




Article

Non-Holonomic Mobile Manipulator Obstacle Avoidance with Adaptive Prioritization

Federico Neri , Giacomo Palmieri *  and Massimo Callegari 

DIISM—Department of Industrial Engineering and Mathematical Sciences, Polytechnic University of Marche, 60131 Ancona, Italy; federico.neri@staff.univpm.it (F.N.); m.callegari@univpm.it (M.C.)

* Correspondence: g.palmieri@staff.univpm.it

Abstract: This paper presents an obstacle avoidance strategy for mobile manipulators consisting of a robotic arm and a base with a non-holonomic differential wheel system. The algorithm makes it possible to avoid obstacles in a dynamic environment, without planning the path a priori. A series of examples are proposed in simulation using Matlab and analyzed to show how the algorithm works if the obstacle interferes with the manipulator or the base. In addition, the possibility of prioritizing the movement of certain parts of the system using the weighted pseudo-inverse matrix is introduced. In this way, it is possible to give movement priority to the base if it is necessary to move the robot over long distances while keeping the manipulator as still as possible. The use of null space to keep the end-effector stationary while it avoids obstacles is also explored, exploiting the system's redundancy and allowing the rest of the kinematic chain and the mobile base to move accordingly. Finally, current standards are analyzed and a solution is shown that allows the robot to vary its behavior to avoid obstacles depending on the distance to the target point.

Keywords: collaborative robotics; obstacle avoidance; human–robot interaction; weighted pseudoinverse matrix; mobile manipulator



Academic Editors: Mónica Ballesta, Oscar Reinoso García and María Flores Tenza

Received: 16 January 2025

Revised: 2 April 2025

Accepted: 11 April 2025

Published: 18 April 2025

Citation: Neri, F.; Palmieri, G.; Callegari, M. Non-Holonomic Mobile Manipulator Obstacle Avoidance with Adaptive Prioritization. *Robotics* **2025**, *14*, 52. <https://doi.org/10.3390/robotics14040052>

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The advent of collaborative robotics allows operators and robots to coexist harmoniously in shared workspaces without the presence of physical barriers. This innovative technology not only greatly improves working conditions in industry, but also paves the way for a nascent research frontier: obstacle avoidance in a dynamic environment [1,2].

In scenarios where the cycle time of specific tasks requires further reduction [3,4], the integration of obstacle avoidance strategies allows the robot to momentarily deviate from predetermined trajectories to avoid potential collisions, thus maintaining uninterrupted robot operation [5]. This solution is particularly advantageous in pick-and-place operations, where, when it detects obstacles in its path during transitions between zones, the robot can autonomously navigate around the obstructions without stopping. Furthermore, if an operator needs to intervene while the robot is operating within a work zone, the approach causes the robot to leave the area. At the end of the intervention, the robot smoothly resumes its task, automatically returning to the designated work zone when the presence of the operator is no longer detected. This solution reduces the cycle time while allowing the operator to work safely. In addition, it can still be supplemented with algorithms that stop the robot if conditions do not allow for safe avoidance [6].

The size of warehouses and the need to handle materials as flexibly as possible necessitated the development of less bulky solutions than conveyors. For this reason,

first AGVs (Automated Guided Vehicles) and then the more modern AMRs (Autonomous Mobile Robots) were developed. Unlike manipulators, which possess the dexterity to maneuver over all degrees of freedom in space, mobile robots are constrained to move in the plane. However, they compensate for this limitation by displaying the ability to traverse extended distances efficiently. [7,8]

Both AMRs and AGVs equipped with manipulators, utilized within test environments, fall within the scope of the IMR (Integrated Mobile Robot) definition and are consequently subject to compliance with relevant standards. However, AGVs and AMRs serve distinct purposes: the former adhere to predetermined paths, while the latter leverage sensors to autonomously navigate around obstacles by computing their trajectories.

Mobile manipulators, which integrate both mobility and handling capabilities, are increasingly adopted in industry due to their flexibility and ability to perform complex tasks in dynamic environments. This growing adoption highlights the importance of a clear regulatory framework, such as the ANSI/RIA R15.08 standard [9,10], which classifies these systems in the IMR category and provides guidelines for their safe operation.

The ANSI/RIA R15.08 standard draws upon pertinent guidance provided by ANSI/RIA R15.06 (2012) and ANSI/ITSDF B56.5 (2019), which focus on the safety of industrial robots and guided industrial vehicles, respectively [11]. In particular, as can be seen from the diagram in Figure 1, this standard classifies mobile manipulators consisting of a mobile base and an arm as Type C IMRs.

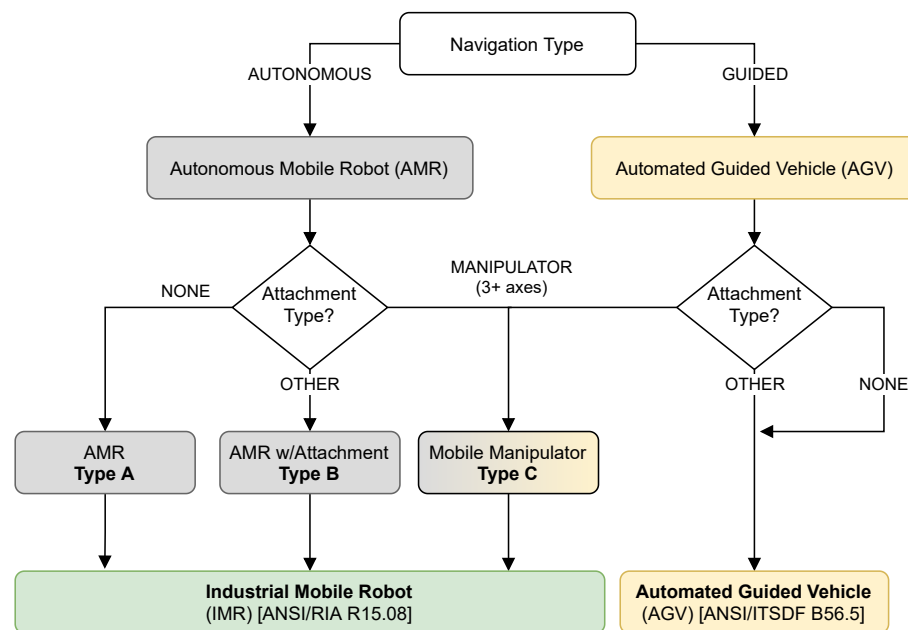


Figure 1. Classification of industrial mobile robots—ANSI/RIA R15.08.

Although there are pre-assembled mobile manipulators available in the market, to ensure optimal performance tailored to specific applications, it is common to separately acquire the base and manipulator and then integrate them. The selection of the base is pivotal, considering that various types and architectures can be employed in mobile robots, thereby influencing their kinematic model. Tagliavini et al. [12] delineate various mechanisms facilitating the locomotion of mobile robots.

The two main macrocategories are omnidirectional and differential wheel systems. Omnidirectional platforms allow the vehicle to move in any direction [13,14] and can be, for example, a four mecanum wheels robot [15], a three omni wheel platform and a two swerve drive system.

Mobile robots with differential wheels are easier to construct and are commonly used in various industrial contexts. However, they have movement limitations. Although they can reach any configuration on the work surface, they cannot control speed in all directions. In essence, this system is non-holonomic, i.e., it can only set the speed along the longitudinal and rotational directions, without the ability to move in the transverse direction [16–18].

The use of a mobile manipulator allows for greater flexibility during task execution, taking advantage of redundancy to choose the best configuration in order to reach a certain end-effector pose while avoiding any obstacles in the workspace. In addition to this, the presence of a mobile base allows the robot to be moved in a large working environment, increasing the limited reachability that the manipulator alone would have.

There are numerous papers describing methodologies for analyzing the safety of manipulators and mobile robots, complying with currently existing standards. Some of them propose a risk assessment for the safe use of mobile robots on construction sites [19], extending the currently existing safety standards to include scenarios not covered. Despite this, it is still unclear what standards must be met when using mobile manipulators.

Some of the research that led to the development of certain strategies was formulated on the basis of competitions, simulations or experiments conducted in the laboratory. However, real industrial environments are more complex and involve more people and objectives. As is well known, the real industrial environment is rather difficult to simulate and replicate. In the next few years, many advanced technologies can be tested more easily in real environments, allowing for existing regulations to be improved [20].

The evolution of robotic systems leads to new questions concerning safety in the industrial world. As described in [21], for mobile manipulators, which combine a movable platform with a robotic arm, the application of different standards is contingent upon the specific function of the manipulator:

1. When the AMR or the AGV is in motion and the robot arm is at rest, the latter can be treated as a load, and the risk assessment should be conducted in accordance with standard mobile robot protocols.
2. When the robotic arm performs assembly or processing tasks while in motion, all relevant robotics standards must be applied during the risk assessment process for the robotic arm.

Each of these possibilities must take into account the various safety issues associated with them. First of all, it must be ensured that the mobile robot is located within the work area, and does not risk becoming stuck or losing its reference system. This can be achieved using odometry [22,23], proximity sensors, visual markers [24,25], GPS sensors [26–28] or other methods [29,30] to create a map and update the robot's position during movement. Panigrahi and Bisoy present a complete review of localization systems, problems, principles and approaches for mobile robots [31].

Marvel and Bostelman [32] present different scenarios that can occur when a mobile manipulator is placed in a collaborative environment where a human is present. Such situations are still unclear and not fully analyzed by security standards. For example, the Cartesian velocities of all joints, the robot and mobile base being separate, must be known at all times. Only in this way can the actual speed of the various parts of the robotic system be analyzed with respect to a reference system that considers the combination of the manipulator and base speeds, allowing for the risk to be analyzed.

In addition, it should be analyzed whether the angle of approach of the human is different from the direction of travel of the mobile robot. Three cases can occur:

1. Human is in the robot's working volume, in the path of the AMR.
2. Human is outside the robot's working volume, in the path of the AMR.
3. Human is in the robot's working volume, out of the AMR's path.

In addition to this, situations related to resuming the system after a lockout can also create problems. When resuming from a paused or stopped state, restarting the movement of a mobile manipulator could trigger an emergency response from the arm if a dual controller is present.

In recent years, many strategies for manipulators have been proposed, such as the design of a switching controller using a sliding mode approach based on neural networks [33], and there are numerous methods that can be used to perceive the work space and allow a mobile manipulator to move in space. For example, the simplest method consists of moving the base and arm separately, using the mobile robot to reach a target or near an object to be manipulated, and then moving the manipulator while holding the base still [34,35]. The review by T. Sandakalum and MH. Ang Jr [36] shows an example of separate movements. Although this approach is well-suited to controlled environments, unexpected movements of the robot and objects can cause failures in complex contexts. Although this approach is well-suited to controlled environments, unexpected movements of the robot and objects can cause failures in complex contexts. In contrast, integrated systems aim to combine locomotion and manipulation control in a single control architecture. This approach allows for more consistent motion planning, where the robotic arm and the mobile platform can adapt to each other in real time [37].

In order to improve the control of the system and make the best use of redundancy, various solutions can be found in the literature that treat the manipulator and the base with a single control, considering both to generate paths [38,39]. However, methods that allow for collision-free path planning have a considerable computation time. Consequently, these systems are not suitable for highly dynamic environments, which require re-planning.

Recent work has made it possible to formulate strategies for controlling mobile manipulators by exploiting contact; for example, M. Weyrer et al. [40] present an approach for the manual guidance of a sensitive mobile manipulator in the workspace, using a force and torque sensor mounted close to the end-effector.

Other studies involving mobile manipulators have made it possible to develop dynamic obstacle avoidance algorithms, exploiting redundancy to perform tasks without colliding with objects or humans in the work area [41–43]. Unfortunately, such studies mainly focus on systems using an omnidirectional mobile base, without analyzing the problems that a non-holonomic system might have. Studies analyzing mobile bases with differential wheels do not consider the possibility of dynamically adapting the priority of movement between the manipulator and the base according to the task performed. Furthermore, they do not explore the use of null space to avoid obstacles while keeping the end-effector in a fixed position [44].

