three Intel RealSense depth cameras is able to capture the presence of people or objects within the work space [27].
A particularly relevant scenario arises when the robot is stationed in a designated work position, and an operator needs to perform a task in the same area or manually reposition the robot. In such cases, the robot temporarily relocates to prevent interference, ensuring seamless human-robot collaboration. Once the operator has vacated the workspace, the robot autonomously returns to its original position without requiring explicit commands or external communication. This adaptive behavior significantly enhances flexibility in collaborative environments, enabling uninterrupted workflows while maintaining operational efficiency and safety.
These results show that the choice of avoidance method depends not only on the nature of the obstacles (static or dynamic), but also on the specific task of the robot. When the main objective is to maintain a position and ensure flexibility in interaction with operators or other dynamic elements of the environment, method 1 is a more effective solution.
From a practical point of view, the integration of these methods in industrial mobile platforms can lead to significant benefits in terms of operational efficiency. The ability to maintain a safe minimum distance by dynamically adjusting the trajectory is particularly relevant for applications involving collaborative robots (cobots), where human-robot interaction is coupled with high-throughput industrial contexts.
A key contribution of this work lies in the integration of an optimised influence zone, which allows the robot to proactively adjust its path before entering critical proximity regions. This innovation reduces the risk of emergency stops and ensures continuous movement, solving one of the main limitations of conventional obstacle avoidance methods. The introduction of the tip into the model further refines trajectory planning in confined environments, demonstrating a clear advantage over conventional circular models, which tend to overestimate safety margins.
The study has some limitations due to the simplifications of the simulation, which does not consider real-life factors such as uneven surfaces, variable friction or errors in the vision sensors. To improve the robustness of the algorithm, future developments could include systems for predicting the trajectory of obstacles through machine learning, integration with advanced sensors and testing on a physical system to refine the model.
In future work, an exploration of the possibility of using variable coefficients for repulsive velocity and recovery parameters to further optimise robot performance is planned. By allowing these coefficients to adapt according to the robot's current state, environment and task requirements, expect to improve both the efficiency of obstacle avoidance and the accuracy of point-to-point trajectory tracking.
Future research will focus on extending the proposed algorithm to multi-robot coordination scenarios, where avoidance strategies will have to consider not only static and dynamic obstacles, but also the interaction between multiple agents operating in the same workspace. Furthermore, an experimental validation will be conducted on a fully autonomous mobile manipulator, combining a 6 degrees of freedom robotic arm with a differential transmission base. In this way, the additional degrees of freedom of the system can be exploited to improve collision avoidance and confer

the robot an optimal motion in terms of smoothness and dexterity. The integration of advanced perception systems, such as LiDAR and depth cameras, will further improve the accuracy of obstacle detection, enhancing situational awareness and decision-making capabilities.

REFERENCES

[1] Dimitrios Bechtsis, Naoum Tsolakis, Menippos Vouzas, and Dimitrios Vlachos. Industry 4.0: Sustainable material handling processes in industrial environments. In *Computer aided chemical engineering*, volume 40, pages 2281–2286. Elsevier, 2017.

[2] Mohd Nayab Zafar and JC Mohanta. Methodology for path planning and optimization of mobile robots: A review. *Procedia computer science*, 133:141–152, 2018.

[3] D Costa et al. A novel class of bio-inspired underwater robots. *INTERNATIONAL JOURNAL OF MECHANICS AND CONTROL*, 23(2):23–35, 2022.

[4] Luca Carbonari, Luigi Tagliavini, Andrea Botta, Paride Cavallone, and Giuseppe Quaglia. Preliminary observations for functional design of a mobile robotic manipulator. In *International Conference on Robotics in Alpe-Adria Danube Region*, pages 39–46. Springer, 2021.

[5] Carmen Visconte, Paride Cavallone, Luca Carbonari, Andrea Botta, and Giuseppe Quaglia. Design of a mechanism with embedded suspension to reconfigure the agri_q locomotion layout. *Robotics*, 10(1):15, 2021.

[6] Piotr Skrzypczyński. Mobile robot localization: where we are and what are the challenges? *Automation 2017: Innovations in Automation, Robotics and Measurement Techniques 1*, pages 249–267, 2017.

[7] Federico Neri, Luca Alfredo Annoni, Giacomo Palmieri, Matteo-Claudio Palpacelli, and Massimo Callegari. Localisation of mobile robots via ultra-wideband systems. In *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, volume 87356, page V007T07A008. American Society of Mechanical Engineers, 2023.