In Section 2, the strategy proposed in this paper takes the obstacle avoidance algorithm proposed by the authors in [45] and extends it to a mobile manipulator, making the necessary modifications to exploit the system's redundancy. This strategy allows the collaborative robot to perform a given task where humans are present, momentarily deviating from the trajectory if an obstacle is perceived. Through the presence of the mobile base, it is possible to move material or perform tasks in distant work areas. The influence region of each part of the robotic system is defined by a radius r which can be set according to the required safety level and can vary according to the relative speed between the obstacle and the manipulator.

The strategy described in this paper considers a mobile manipulator consisting of a 6 DOF robotic arm and a moving base with differential wheels. This solution makes it possible to study one of the most common combinations in industry, which presents various kinematic complexities. Despite this, the algorithm is presented in a generic form

and can easily be extended to a system with a 7-axis arm or other mobile manipulators with high redundancy [46].

Different from other studies, the strategy proposed by the authors allows for the collision to be avoided dynamically, without the need to come into contact with the object. In addition to this, the behavior of the moving base with differential wheels is analyzed in detail and a solution is proposed that allows for constraints to be respected in order to execute a task and simultaneously avoid any obstacles. The proposed solution also makes it possible to use null space to handle situations in which it is desired to keep the end-effector stationary, in the event that a human interferes with the base.

The proposed strategy introduces several innovations compared to previous work in the field of mobile manipulators. Unlike traditional approaches that often treat the manipulator and the mobile base as separate entities [36,47], this method integrates control of both to dynamically exploit redundancy, enabling obstacle avoidance without requiring physical contact. This distinguishes it from other work, which either focused on contact-based strategies or was limited to systems with omnidirectional bases, neglecting the challenges imposed by non-holonomic platforms.

Moreover, by employing a strategy based on the weighted pseudo-inverse matrix, the system can offer varying levels of flexibility. For instance, in certain scenarios, it is feasible to prioritize the movement of the mobile base by limiting the manipulator's speed. Alternatively, the system can mainly rely on the arm for obstacle avoidance by restricting the movement of the mobile base. Additionally, according to the standards, the proposed solution suggests that, when the robot is far from its target, only the mobile base should be used for movement, keeping the arm stationary. As the robot approaches closer to the target, the manipulator is gradually activated, utilizing all available redundancy to avoid obstacles and achieve the final position.

The main objective of this work is the development of a strategy based on the weighted pseudo-inverse matrix, which exploits the redundancy of a mobile manipulator to perform tasks while simultaneously avoiding potential obstacles in the workspace. The progressive activation of the manipulator's speed as the system approaches the target, combined with the optimization in null space, represents a novel element in the balance between task execution and obstacle avoidance.

To complete this study, some simulations are presented and the system behavior is analyzed in detail in Section 3. The results obtained are discussed and the conclusions of the work and future prospects are described in Section 4.

2. Obstacle Avoidance Algorithm

This section describes the obstacle avoidance algorithm for a generic mobile manipulator, which consists of a robotic arm with a number of actuators greater than or equal to 6 (depending on its degree of redundancy) positioned on a mobile base with differential wheels. The vector \mathbf{q} describes the configuration of the mobile manipulator and $\dot{\mathbf{q}} = [v \ \omega \ \dot{q}_1 \ \dot{q}_2 \ \dots \ \dot{q}_n]^T$ is the velocity vector, where the n components correspond to the n degrees of freedom (DOF) of the manipulator. In Cartesian space, according to the Euler angle convention ZYZ, the pose of the end-effector is defined by the vector $\mathbf{x} = [x \ y \ z \ \alpha \ \beta \ \gamma]^T$. In matrix form, the velocity kinematics of such system can be written as

$$\dot{\mathbf{x}} = \mathbf{J}\dot{\mathbf{q}} = \begin{bmatrix} \mathbf{J}_p \\ \mathbf{J}_r \end{bmatrix} \dot{\mathbf{q}}, \quad (1)$$

where \mathbf{J} is the $6 \times m$ Jacobian, where m is the dimension of the vector $\dot{\mathbf{q}}$ ($m \geq 8$), \mathbf{J}_p the submatrix of the position and \mathbf{J}_r of the rotation.

To simplify, it is possible to represent the mobile robot as a cubic structure equipped with two wheels symmetrically positioned on its sides. The distance between these wheels is denoted as D , and their respective radii are represented by R .

The basis $\{X_R, Y_R\}$ defines the axes relative to \mathbf{P}_{mp} on the robot chassis and thus represents the robot's local reference frame. Due to the non-holonomic constraint, the robot cannot move along the Y_R direction, so its velocity can be described in the local frame using the vector $[v, \omega]^T$, where v represents the forward velocity and ω the angular velocity of the robot. Regarding the wheels, the linear speed of the right wheel is given by $v_r = \omega_r R$, while the linear speed of the left wheel is $v_l = \omega_l R$.

By applying a rotation of angle θ_b around the z axis, it is possible to express the velocity vector in the global reference system, as described in Equation (2):

$$\begin{bmatrix} \dot{x}_b \\ \dot{y}_b \\ \dot{\theta}_b \end{bmatrix} = \begin{bmatrix} \cos \theta_b & 0 \\ \sin \theta_b & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix}. \tag{2}$$

Here, $[\dot{x}_b, \dot{y}_b, \dot{\theta}_b]^T$ represents the velocity of the base in the global reference system, with \mathbf{P}_{mp} indicating the position of the central point of the mobile robot. The orientation of the robot with respect to the X_w axis of the global reference frame W is given by θ_b .

The Jacobian matrix that relates the angular velocity of the wheels to the velocity vector in the robot's local reference frame can be derived as follows (Equation (3)):

$$\begin{bmatrix} v \\ \omega \end{bmatrix} = \begin{bmatrix} \frac{R}{2} & \frac{R}{2} \\ -\frac{R}{D} & \frac{R}{D} \end{bmatrix} \begin{bmatrix} \omega_l \\ \omega_r \end{bmatrix}. \tag{3}$$

2.1. Mobile Robot Obstacle Avoidance

The possibility of making the end-effector follow a straight trajectory by utilizing all available degrees of freedom will also be analyzed in this paper. This is relevant for tasks such as welding or painting along a path. However, when the target point is far from the starting position of the mobile manipulator, moving the arm joints is often unnecessary. To reach the desired position, or to get closer to it, it is sufficient to move in the plane with the mobile base while avoiding obstacles, keeping the robot joints fixed. Over long distances, moving the robot arm can be inefficient and potentially dangerous. Once the manipulator reaches an area close enough to the target, defined as a distance that can be set within the algorithm, the manipulator can be activated to achieve the target using all available degrees of freedom. Therefore, a strategy to avoid obstacles using only the moving base is initially analyzed.

Since in a mobile differential wheel robot it is not possible to impose a speed along the Y_w axis, it is necessary to plan the path in such a way that an end point with a desired orientation can be reached. To generate a smooth point-to-point trajectory that adheres to the constraints of a differential wheeled mobile robot, a parametric cubic polynomial approach was employed [48]. Given the initial and final poses of the mobile base, respectively defined as $\mathbf{x}_i = [x_i \ y_i \ \theta_i]^T$ and $\mathbf{x}_f = [x_f \ y_f \ \theta_f]^T$, the resulting trajectory is expressed as

$$\begin{bmatrix} x_b(s) \\ y_b(s) \end{bmatrix} = \begin{bmatrix} -(s-1)^3 x_i + s^3 x_f + \alpha_x s^2 (s-1) + \beta_x s (s-1)^2 \\ -(s-1)^3 y_i + s^3 y_f + \alpha_y s^2 (s-1) + \beta_y s (s-1)^2 \end{bmatrix}, \tag{4}$$

where the parameter $s \in [0, 1]$ evolves over time following a fifth polynomial function. The coefficients $\alpha_x, \alpha_y, \beta_x, \beta_y$ are defined as

$$\begin{bmatrix} \alpha_x \\ \alpha_y \end{bmatrix} = \begin{bmatrix} k \cos \theta_f - 3x_f \\ k \sin \theta_f - 3y_f \end{bmatrix} \quad \begin{bmatrix} \beta_x \\ \beta_y \end{bmatrix} = \begin{bmatrix} k \cos \theta_i + 3x_i \\ k \sin \theta_i + 3y_i \end{bmatrix}. \quad (5)$$

The parameter k , in the equation, is adjustable to modify the radius of curvature of the path. The orientation $\theta_b(s)$ is then computed to ensure that the X_R axis remains tangential to the trajectory at all points. Once the trajectory is generated, a reference velocity profile can be derived as $v_r = \sqrt{\dot{x}_b^2 + \dot{y}_b^2}$ and $\omega_r = \dot{\theta}_b$.

If an obstacle obstructs the robot's movement, the proposed motion control system can dynamically adjust the trajectory in real time. When the platform approaches an obstacle and the separation distance falls below a critical threshold, defined by the safety radius r , a repulsive velocity vector is generated. Consequently, the platform is surrounded by a safety area that takes the shape of a rectangle with rounded corners, each with a common radius r , as shown in Figure 2a. This safety region can be visualized as a vertical extrusion of the safety area. The repulsive velocity vector \dot{x}_0 is applied at the point \mathbf{P}_r , which corresponds to the closest point on the platform's contour to the obstacle's position \mathbf{P}_o (Figure 2). The magnitude of \dot{x}_0 acts as a control parameter, while its direction is determined by the unit vector along the distance $\mathbf{d}_o = \mathbf{P}_r - \mathbf{P}_o$, resulting in $\dot{x}_0 = \dot{x}_0 \hat{\mathbf{d}}_o$.

The kinematic expression of \dot{x}_0 can be formulated as follows, where \mathbf{J}_0 represents the Jacobian matrix associated with point \mathbf{P}_r :

$$\dot{x}_0 = \begin{bmatrix} \cos \theta_b & -p_y \cos \theta_b - p_x \sin \theta_b \\ \sin \theta_b & p_x \cos \theta_b - p_y \sin \theta_b \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} = \mathbf{J}_0 \begin{bmatrix} v \\ \omega \end{bmatrix}, \quad (6)$$

where p_x and p_y represent the coordinates of \mathbf{P}_r in the local reference frame. The algorithm is structured to detect the robot segment with the highest collision risk by measuring the distance from each obstacle to each part of the robot's perimeter. This allows for identifying point \mathbf{P}_r with the shortest distance to the obstacle and computing the distance vector \mathbf{d}_o [49].

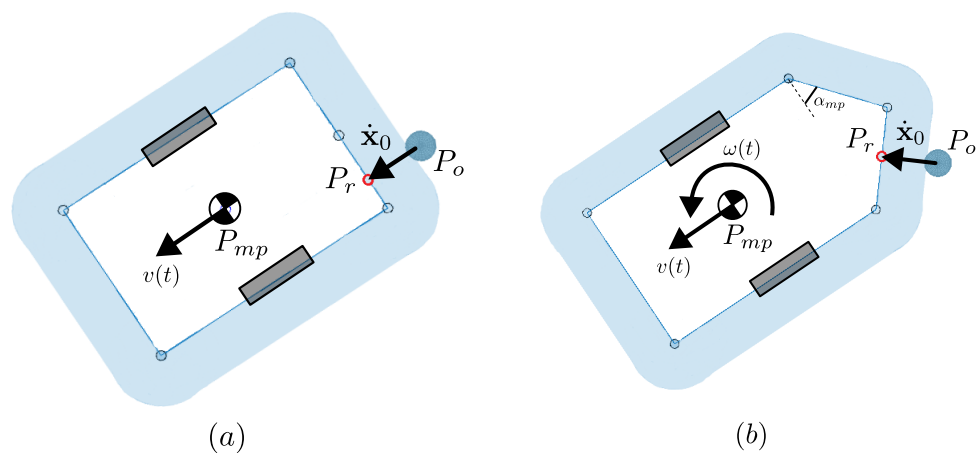


Figure 2. Safety region delimiting the robot's footprint, with 4 segments (a) and 5 segments (b).

As illustrated in Figure 2a, a problem may arise if an obstacle blocks the front edge of the platform. In this case, the control system can generate a repulsive velocity by adjusting only the translation speed, without introducing any rotational movement. In this way, the platform moves backwards along a straight trajectory. Once the obstacle leaves

the region of influence, the system orders the platform to return to its original position, potentially causing a stall. To overcome this limitation, a safety region was introduced consisting of five segments that form a pointed structure at the front of the platform, such as illustrated in Figure 2b. This approach facilitates the computation of the necessary rotational and translational velocities to navigate around the obstacle, as described in previous studies [50]. In particular, this results in a correctly oriented rotational repulsive velocity, i.e., counterclockwise in the example in the figure. If, for a generic obstacle placed at that position, a clockwise velocity were generated, the robot, instead of avoiding it, would end up heading towards it.

The control law is designed to prioritize trajectory tracking while incorporating a secondary task that activates when an obstacle is detected, enabling the system to adjust its motion and avoid collisions:

$$\dot{\mathbf{q}} = \mathbf{v}_f + \mathbf{J}_0^{-1} \dot{\mathbf{x}}_0. \quad (7)$$

The term \mathbf{v}_f of the previous equation represents the planned velocity profile with error compensation, as defined by Maalouf et al. [51] and Kanayama et al. [52]:

$$\mathbf{v}_f = \begin{bmatrix} v_f \\ \omega_f \end{bmatrix} = \begin{bmatrix} v_r \cos e_\theta + k_x e_x \\ \omega_r + v_r k_y e_y + k_\theta \sin e_\theta \end{bmatrix}. \quad (8)$$

The terms e_x , e_y and e_θ represent the error components between the planned and actual poses with respect to the robot's local reference frame, while k_x , k_y and k_θ are the corresponding gain parameters.

An alternative strategy for generating repulsive motion during obstacle avoidance relies solely on the rotational movement of the platform. In this case, the repulsive velocity component perpendicular to the platform's edge is constrained to a predefined magnitude $v_{0n} = \dot{x}_0$, whereas the tangential component v_{0t} remains unconstrained. By enforcing a null velocity at the platform's central point \mathbf{P}_{mp} , the angular velocity can be determined as $\omega = \dot{x}_0 / (p_y \cos \alpha_{mp} - p_x \sin \alpha_{mp})$ which generates the required repulsive velocity. The geometric parameters involved in this approach are illustrated in Figure 3 and the resulting control law is expressed as follows:

$$\dot{\mathbf{q}} = \mathbf{v}_f + \begin{bmatrix} 0 \\ \dot{x}_0 \\ p_y \cos \alpha_{mp} - p_x \sin \alpha_{mp} \end{bmatrix}, \quad (9)$$

which is an alternative to Equation (7). In order to generate a rotation in the correct direction, the following geometric condition must be fulfilled in this type of control:

$$\alpha_{mp} > \tan^{-1} \left(\frac{D}{L} \right), \quad (10)$$

where the distance between the wheels is denoted as D and the length of the robot is denoted by L . In the case described in this paper, since $L = 890$ mm and $D = 580$ mm, it is necessary to impose $\alpha_{mp} > 33^\circ$. For this reason, an angle $\alpha_{mp} = 40^\circ$ was set.

Both approaches can be used to implement an obstacle avoidance strategy. However, since Equation (7) also generates a translational repulsive velocity, when the robot is in motion it is preferable to use the strategy of Equation (9), using only the rotational velocity to avoid obstacles. Nevertheless, the first mode was used in combination with the manipulator to avoid obstacles with a redundant system. This choice is due to the fact that, when the end-effector is in end-position and an obstacle interferes with the base, it can be useful to avoid the collision by moving the base in both translation and rotation, ensuring that the end-effector remains fixed in the target by exploiting redundancy.

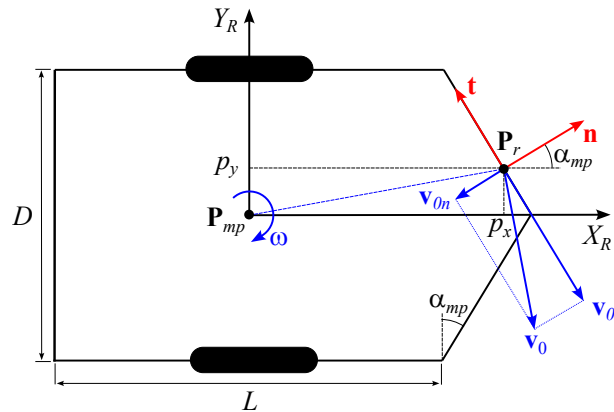


Figure 3. Geometric parameters for Equation (9).

2.2. Mobile Manipulator Obstacle Avoidance

In specific situations; for example, when the mobile manipulator is close to the destination point, it may be useful to use a strategy that allows the system to be controlled, moving both the robot and the base, to reach the end position while avoiding obstacles along the way or to keep the end-effector at the destination pose if an obstacle interferes with the base or the manipulator’s kinematic chain.

The collision avoidance algorithms were designed for a mobile manipulator consisting of UR10 serial arm (6 DoF) and a MiR100 platform (differential drive), but implementation to other mobile manipulators is simple due to the kinematic structure, with the exception of link lengths and base dimensions. The algorithm formulation described in this paper is generic and can be extended to other manipulators, even those with more than 6 degrees of freedom, using the correct Jacobian matrix. The kinematic chain, characterized by segments delimited by control points is shown in Figure 4, whereas Table 1 summarizes the dimensions of the segments. The MiR100 platform is equipped with a two-wheel differential drive system for locomotion, with two driven wheels and four additional free wheels for stabilization. In the CAD model, used in figure, only the four free wheels are visible, while the driven drive wheels are not shown but are located in the center of the mobile platform.

Table 1. Kinematic parameters for the mobile manipulator (UR10 + MIR100).

h [m]	l [m]	d_1 [m]	s_1 [m]	a_2 [m]
0.6670	0.2270	0.1280	0.1760	0.6127
s_3 [m]	a_3 [m]	d_4 [m]	d_5 [m]	d_6 [m]
0.1278	0.5716	0.1157	0.1157	0.0922

Since the system is redundant, it must be considered that the inverse velocity kinematic problem can be solved as

$$\dot{\mathbf{q}} = \mathbf{J}^\dagger \dot{\mathbf{x}} + \mathbf{N}\dot{\mathbf{q}}_0, \tag{11}$$

where $\mathbf{J}^\dagger = \mathbf{J}^T(\mathbf{J}\mathbf{J}^T)^{-1}$ is the Moore–Penrose pseudoinverse of the Jacobian matrix \mathbf{J} , $\mathbf{N} = \mathbf{I} - \mathbf{J}^\dagger\mathbf{J}$ is the projection into the null space of \mathbf{J} and the joint null space velocity $\dot{\mathbf{q}}_0$ has the effect of generating internal motions while maintaining the pose of the end-effector unchanged.

In addition, a damped least squares pseudoinverse must be used to mitigate singularity problems. Thus, the inverse of the Jacobian matrix \mathbf{J} can be calculated as follows:

$$\mathbf{J}^* = \mathbf{J}^T(\mathbf{J}\mathbf{J}^T + \lambda^2\mathbf{I})^{-1}. \tag{12}$$

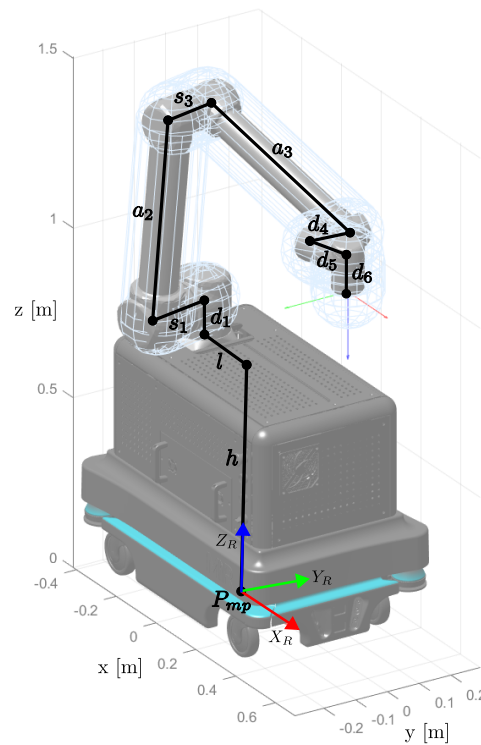


Figure 4. Kinematic chain for the mobile manipulator composed of an UR10 robotic arm and a mir100 mobile base.

In this case, the symbol λ represents the damping factor, which is determined by the minimum singular value of the Jacobian matrix. This parameter serves to balance accuracy (lower values) and numerical robustness (higher values). In particular, when dealing with a Jacobian matrix of full rank \mathbf{J} , the damping factor λ is set to 0.

For the mobile robot, the algorithm is designed to identify the manipulator link most at risk of collision by calculating the distance of each obstacle from every part of the robot’s structure. This process relies on two control points positioned at the extremities of each link. The obstacle’s center is denoted as \mathbf{P}_o , and its distance from the robot body is determined based on the diagrams in Figure 5. In the figure, \mathbf{d}_l represents the vector defining the link segment, while \mathbf{d}_p and \mathbf{d}_d correspond to the distances between the obstacle and the proximal and distal ends of the link, respectively. The point on the segment closest to the obstacle is denoted as \mathbf{P}_r and the Jacobian \mathbf{J}_0 associated with the linear velocity of \mathbf{P}_r is defined as

$$\mathbf{J}_0 = \begin{bmatrix} \mathbf{J}_{0p} \\ \mathbf{J}_{0r} \end{bmatrix}. \tag{13}$$

The robot’s movement is affected by the presence of an obstacle when the d_o distance is less than a certain radius r , a parameter intrinsic to the algorithm. Consequently, the pertinent zone of each link can be conceptualized as a cylindrical structure bounded by two hemispheres, all of which share the radius r . The repulsive velocity \mathbf{x}_0 is therefore implemented at the point \mathbf{P}_r . More specifically:

- (a) $\cos \beta < 0$; the point \mathbf{P}_r is localized to the distal extremity of the link:

$$\mathbf{d}_o = \mathbf{d}_d \quad x = 1 ; \tag{14}$$

- (b) $\cos \alpha \geq 0$ and $\cos \beta \geq 0$; the distance $\mathbf{d}_o = \mathbf{P}_r - \mathbf{P}_o$ is orthogonal to the link, and the position of \mathbf{P}_r along the link is defined by the scalar parameter x :

$$\mathbf{d}_o = \frac{|\mathbf{d}_l \times \mathbf{d}_p|}{d_l} \quad x = \frac{d_p \cos \alpha}{d_l}, \quad 0 < x < 1 ; \quad (15)$$

- (c) $\cos \alpha < 0$; the point \mathbf{P}_r is localized to the proximal extremity of the link:

$$\mathbf{d}_o = \mathbf{d}_p \quad x = 0 . \quad (16)$$

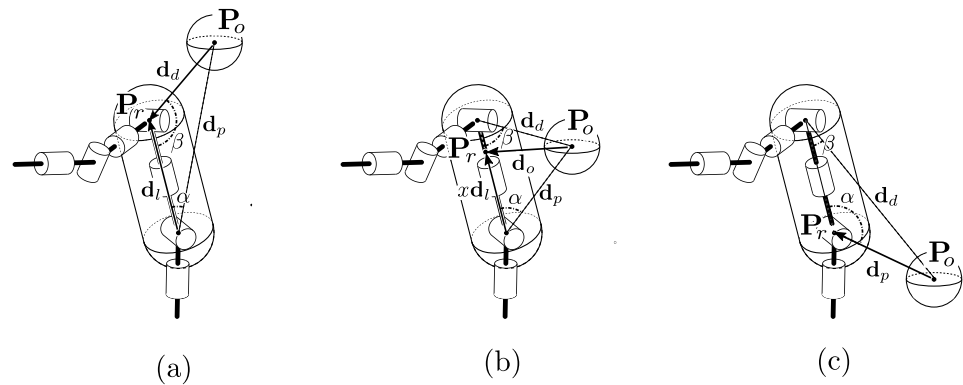


Figure 5. Obstacle–link distance calculation; cases (a–c) refer to Equations (14), (15) and (16), respectively.