[8] Bo Tao, Xingwei Zhao, Sijie Yan, and Han Ding. Kinematic modeling and control of mobile robot for large-scale workpiece machining. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 236(1-2):29–38, 2022.

[9] Oussama Khatib. Real-time obstacle avoidance for manipulators and mobile robots. *The international journal of robotics research*, 5(1):90–98, 1986.

[10] Gregor Klancar, Drago Matko, and Saso Blazic. Mobile robot control on a reference path. In *Proceedings of the 2005 IEEE International Symposium on, Mediterrean Conference on Control and Automation Intelligent Control, 2005.*, pages 1343–1348. IEEE, 2005.

[11] Lixing Liu, Xu Wang, Xin Yang, Hongjie Liu, Jianping Li, and Pengfei Wang. Path planning techniques for mobile robots: Review and prospect. *Expert Systems with Applications*, 227:120254, 2023.

[12] Antonio Sgorbissa and Renato Zaccaria. Planning and obstacle avoidance in mobile robotics. *Robotics and Autonomous Systems*, 60(4):628–638, 2012.

[13] Yunchao Tang, Shaojun Qi, Lixue Zhu, Xianrong Zhuo, Yunqi Zhang, and Fan Meng. Obstacle avoidance motion in mobile robotics. *Journal of System Simulation*, 36(1):1–26, 2024.

[14] Anish Pandey, Shalini Pandey, and D Parhi. Mobile robot navigation and obstacle avoidance techniques: A review. *Int Rob Auto J*, 2(3):00022, 2017.

[15] Shuzhi Sam Ge and Yun J Cui. Dynamic motion planning for mobile robots using potential field method. *Autonomous robots*, 13:207–222, 2002.

[16] Rocco Galati, Giacomo Mantriota, and Giulio Reina. An eco-friendly and sustainable route planning for a floor-marking omnidirectional robot. *International Journal of Mechanics and Control*, 23(2):73 – 83, 2022. Cited by: 2.

[17] Johann Borenstein and Yoram Koren. Real-time obstacle avoidance for fast mobile robots. *IEEE Transactions on systems, Man, and Cybernetics*, 19(5):1179–1187, 1989.

[18] Weihao Li, Chenguang Yang, Yiming Jiang, Xiaofeng Liu, Chun-Yi Su, et al. Motion planning for omnidirectional wheeled mobile robot by potential field method. *Journal of Advanced Transportation*, 2017, 2017.

[19] Patrick F Muir and Charles P Neuman. Kinematic modeling for feedback control of an omnidirectional wheeled mobile robot. *Autonomous robot vehicles*, pages 25–31, 1990.

[20] Sheha Ame Mnubi. Motion planning and trajectory for wheeled mobile robot. *International Journal of Science and Research*, 5(1):1064–1068, 2016.

[21] Soonshin Han, ByoungSuk Choi, and JangMyung Lee. A precise curved motion planning for a differential driving mobile robot. *Mechatronics*, 18(9):486–494, 2008.

[22] Martín Velasco-Villa, Raúl Dalí Cruz-Morales, Alejandro Rodriguez-Angeles, and Carlos A Domínguez-Ortega. Observer-based time-variant spacing policy for a platoon of non-holonomic mobile robots. *Sensors*, 21(11):3824, 2021.

[23] Ksenia Shabalina, Artur Sagitov, and Evgeni Magid. Comparative analysis of mobile robot wheels design. In *2018 11th International Conference on Developments in esystems Engineering (dese)*, pages 175–179. IEEE, 2018.

[24] Federico Neri, Matteo Forlini, Cecilia Scoccia, Giacomo Palmieri, and Massimo Callegari. Experimental evaluation of collision avoidance techniques for collaborative robots. *Applied Sciences*, 13(5):2944, 2023.

[25] Elie Maalouf, Maarouf Saad, and Hamadou Saliah. A higher level path tracking controller for a four-wheel differentially steered mobile robot. *Robotics and autonomous systems*, 54(1):23–33, 2006.

[26] Yutaka Kanayama, Yoshihiko Kimura, Fumio Miyazaki, and Tetsuo Noguchi. A stable tracking control method for an autonomous mobile robot. In *Proceedings., IEEE International Conference on Robotics and Automation*, pages 384–389. IEEE, 1990.

[27] Matteo Forlini, Federico Neri, Cecilia Scoccia, Luca Carbonari, and Giacomo Palmieri. Collision avoidance in collaborative robotics based on real-time skeleton tracking. In *International Conference on Robotics in Alpe-Adria Danube Region*, pages 81–88. Springer, 2023.