Once the distance and position of the most critical obstacle has been determined, it is possible to obtain the speed to be imposed on the robot’s joints $\dot{\mathbf{q}} = [v \ \omega \ \dot{q}_1 \ \dot{q}_2 \ \dots \ \dot{q}_n]^T$ by adapting the algorithm for redundant manipulators proposed by the authors in [45]. The necessary modifications are made to account for the mobile base with differential wheels. For instance, if the obstacle is located at the bottom, i.e., below the base of the manipulator, it will affect the moving platform rather than directly impacting the manipulator’s kinematic chain. In this case, \mathbf{P}_r and the respective Jacobian \mathbf{J}_{0p} will be calculated as described previously in Equation (6) in which the algorithm is referred to the mobile base.

The algorithm is based on the closed-loop inverse kinematics (CLIK) approach and the null space method for redundancy control. Joint velocities can be calculated as

$$\dot{\mathbf{q}} = \mathbf{J}^*(\dot{\mathbf{x}} + \mathbf{K}\mathbf{e}) + a_n(\mathbf{J}_{0p}\mathbf{N})^*(a_v v_{rep} \hat{\mathbf{d}}_0 - \mathbf{J}_{0p}\mathbf{J}^*\dot{\mathbf{x}}), \quad (17)$$

$\dot{\mathbf{x}}$ represents the vector of planned velocities, and $\dot{\mathbf{x}}_0 = a_v v_{rep} \hat{\mathbf{d}}_0$ denotes the repulsive velocity resulting from the presence of an obstacle. In this expression, v_{rep} is a customizable magnitude, and a_v is an activation factor determined by the distance from the obstacle \mathbf{d}_0 . On the other hand, the term a_n exerts its influence with a weight that serves to balance the impact of the homogeneous solution, which is associated with collision avoidance, in relation to the overall solution [45].

The term \mathbf{e} represents a vector encompassing position and orientation errors, denoted as $(\mathbf{e}_p$ and $\mathbf{e}_r)$, respectively. Its definition is defined as

$$\mathbf{e} = \begin{bmatrix} \mathbf{e}_p \\ \mathbf{e}_r \end{bmatrix} = \begin{bmatrix} \mathbf{P}_d - \mathbf{P}_e \\ \frac{1}{2}(\mathbf{i} \times \mathbf{i}_d + \mathbf{j} \times \mathbf{j}_d + \mathbf{k} \times \mathbf{k}_d) \end{bmatrix}, \quad (18)$$

where \mathbf{P}_e denotes the position of the end-effector, and \mathbf{P}_d indicates the desired planned variable. Instead, \mathbf{i} , \mathbf{j} and \mathbf{k} are the unit vectors of the end-effector reference frame. In (17), the error \mathbf{e} is multiplied by $\mathbf{K} = k_e \mathbf{I}$, where $\mathbf{k}_e = [k_{ep} \ k_{ep} \ k_{ep} \ k_{er} \ k_{er} \ k_{er}]$ is a gain

vector. It is possible to give different gains to the orientation (k_{er}) or positioning (k_{ep}) components of the error.

In case an obstacle is going to interfere with the end-effect of the robot, an additional term of velocity is added to Equation (17):

$$\dot{\mathbf{q}} = \mathbf{J}^*(\dot{\mathbf{x}} + \mathbf{K}\mathbf{e}) + a_h(\mathbf{J}_{0p}\mathbf{N})^*(a_v v_{rep} \hat{\mathbf{d}}_0 - \mathbf{J}_{0p}\mathbf{J}^* \dot{\mathbf{x}}) + \mathbf{J}_p^* a_{ee} v_{rep} \hat{\mathbf{d}}_{ee}, \quad (19)$$

where \mathbf{J}_p is the translation part of the Jacobian, a_{ee} an additional activation parameter and $\hat{\mathbf{d}}_{ee}$ is the direction of the distance between the obstacle and the end-effector.

2.3. Weighted Pseudoinverse Matrix

Since the analyzed system is redundant, it has the ability to respond to the presence of obstacles in various ways. For instance, if the mobile manipulator is stationary and an obstacle interferes with the end-effector, it is possible to react by applying a velocity $\dot{\mathbf{q}}$ to all elements of the joint space. This would result in the movement of both the robotic arm and the mobile base. Alternatively, the system can respond by moving only the mobile platform, applying velocities primarily to v and ω , while the arm reaches very low velocities or, in extreme cases, remains stationary. For this purpose, the weighted pseudoinverse matrix \mathbf{J}_w^* can be used, where \mathbf{J} is the Jacobian and \mathbf{W} is a diagonal matrix of the weights associated with each element of the joint space [53,54]. Its formulation is given by

$$\mathbf{J}_w^* = \mathbf{W}^{-1} \mathbf{J}^T (\mathbf{J} \mathbf{W}^T \mathbf{J}^T + \lambda^2 \mathbf{I})^{-1}. \quad (20)$$

The \mathbf{W} matrix is an $m \times m$ diagonal matrix (where m is the number of Jacobian columns) with non-negative elements along the diagonal, whose values reflect the relative importance. The matrix is defined as

$$\mathbf{W} = \begin{bmatrix} w_1 & & \\ & \ddots & \\ & & w_m \end{bmatrix} = \begin{bmatrix} \mathbf{W}_{mp} & \\ & \mathbf{W}_a \end{bmatrix}. \quad (21)$$

The first two elements of the diagonal (w_1, w_2), i.e., belonging to the $\mathbf{W}_{mp} = w_{mp} \mathbf{I}$ matrix, weight the movement of the moving platform, while the remaining n elements (q_1, \dots, q_n), i.e., belonging to the $\mathbf{W}_a = w_a \mathbf{I}$ matrix, define the coefficients affecting the movement of the arm.

This matrix is a generalization of the classical pseudoinverse matrix that takes into account the differences in importance between the different elements of the vector. This matrix is often used in applications related to the inverse kinematics of robots, to effectively handle situations wherein one wants to give more importance to some elements of the joint space, being perhaps more critical or sensitive than others. By introducing appropriate weights, it is possible to influence the resolution of inverse kinematic solutions, ensuring a more accurate and adaptable control of the robot in different operational scenarios. In the context of inverse kinematics of robots, this formulation allows for flexible handling of movement priorities. This means that the increment of element j of the joint vector \mathbf{q} is determined by the value of element w_j of the weighting matrix \mathbf{W} . When the value of element w_j is significantly larger than the other elements, the increment in the corresponding joint vector element, q_j , becomes small. Conversely, when the value of w_j is relatively small, it becomes larger. In essence, a large value of the coefficient w_j corresponds to a more constrained or sensitive joint, resulting in smaller variations, while a smaller value allows for larger variations of q_j . Sections 3.1 and 3.3 provide supporting examples for

the concepts introduced. In addition, a link to a repository with additional videos and MATLAB simulation code is provided in the Supplemental Materials.

3. Simulations

The strategy was verified in simulation using MATLAB R2022b software. A series of tests were carried out to analyze various situations in which the mobile manipulator may need to avoid obstacles; for example, in an industrial environment wherein objects or people may be present. In the examples presented, obstacles, regardless of type, will be simulated as spheres. The algorithm is designed to be independent of the type of sensors used for obstacle detection; it is sufficient to have as input the Cartesian coordinates of the detected objects, whether they are grouped in clusters or consist of multiple objects.

A limit \dot{q}_{max} of angular velocity was imposed on all joints of the robotic arm as a saturation threshold, while for the mobile robot a limit v_{max} and ω_{max} of the local reference frame was imposed in order to avoid high velocities that may occur when planning a collision avoidance task. The resulting positioning error is compensated in subsequent movements by a proportional correction term. Table 2 shows the parameters used in common for all simulations.

Table 2. Main parameter values used for all the simulations.

r_{inf} [m]	r_{sup} [m]	v_{inf} [m/s]	v_{sup} [m/s]	λ_{max}	ϵ	k [m]
0.10	0.13	0.1	0.5	10^{-3}	10^{-3}	5
\dot{q}_{max} [rad/s]	v_{max} [m/s]	ω_{max} [rad/s]	v_{rep} [m/s]	k_x [s ⁻¹]	k_y [rad/m ²]	k_θ [rad/s]
$\frac{\pi}{2}$	0.5	$\frac{\pi}{2}$	1	2	2	0.5

For the safety zone enclosing the robot’s links, 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 determined as a linear function of the relative obstacle/robot velocity v_o , with lower and upper bounds ($r_{min} = r_{inf}$ for $v_o \leq v_{inf}$, $r_{min} = r_{sup}$ for $v_o \geq v_{inf}$).

3.1. Example 1

The first simulation shows a situation where the robot tool is at the point to be reached and an obstacle interferes with the end-effector. In this case, the mobile manipulator can avoid the obstacle by using redundancy, and the weighted matrix can be managed to prioritize movements of either the base or the arm. The robot is initially stationary as it is already in the desired position. When the obstacle interferes, it starts moving to avoid it and then returns to the desired position once the obstacle exits the safety zone. In this case, it is not necessary to plan a path; the arrival point is sufficient since the goal is to stay as close as possible to the final position.

Figure 6 shows the movement of the robot to avoid the obstacle. The execution period is $T = 5$ s and both the kinematic scheme that allows the algorithm to run and the CAD view are proposed. In the first two frames, an arrow indicates the direction in which the obstacle is moving.

The parameters used in the example presented are shown in Table 3.

Table 3. Parameter values used for the simulation 1.

k_{ep} [s ⁻¹]	k_{er} [s ⁻¹]	T [s]
1	2	5

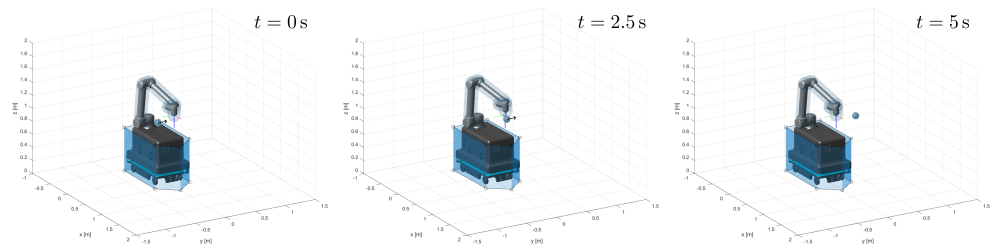


Figure 6. Example 1: Frames of the simulation showing the mobile manipulator and the safety volumes.

This example shows how the behavior of the mobile manipulator varies if different weights are set to the weighted matrix.

When equal importance is given to the movements of the manipulator and the platform ($w_{mp} = w_a = 1$), the robot has the opportunity to avoid the obstacle by moving all available joints. This solution is shown in Figure 6 and the graph showing how the minimum robot–obstacle distance varies is shown in Figure 7. Although these graphs are useful for understanding the situation being analyzed, they do not allow for the difference in movement to be analyzed as the weights change. For this reason, they are only presented for the case in which arm and base have the same priority of movement.

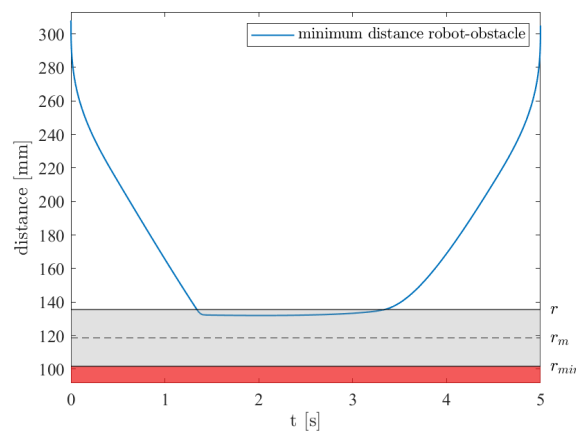


Figure 7. Example 1: minimum robot–obstacle distance for the motion shown in Figure 6.

Speed trend graphs are shown to analyze the different obstacle avoidance modes. In each figure, the profile of the velocities v and ω of the mobile robot, with respect to the local reference system, is presented on the left, while on the right the velocity profile of the manipulator joints.

Figure 8 shows the trend in the case where the same unit weight is assigned to the arm and the moving base ($w_{mp} = w_a = 1$). In this case, the velocities generated, which allow for the obstacle to be avoided and the position subsequently recovered, have comparable values.

In Figure 9, movement priority is given to the mobile robot, while still leaving the manipulator free to move, but limiting its speed. The weights associated with the mobile robot and manipulator are $w_{mp} = 10^{-5}$ and $w_a = 10^5$, respectively. In this case, as can be seen from the graph, the local speed values of the mobile robot are higher. It should be emphasized that some movement directions can only be executed by certain joints of the manipulator; for example, to generate an upward velocity. That is why the speed trend of some joints remains unchanged.

Figure 10 shows the trend of speeds in the case when the priority of movements is given to the manipulator. The weights associated with the mobile robot and manipulator are $w_{mp} = 10^5$ and $w_a = 10^{-5}$, respectively. Since the arm has more dexterity of movement,

in this case it is possible to avoid the obstacle only with the manipulator. For this reason, the speed generated for the mobile robot becomes zero, while that of the manipulator has high values.

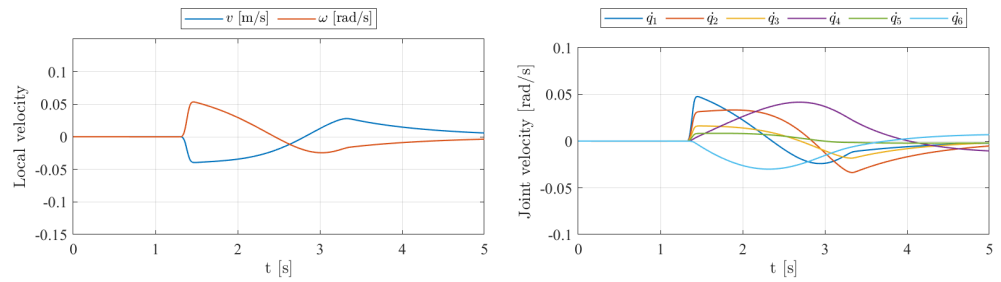


Figure 8. Example 1, case 1: local velocity and joint speeds for the mobile robot and the manipulator when they have the same priority.

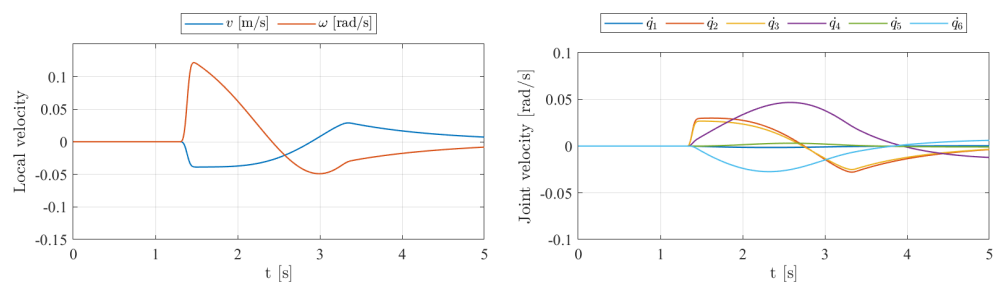


Figure 9. Example 1, case 2: local velocity and joint speeds for the mobile robot and the manipulator when the mobile base has priority.

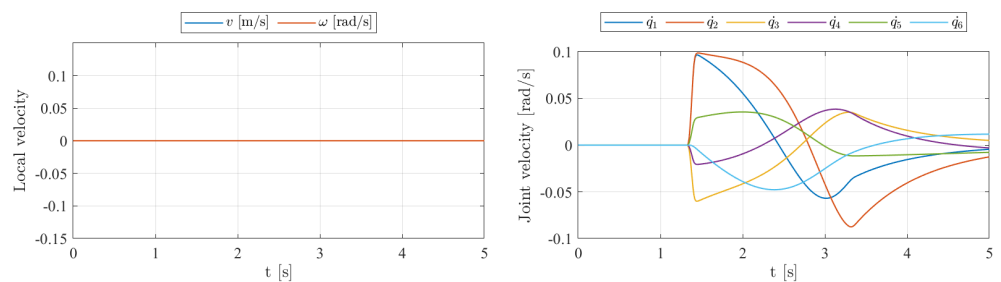


Figure 10. Example 1, case 3: local velocity and joint speeds for the mobile robot and the manipulator when the arm has priority.

The velocity profiles of the end-effector are also presented for the three different cases. Figure 11 illustrates the evolution of the six Cartesian velocity components when the manipulator and the base share the same movement priority. Figure 12 depicts the scenario where movement priority is assigned to the mobile base, while Figure 13 shows the case in which the manipulator is given priority.

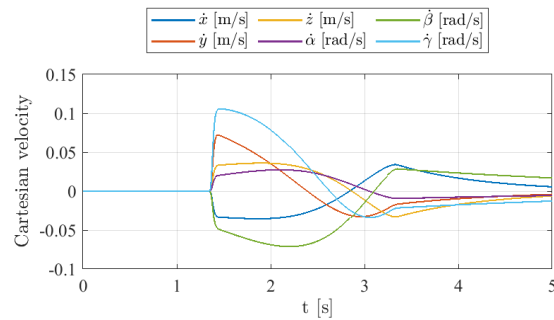


Figure 11. Example 1, case 1: Cartesian velocity of the end-effector for the mobile robot and the manipulator when they have the same priority.

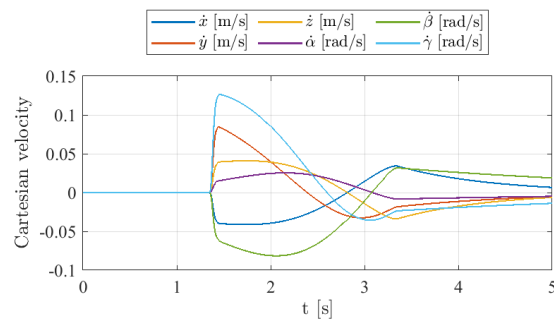


Figure 12. Example 1, case 2: Cartesian velocity of the end-effector for the mobile robot and the manipulator when the mobile base has priority.

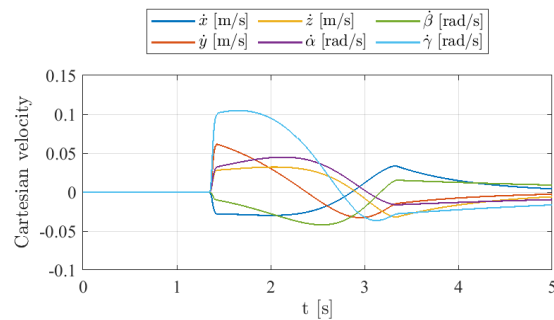


Figure 13. Example 1, case 3: Cartesian velocity of the end-effector for the mobile robot and the manipulator when the arm has priority.

Tables 4–6 show, respectively, the results of case 1, 2 and 3 of Example 1. A comparison of the path lengths performed by the end-effector is shown. In addition, the maximum local velocities $|v|_{max}$ and $|\omega|_{max}$ performed by the mobile robot during the path are proposed; together with the maximum velocity $|\dot{q}|_{max}$ reached by the manipulator joints and the maximum of the linear speed norm of the end-effector $\|\dot{\mathbf{x}}_p\|_{max}$.

Table 4. Results from simulations of Example 1, case 1.

ΔL [m]	$\ \dot{\mathbf{x}}_p\ _{max}$ [m/s]		
0.141	0.085		
$ v _{max}$ [m/s]	$ \omega _{max}$ [rad/s]	$ \dot{q} _{max}$ [rad/s]	
0.040	0.054	0.048	

Table 5. Results from simulations of Example 1, case 2.

ΔL [m]	$ \dot{\mathbf{x}}_p _{max}$ [m/s]	
0.137	0.100	
$ v _{max}$ [m/s]	$ \omega _{max}$ [rad/s]	$ \dot{q} _{max}$ [rad/s]
0.039	0.122	0.047

Table 6. Results from simulations of Example 1, case 3.

ΔL [m]	$ \dot{\mathbf{x}}_p _{max}$ [m/s]	
0.144	0.073	
$ v _{max}$ [m/s]	$ \omega _{max}$ [rad/s]	$ \dot{q} _{max}$ [rad/s]
0.000	0.000	0.099

3.2. Example 2

The second example shows the possibility of using null space to keep the position of the end-effector fixed in one point and avoid the obstacle by moving the rest of the kinematic chain. This situation can be useful if an object is attached to the robot tool to be kept fixed, or if a precision operation is to be performed, but leaving the moving base free to move to avoid obstacles. In order for the positioning accuracy of the end-effector to be sufficiently high, the values of the coefficients k_{ep} and k_{er} were increased. Figure 14 shows the simulation, performed over a period of $T = 5$ s. The parameters used in the example presented are shown in Table 7.

Table 7. Parameter values used for the simulation 2.

k_{ep} [s^{-1}]	k_{er} [s^{-1}]	T [s]
50	50	5

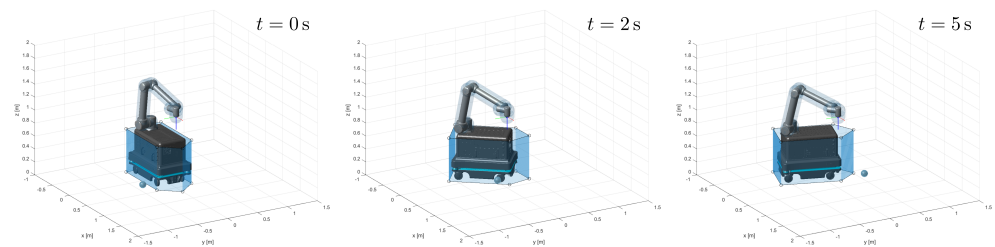


Figure 14. Example 2: Frames of the simulation showing the mobile manipulator and the safety volumes.

In this test, all manipulator joints and the moving base were given the same importance of movement ($w_{mp} = w_a = 1$). The graph showing how the minimum robot–obstacle distance varies is shown in Figure 15. Looking at the trend, it is possible to see how the obstacle remains in the vicinity of the moving base for almost the entire simulation period. This is due to the fact that the platform has a large size and the simulated obstacle travels in a straight line through the initial position of the base.

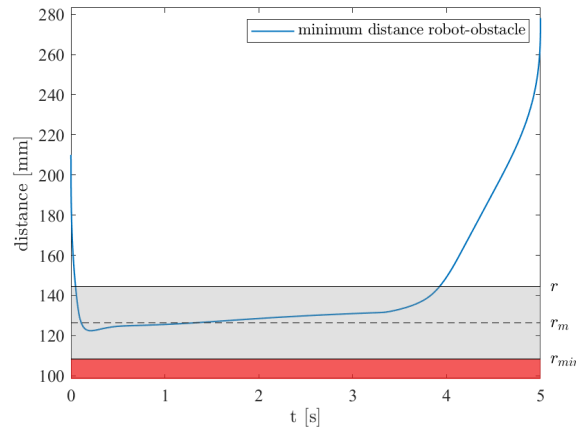


Figure 15. Example 2: minimum robot–obstacle distance for the motion shown in Figure 14.

Figure 16 shows the evolution of the generated velocities. The angular velocities of the base ω and the first joint of the manipulator $\dot{\theta}_1$ have the highest values, as they are fundamental for achieving avoidance. The maximum set speeds are not reached, because the period T allows the obstacle to be avoided with low speeds. Table 8 shows the simulation results.

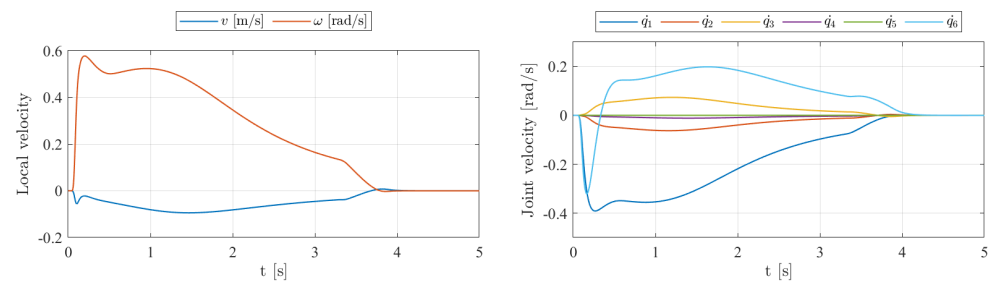


Figure 16. Example 2: local velocity and joint speeds for the mobile robot and the manipulator.

Table 8. Results from simulations of Example 2.

$ v _{max}$ [m/s]	$ \omega _{max}$ [rad/s]	$ \dot{q} _{max}$ [rad/s]
0.094	0.578	0.392

3.3. Example 3

In the third test presented, the mobile manipulator has to reach, with the end-effector, a target point located several meters away from the starting point. Along the way, there are three obstacles: the first obstacle is positioned stationary on the ground; the second obstacle performs a straight trajectory by interfering with the arm tool; the third obstacle is stationary along the robot’s trajectory at a height comparable with that of the end-effector, in an area close to the target to be reached.

This situation reflects a typical industrial application where the mobile manipulator has to transport an object from one work area to another and then, once it reaches the end point, deposit it.

If the point to be reached is located in an area far away from the target point, it may be unnecessary to move the arm. Moving the robot joints is energy-consuming and, unless it is necessary to follow a predetermined trajectory (painting or welding), unnecessary. For this reason, in the first test presented, when the robot is at a distance from the target $d_t > 3$ m, the obstacle avoidance strategy described in the Section 2.1 is used by imposing a speed that allows the mobile base to follow a desired trajectory and avoid any obstacles in the path by imposing the repulsive speed of Equation (9).

In this case, only the movement of the base is used, considering the safety area and the obstacle as if they were projected onto the plane. For simplicity, the distance d_t was calculated as the Euclidean distance between the two projections in the plane of the end point to be reached and the instantaneous position of the end-effector. In this area, the planning and obstacle avoidance algorithm operates on the plane.

When the distance d_t becomes sufficiently small, it is possible to begin a transition phase in which the manipulator begins to move. In this way, it is possible to increase the dexterity of the movements and differentiate the avoidance of obstacles when they are located at a height above that of the platform (h), interfering with the arm. In fact, in the previous case, even when the obstacle is located at a height greater than h , only the base is used to avoid the object.

The five segments used to define the base, in particular, were positioned along the perimeter of the platform, with the presence of a tip, at a variable height coinciding with the possible height of the obstacle. On the other hand, when $d_t \leq 3$ m, this solution is adopted only if the height at which the obstacle is positioned is less than the height of the platform h .

In the transition phase, i.e., when $1.5 \text{ m} < d_t \leq 3 \text{ m}$, the weighted matrix in the pseudoinverse of the Jacobian (Equation (20)) is used to avoid a rapid movement of the robot and to give the priority of the movements to the moving platform. In this way, the strategy described in Section 2.2 can be used to exploit redundancy, while still giving priority of movements to the base.

When the robot is in an area sufficiently close to the end point to be reached, in the test case considered $d_t \leq 1.5$ m, it will be possible to make full use of the robot's redundancy to make the end-effector follow the desired trajectory and avoid obstacles. In this area, the arm and base are given the same weight, so that the available dexterity can be utilized. In this area, the planning and obstacle avoidance algorithm operates in three-dimensional space, acting on the end-effector.

In this way, three zones can be defined: the first in which only the base moves and the arm is kept fix; the second in which there is a transition phase and the arm starts to move, but the base is given priority for movement thanks to the weighted matrix; finally, when the robot is sufficiently close to the point to be reached, zone three allows for full exploitation of the redundancy of the mobile manipulator to reach the target and avoid obstacles, giving equal weight to the arm and the base.

Figure 17 shows the division of the workspace for the presented example. The mobile manipulator is positioned in the starting configuration, with the end-effector at the initial point P_i . P_f indicates the target point to be reached. There are three obstacles along the robot's path. Obstacle 1 is located in zone 1, at point O_1 . The other two obstacles interfere with the end-effector and part of the manipulator. The second one is located in zone 2 and moves from point O_{2i} to O_{2f} . The third is stationary at point O_3 , located in zone 3.

The parameters used in the example presented are shown in Table 9.

Table 9. Parameter values used for the simulation 3.

$k_{ep} [\text{s}^{-1}]$	$k_{er} [\text{s}^{-1}]$	$T [\text{s}]$
1	2	30

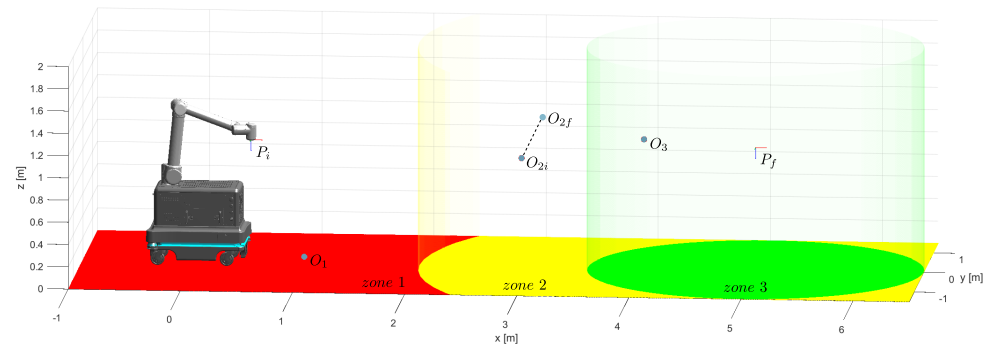


Figure 17. Workspace for simulation 3 divided into three zones: Zone 1 in which only the mobile robot is used; Zone 2 in which the arm can move but giving more importance to the movements of the base; Zone 3 in which the base and manipulator have the same weight.

Figure 18 illustrates the robot’s motion, with frames captured at intervals between $T = 0$ and $T = 30$ s. Obstacles are represented as circles and as the robot enters their region of influence, the algorithm responds by generating a repulsive velocity component to change its trajectory. The plotted paths include both the initially planned path (dashed line) and the actual trajectory adjusted by the avoidance algorithm (solid line). As the robot overcomes the obstacle, the control strategy progressively corrects any deviation through a proportional feedback mechanism, ensuring that the robot returns to the planned path. Furthermore, Figure 18 highlights the closest interaction point P_r , which represents the segment of the robot’s footprint most at risk of collision. This critical point is marked with a small circle to emphasize its role in the obstacle avoidance process. When the robot is in zone 2, the weights associated with the mobile robot and manipulator are $w_{mp} = 10^{-3}$ and $w_a = 10^3$, respectively. In this way, movement priority is given to the mobile base. When the mobile manipulator arrives close enough to the target point P_f , it enters zone 3 and equal importance is given to the movements of the manipulator and the platform ($w_{mp} = w_a = 1$).

Figure 19 shows the trend of the minimum distance between the obstacle and the robot. The obstacle avoidance control is activated when the distance becomes less than the threshold r that delimits the influence region around the robot’s control point. In more detail, r is the control distance defining the influence region, r_{min} the distance below which no action is possible and the robot’s speed is reset to zero to stop it and r_m is a critical distance intermediate between r_{min} and r . No influence of the obstacle is desired if $d_o > r$, while the robot stops if $d_o < r_{min}$. Between these limits, the coefficients vary continuously from 0 to 1, as described in paper [45]. It can be seen from the graph that the minimum distance becomes less than r in three situations, corresponding to the three obstacles encountered along the path.

The speed profiles of the mobile robot and manipulator are shown in Figure 20 and Figure 21, respectively. The velocity profile v and ω of the mobile robot has peaks mainly in the first two areas, as it has priority of movement. In Figure 21, the separation into the three zones is evident. In zone 1, the joint velocity is zero, as the manipulator is not used for either planning or obstacle avoidance. In zone 2, there is a joint velocity trend, but with reduced values compared to zone 3, as it can move but still giving priority to the moving base. In addition, in zone 3 the manipulator interferes with the third obstacle twice: the first involves the end-effector and the second a part of the wrist. In both cases, the algorithm generates a repulsive velocity that allows for the obstacle to be avoided.

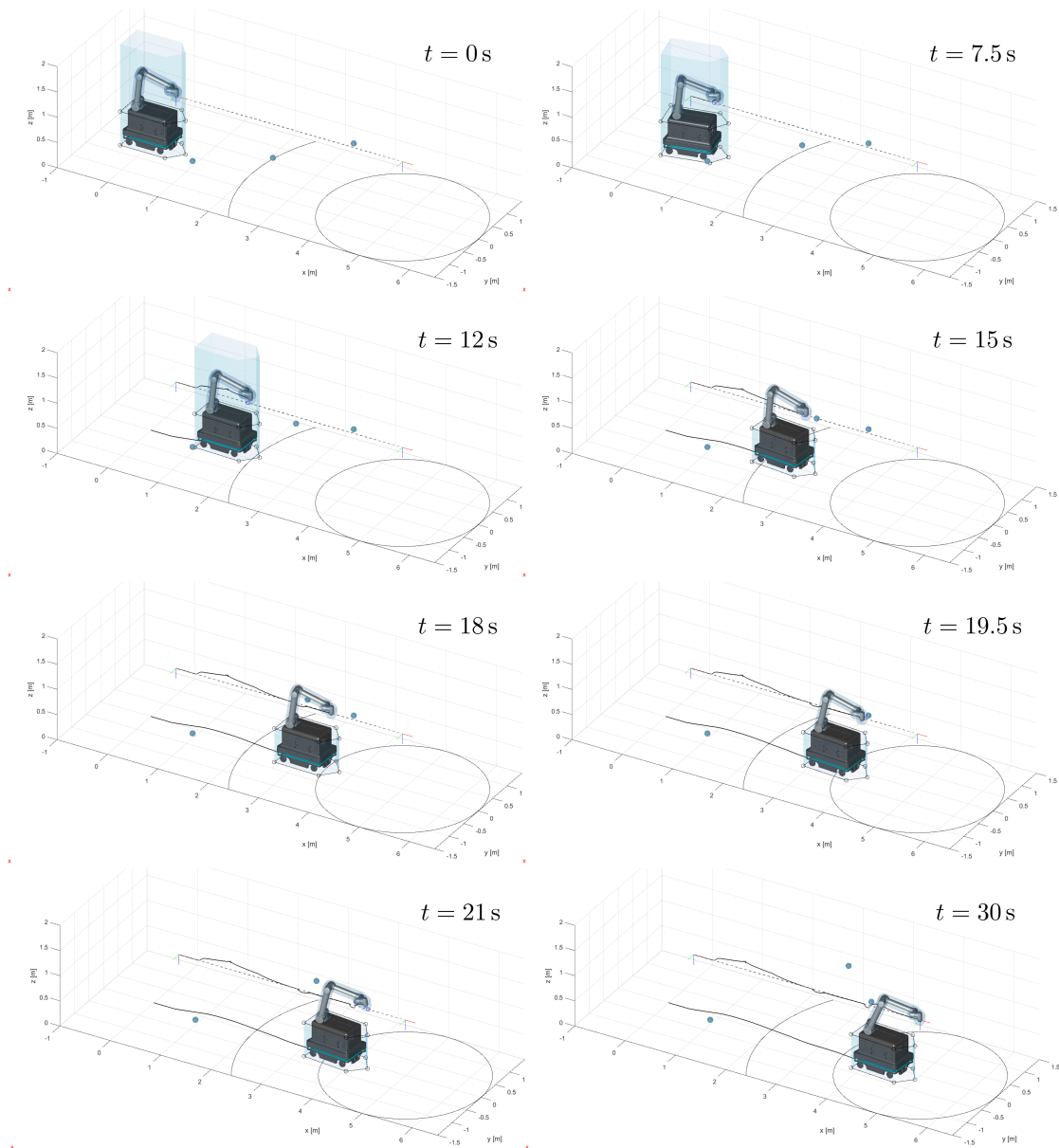


Figure 18. Example 3: Frames of the simulation showing the mobile manipulator and the safety volumes.

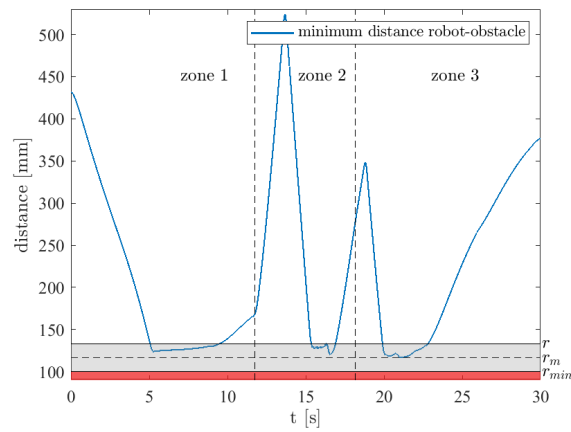


Figure 19. Example 3: minimum robot–obstacle distance for the motion shown in Figure 18.

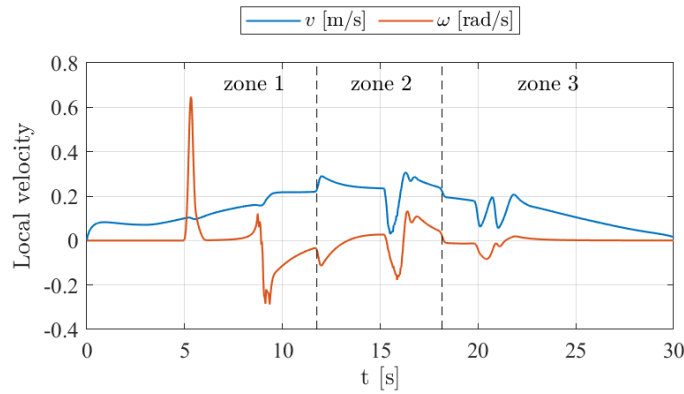


Figure 20. Example 3: local velocity of the mobile robot.

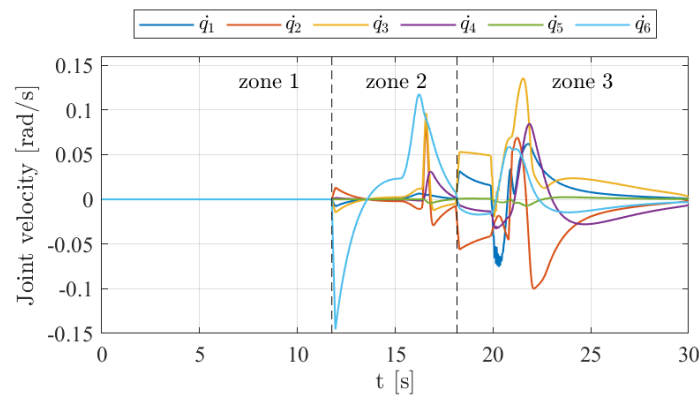


Figure 21. Example 3: joint speeds of the manipulator.

Table 10 shows the simulation results. The path length effectively executed by the end-effector is longer than the planned one, increasing by 228 mm to allow the robot to avoid obstacles.

Table 10. Results from simulations of Example 3.

$L_{planned}$ [m]	$L_{effective}$ [m]	$ \dot{\mathbf{x}}_p _{max}$ [m/s]
4.500	4.728	0.457
$ v _{max}$ [m/s]	$ \omega _{max}$ [rad/s]	$ \dot{q} _{max}$ [rad/s]
0.306	0.647	0.145

4. Conclusions

In this paper, an obstacle avoidance strategy for mobile manipulators with a differential wheel base is described and tested in simulation. In addition to this, the possibility of prioritizing the movement of some parts of the system is introduced; for example, limiting the movement of the arm when long distances are traveled, or limiting the movement of the base when the robot is in the proximity of the point to be reached. A series of examples are presented to evaluate how the algorithm works.

Section 3.1 shows how the behavior of the mobile manipulator varies if different weights are set to the weighted matrix to avoid an obstacle interfering with the end-effector.

- Case 1 with $\Delta L = 0.141$ m sits between the two extremes, finding a compromise between path length and overall speed.
- Case 2, focusing on the mobile robot’s movement, has the highest end-effector velocity, suggesting faster movement during the path.

- Case 3, prioritizing the manipulator, exhibits the longest path and shows significantly higher joint velocities ($|\dot{q}|_{max} = 0.099$ rad/s), indicating more dynamic and complex movements to navigate obstacles.

The distribution of weights between the mobile base and manipulator significantly influences the overall system behavior. Prioritizing the manipulator (Case 3) may be advantageous in situations requiring greater dexterity and flexibility in movements, while prioritizing the mobile robot (Case 2) may favor quicker and more direct navigation.

In summary, the analysis of data and graphs confirms that the choice of weights and control strategies critically impacts the system's performance during navigation and obstacle avoidance. These findings are essential for optimizing the robot's behavior in complex and dynamic environments.

The examples show the ability of the mobile manipulator to avoid obstacles when they interfere with both the base and the arm links. In particular, in Section 3.2, if the robot is already at the desired point and an obstacle interferes with the platform, the null space will allow the robot to move away while keeping the end-effector fixed.

In Section 3.3, current standards are taken into account to show a case study in which the system has to travel a long distance to reach a predetermined destination while avoiding obstacles located along the trajectory. This scenario is divided into three operational zones:

- Zone 1 (farthest zone) where the robot primarily uses the mobile base for movement, keeping the arm stationary.
- Zone 2 (intermediate zone) where the arm is utilized for obstacle avoidance while the base still has movement priority.
- Zone 3 (closest zone) where both manipulator and moving platform are used with equal importance, exploiting system redundancy for precise positioning.

This dual-strategy approach, where the mobile base is used for long-distance travel and the manipulator arm is activated for final precision positioning, offers a comprehensive solution for tasks requiring straight trajectory end-effector motion. This methodology enhances efficiency, safety and precision, making it suitable for various industrial and service robotics applications.

The test results demonstrate the effectiveness of the proposed algorithm for navigating the robot through three operational zones. Initially, the planned route required the robot to travel 4.5 m. During the test, the robot was able to adapt using the obstacle avoidance strategy, with a slight increase in the distance traveled. This increase was 5% over the originally planned distance, indicating that the algorithm allowed the robot to dynamically adapt its trajectory over the time period considered, while maintaining safety and efficiency.

The results obtained made it possible to verify the functioning of the algorithm by providing velocity trends showing the behavior of the system when it interferes with obstacles. Upcoming research will explore the use of variable weights to further optimize the priority between the manipulator and the moving base, evaluating how the system's behavior varies as the coefficients change. Further future work includes the analysis of the stability of the system [55] and the implementation on a real mobile manipulator consisting of a UR collaborative robot positioned on a MIR mobile base. In this way, it will be possible to test the operation on a real system, where the frequency with which the robot is controlled and the accuracy of the sensors used for obstacle detection can influence the system.

Supplementary Materials: The following supporting information can be downloaded at: https://github.com/fedene97/mobile_manipulator_obstacle_avoidance (accessed on 1 April 2025).

Author Contributions: Conceptualization, G.P. and M.C.; validation, G.P.; investigation, F.N. and G.P.; writing—original draft preparation, F.N.; writing—review and editing, G.P.; supervision, G.P. and M.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Salman, M.; Khan, H.; Lee, M.C. Perturbation observer-based obstacle detection and its avoidance using artificial potential field in the unstructured environment. *Appl. Sci.* **2023**, *13*, 943. [CrossRef]
2. Prigozin, A.; Degani, A. Interacting with Obstacles Using a Bio-Inspired, Flexible, Underactuated Multilink Manipulator. *Biomimetics* **2024**, *9*, 86. [CrossRef]
3. Frohm, J.; Lindström, V.; Winroth, M.; Stahre, J. The industry's view on automation in manufacturing. *IFAC Proc. Vol.* **2006**, *39*, 453–458. [CrossRef]
4. Antwiadjei, L. Evolution of Business Organizations: An Analysis of Robotic Process Automation. *Eduzone Int. Peer Rev. Multidiscip. J.* **2021**, *10*, 101–105.
5. Peidro, A.; Haug, E.J. Obstacle avoidance in operational configuration space kinematic control of redundant serial manipulators. *Machines* **2023**, *12*, 10. [CrossRef]
6. Scalera, L.; Lozer, F.; Giusti, A.; Gasparetto, A. An experimental evaluation of robot-stopping approaches for improving fluency in collaborative robotics. *Robotica* **2024**, *42*, 1386–1402. [CrossRef]
7. Niloy, M.A.; Shama, A.; Chakraborty, R.K.; Ryan, M.J.; Badal, F.R.; Tasneem, Z.; Ahamed, M.H.; Moyeen, S.I.; Das, S.K.; Ali, M.F.; et al. Critical design and control issues of indoor autonomous mobile robots: A review. *IEEE Access* **2021**, *9*, 35338–35370. [CrossRef]
8. Rubio, F.; Valero, F.; Llopis-Albert, C. A review of mobile robots: Concepts, methods, theoretical framework, and applications. *Int. J. Adv. Robot. Syst.* **2019**, *16*, 1729881419839596. [CrossRef]
9. Braga, R.G.; Tahir, M.O.; Iordanova, I.; St-Onge, D. Robotic deployment on construction sites: Considerations for safety and productivity impact. *arXiv* **2024**, arXiv:2404.13143.
10. Rose, J. New industrial mobile robot safety standard, R15. 08: R15. 08, American National Standard for Industrial Mobile Robots—Safety Requirements—Part 1: Requirements for the Industrial Mobile Robot. *Control Eng.* **2020**, *67*, M1.
11. Belzile, B.; St-Onge, D. Safety first: On the safe deployment of robotic systems. In *Foundations of Robotics: A Multidisciplinary Approach with Python and ROS*; Springer: Berlin/Heidelberg, Germany, 2022; pp. 415–439.
12. Tagliavini, L.; Colucci, G.; Botta, A.; Cavallone, P.; Baglieri, L.; Quaglia, G. Wheeled mobile robots: State of the art overview and kinematic comparison among three omnidirectional locomotion strategies. *J. Intell. Robot. Syst.* **2022**, *106*, 57. [CrossRef] [PubMed]
13. Li, W.; Yang, C.; Jiang, Y.; Liu, X.; Su, C.Y. Motion planning for omnidirectional wheeled mobile robot by potential field method. *J. Adv. Transp.* **2017**, *2017*, 4961383. [CrossRef]
14. Muir, P.F.; Neuman, C.P. Kinematic modeling for feedback control of an omnidirectional wheeled mobile robot. In *Autonomous Robot Vehicles*; Springer: New York, NY, USA, 1990; pp. 25–31.
15. Popov, V.; Topalov, A.V.; Stoyanov, T.; Ahmed-Shieva, S. Kinematic Modeling with Experimental Validation of a KUKA®–Kinova® Holonomic Mobile Manipulator. *Electronics* **2024**, *13*, 1534. [CrossRef]
16. Mnubi, S.A. Motion planning and trajectory for wheeled mobile robot. *Int. J. Sci. Res.* **2016**, *5*, 1064–1068.
17. Han, S.; Choi, B.; Lee, J. A precise curved motion planning for a differential driving mobile robot. *Mechatronics* **2008**, *18*, 486–494. [CrossRef]
18. Velasco-Villa, M.; Cruz-Morales, R.D.; Rodriguez-Angeles, A.; Domínguez-Ortega, C.A. Observer-based time-variant spacing policy for a platoon of non-holonomic mobile robots. *Sensors* **2021**, *21*, 3824. [CrossRef]
19. Belzile, B.; Wanang-Siyapdjie, T.; Karimi, S.; Gomes Braga, R.; Iordanova, I.; St-Onge, D. From safety standards to safe operation with mobile robotic systems deployment. *arXiv* **2025**, arXiv:2502.20693.
20. Yang, M.; Yang, E.; Zante, R.C.; Post, M.; Liu, X. Collaborative mobile industrial manipulator: A review of system architecture and applications. In Proceedings of the 2019 25th International Conference on Automation and Computing (ICAC), Lancaster, UK, 5–7 September 2019; pp. 1–6.

21. Markis, A.; Papa, M.; Kaselautzke, D.; Rathmair, M.; Sattinger, V.; Brandstötter, M. Safety of mobile robot systems in industrial applications. In Proceedings of the ARW & OAGM Workshop, 2019, Steyr, Austria, 9–10 May 2019; Volume 2019, pp. 26–31.
22. Iriando, A.; Lazkano, E.; Ansuategi, A.; Rivera, A.; Lluvia, I.; Tubío, C. Learning positioning policies for mobile manipulation operations with deep reinforcement learning. *Int. J. Mach. Learn. Cybern.* **2023**, *14*, 3003–3023. [[CrossRef](#)]
23. Shen, B.; Lin, X.; Xu, G.; Zhou, Y.; Wang, X. A Low Cost Mobile Manipulator for Autonomous Localization and Grasping. In Proceedings of the 2021 5th International Conference on Robotics and Automation Sciences (ICRAS), Wuhan, China, 11–13 June 2021; pp. 193–197.
24. Annusewicz-Mistal, A.; Pietrala, D.S.; Laski, P.A.; Zwierzchowski, J.; Borkowski, K.; Bracha, G.; Borycki, K.; Kostecki, S.; Włodarczyk, D. Autonomous Manipulator of a Mobile Robot Based on a Vision System. *Appl. Sci.* **2022**, *13*, 439. [[CrossRef](#)]
25. Yang, M.; Yang, E. Two-stage multi-sensor fusion positioning system with seamless switching for cooperative mobile robot and manipulator system. *Int. J. Intell. Robot. Appl.* **2023**, *7*, 275–290. [[CrossRef](#)]
26. Sasiadek, J.; Wang, Q. Low cost automation using INS/GPS data fusion for accurate positioning. *Robotica* **2003**, *21*, 255–260. [[CrossRef](#)]
27. Norman, A.R.; Schönberg, A.; Gorlach, I.A.; Schmitt, R. Validation of iGPS as an external measurement system for cooperative robot positioning. *Int. J. Adv. Manuf. Technol.* **2013**, *64*, 427–446. [[CrossRef](#)]
28. Raskaliyev, A.; Patel, S.; Sobh, T. A dynamic model for GPS based attitude determination and testing using a serial robotic manipulator. *J. Adv. Res.* **2017**, *8*, 333–341. [[CrossRef](#)] [[PubMed](#)]
29. Yousif, K.; Bab-Hadiashar, A.; Hoseinnezhad, R. An overview to visual odometry and visual SLAM: Applications to mobile robotics. *Intell. Ind. Syst.* **2015**, *1*, 289–311. [[CrossRef](#)]
30. Kolhatkar, C.; Wagle, K. Review of SLAM algorithms for indoor mobile robot with LIDAR and RGB-D camera technology. In *Innovations in Electrical and Electronic Engineering: Proceedings of ICEEE 2020*; Springer: Singapore, 2021; pp. 397–409.
31. Panigrahi, P.K.; Bisoy, S.K. Localization strategies for autonomous mobile robots: A review. *J. King Saud Univ.-Comput. Inf. Sci.* **2022**, *34*, 6019–6039. [[CrossRef](#)]
32. Marvel, J.; Bostelman, R. Towards mobile manipulator safety standards. In Proceedings of the 2013 IEEE International Symposium on Robotic and Sensors Environments (ROSE), Washington, DC, USA, 21–23 October 2013; pp. 31–36.
33. Zhao, X.; Liu, Z.; Jiang, B.; Gao, C. Switched controller design for robotic manipulator via neural network-based sliding mode approach. *IEEE Trans. Circuits Syst. II Express Briefs* **2022**, *70*, 561–565. [[CrossRef](#)]
34. Srinivasa, S.S.; Ferguson, D.; Helfrich, C.J.; Berenson, D.; Collet, A.; Diankov, R.; Gallagher, G.; Hollinger, G.; Kuffner, J.; Weghe, M.V. HERB: A home exploring robotic butler. *Auton. Robot.* **2010**, *28*, 5–20. [[CrossRef](#)]
35. Štibinger, P.; Broughton, G.; Majer, F.; Rozsypálek, Z.; Wang, A.; Jindal, K.; Zhou, A.; Thakur, D.; Loianno, G.; Krajník, T.; et al. Mobile manipulator for autonomous localization, grasping and precise placement of construction material in a semi-structured environment. *IEEE Robot. Autom. Lett.* **2021**, *6*, 2595–2602. [[CrossRef](#)]
36. Sandakalum, T.; Ang, M.H., Jr. Motion planning for mobile manipulators—A systematic review. *Machines* **2022**, *10*, 97. [[CrossRef](#)]
37. Leonori, M.; Gandarias, J.M.; Ajoudani, A. MOCA-S: A Sensitive Mobile Collaborative Robotic Assistant exploiting Low-Cost Capacitive Tactile Cover and Whole-Body Control. *arXiv* **2022**, arXiv:2202.13401. [[CrossRef](#)]
38. Chen, D.; Liu, Z.; von Wichert, G. Uncertainty-aware arm-base coordinated grasping strategies for mobile manipulation. *J. Intell. Robot. Syst.* **2015**, *80*, 205–223. [[CrossRef](#)]
39. Shao, J.; Xiong, H.; Liao, J.; Song, W.; Chen, Z.; Gu, J.; Zhu, S. Rrt-goalbias and path smoothing based motion planning of mobile manipulators with obstacle avoidance. In Proceedings of the 2021 IEEE International Conference on Real-time Computing and Robotics (RCAR), Xining, China, 15–19 July 2021; pp. 217–222.
40. Weyrer, M.; Brandstötter, M.; Husty, M. Singularity avoidance control of a non-holonomic mobile manipulator for intuitive hand guidance. *Robotics* **2019**, *8*, 14. [[CrossRef](#)]
41. Li, W.; Xiong, R. Dynamical obstacle avoidance of task-constrained mobile manipulation using model predictive control. *IEEE Access* **2019**, *7*, 88301–88311. [[CrossRef](#)]
42. Heins, A.; Schoellig, A.P. Keep it upright: Model predictive control for nonprehensile object transportation with obstacle avoidance on a mobile manipulator. *IEEE Robot. Autom. Lett.* **2023**, *8*, 7986–7993. [[CrossRef](#)]
43. Zhang, H.; Sheng, Q.; Hu, J.; Sheng, X.; Xiong, Z.; Zhu, X. Cooperative transportation with mobile manipulator: A capability map-based framework for physical human–robot collaboration. *IEEE/ASME Trans. Mechatronics* **2022**, *27*, 4396–4405. [[CrossRef](#)]
44. Papadopoulos, E.; Papadimitriou, I.; Poulakakis, I. Polynomial-based obstacle avoidance techniques for nonholonomic mobile manipulator systems. *Robot. Auton. Syst.* **2005**, *51*, 229–247. [[CrossRef](#)]
45. Palmieri, G.; Scoccia, C. Motion planning and control of redundant manipulators for dynamical obstacle avoidance. *Machines* **2021**, *9*, 121. [[CrossRef](#)]
46. Bajrami, A.; Palpacelli, M.C.; Carbonari, L.; Costa, D. Posture Optimization of the TIAGo Highly-Redundant Robot for Grasping Operation. *Robotics* **2024**, *13*, 56. [[CrossRef](#)]

47. Zhang, S.; Cheng, S.; Jin, Z. A control method of mobile manipulator based on null-space task planning and hybrid control. *Machines* **2022**, *10*, 1222. [[CrossRef](#)]
48. Siciliano, B.; Sciavicco, L.; Villani, L.; Oriolo, G. Mobile robots. In *Robotics: Modelling, Planning and Control*; Springer: London, UK, 2009; pp. 469–521.
49. Neri, F.; Forlini, M.; Scoccia, C.; Palmieri, G.; Callegari, M. Experimental evaluation of collision avoidance techniques for collaborative robots. *Appl. Sci.* **2023**, *13*, 2944. [[CrossRef](#)]
50. Neri, F.; Palmieri, G.; Costa, D.; Callegari, M. Dynamic Obstacle Avoidance for Non-holonomic Mobile Robots with Differential Wheels. In *Advances in Italian Mechanism Science, Proceedings of the International Conference of IFToMM ITALY, Turin, Italy, 11–13 September*; Springer: Cham, Switzerland, 2024; pp. 265–272.
51. Maalouf, E.; Saad, M.; Saliah, H. A higher level path tracking controller for a four-wheel differentially steered mobile robot. *Robot. Auton. Syst.* **2006**, *54*, 23–33. [[CrossRef](#)]
52. Kanayama, Y.; Kimura, Y.; Miyazaki, F.; Noguchi, T. A stable tracking control method for an autonomous mobile robot. In *Proceedings of the IEEE International Conference on Robotics and Automation, Cincinnati, OH, USA, 13–18 May 1990*; pp. 384–389.
53. Hu, T.; Wang, T.; Li, J.; Qian, W. Gradient Projection of Weighted Jacobian Matrix Method for Inverse Kinematics of a Space Robot With a Controlled-Floating Base. *J. Dyn. Syst. Meas. Control* **2017**, *139*, 051013. [[CrossRef](#)]
54. Ogawa, S.; Konno, A. Mobile manipulation of a humanoid robot. In *Proceedings of the 2012 IEEE/SICE International Symposium on System Integration (SII), Fukuoka, Japan, 16–18 December 2012*. [[CrossRef](#)]
55. Zhu, Y.; Liu, Z.; Jiang, B.; Zhu, Q. Adaptive fixed-time fuzzy fault-tolerant control for robotic manipulator with unknown friction and composite actuator faults. *J. Frankl. Inst.* **2024**, *361*, 107025. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.