

Article

# Evaluation of the Path-Tracking Accuracy of a Three-Wheeled Omnidirectional Mobile Robot Designed as a Personal Assistant

Jordi Palacín , Elena Rubies, Eduard Clotet and David Martínez

Robotics Laboratory, Universitat de Lleida, Jaume II, 69, 25001 Lleida, Spain; helenarubies@gmail.com (E.R.); eduard.clotet@udl.cat (E.C.); david.martinez@udl.cat (D.M.)

\* Correspondence: palacin@diei.udl.cat

**Abstract:** This paper presents the empirical evaluation of the path-tracking accuracy of a three-wheeled omnidirectional mobile robot that is able to move in any direction while simultaneously changing its orientation. The mobile robot assessed in this paper includes a precise onboard LIDAR for obstacle avoidance, self-location and map creation, path-planning and path-tracking. This mobile robot has been used to develop several assistive services, but the accuracy of its path-tracking system has not been specifically evaluated until now. To this end, this paper describes the kinematics and path-planning procedure implemented in the mobile robot and empirically evaluates the accuracy of its path-tracking system that corrects the trajectory. In this paper, the information gathered by the LIDAR is registered to obtain the ground truth trajectory of the mobile robot in order to estimate the path-tracking accuracy of each experiment conducted. Circular and eight-shaped trajectories were assessed with different translational velocities. In general, the accuracy obtained in circular trajectories is within a short range, but the accuracy obtained in eight-shaped trajectories worsens as the velocity increases. In the case of the mobile robot moving at its nominal translational velocity, 0.3 m/s, the root mean square (RMS) displacement error was 0.032 m for the circular trajectory and 0.039 m for the eight-shaped trajectory; the absolute maximum displacement errors were 0.077 m and 0.088 m, with RMS errors in the angular orientation of 6.27° and 7.76°, respectively. Moreover, the external visual perception generated by these error levels is that the trajectory of the mobile robot is smooth, with a constant velocity and without perceiving trajectory corrections.

**Keywords:** omnidirectional mobile robot; omnidirectional wheel; path tracking; kinematic model; inverse kinematic model



**Citation:** Palacín, J.; Rubies, E.; Clotet, E.; Martínez, D. Evaluation of the Path-Tracking Accuracy of a Three-Wheeled Omnidirectional Mobile Robot Designed as a Personal Assistant. *Sensors* **2021**, *21*, 7216. <https://doi.org/10.3390/s21217216>

Academic Editor: Andrey V. Savkin

Received: 12 September 2021

Accepted: 28 October 2021

Published: 29 October 2021

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2021 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 popularity of vehicles using an omnidirectional motion system in the field of robotics is on the rise [1–3]. An omnidirectional motion is one of the principal requirements for mobile robots designed to operate in complex and unstructured environments in order to develop services such as workshop assistance [4], domestic [5] or home assistance [6] and health-care assistance [7]. The main benefit of an omnidirectional motion system is that it provides three degrees of freedom (DOFs) in a ground plane, allowing displacements in any direction while changing its orientation. This omnidirectional mobility is usually achieved with the use of three or four omnidirectional wheels.

The main drawback of using a mobile robot with omnidirectional wheels is the evaluation of the odometry because the odometry is highly influenced by the practical implementation details such as the mechanical performances of the omnidirectional wheels, the accuracy of the direct and inverse kinematic models, the electrical performances of the motors used to drive the wheels, the rotary encoders used to estimate the angular rotational velocity of the wheels [8] and the tuning of the motor controllers. For example, Tsai et al. [9] and Tri et al. [10] proposed the implementation of omnidirectional mobile robots using

three double-line omnidirectional wheels (also known as double parallel wheels), easy to manufacture but with the drawback of having two radial distances relative to the center of the mobile robot depending on the inner or outer parallel wheel in effective contact with the ground. In this case, this duality originates the existence of eight different kinematic models and additional mechanical constraints during a displacement. Nevertheless, the model complexity originated by the use of double-line parallel wheels and other similar wheels is usually addressed by computing an average radial distance for the wheel. For example, Lin et al. [11] addressed the trajectory errors originated by a mobile robot using double-line omnidirectional wheels by implementing a calibration procedure. In a similar direction, Maddahi et al. [12] proposed a calibration procedure applied to a lightweight mobile robot using three simple single-line omnidirectional wheels (also known as omnidirectional wheels with multiple passive rollers). In this case, this procedure assumes that the motors of the mobile robot are able to reach instantaneously the target angular rotational speed. The importance of the control of the motors used to drive the wheels of an omnidirectional mobile robot was specifically highlighted and analyzed by Li et al. [13] and by Tri et al. [10] in the case of using several omnidirectional configurations. Finally, there are many alternatives that can be used to control the trajectory of an omnidirectional mobile robot such as the use of PID controllers [14], the use of a model predictive control strategy (MPCS) [15] or the use of the anisotropic characteristics of the mobile robot [16]. Alternatively, the motion control implemented in the mobile robot assessed in this paper is based on the deterministic computation of the motion command required to reach the next discrete position and orientation.

#### *New Contribution*

This paper is inspired by the work of Li et al. [13] that presented the simulation of some trajectories performed by a three-wheeled omnidirectional mobile robot and also analyzed the limitations found in a real implementation of these trajectories. In this direction, the new contribution of this paper is the experimental evaluation of the path-tracking accuracy of the omnidirectional mobile robot designed at the University of Lleida [17] to operate as an assistant personal robot (APR). This mobile robot has been used to develop several assistive services, but the accuracy of its path-tracking system has not been specifically evaluated until now.

Previously, Moreno et al. [18] analyzed in 2016 the basic motion performance of the first mobile robot prototype designed as an APR (named APR-01 [17]). This first prototype was designed for robotic services requiring only remote manual tele-control and did not have path-planning and path-following capabilities as it was not planned to move autonomously. The experimentation with the APR-01 fostered the development of a second improved prototype (named APR-02) designed to fully operate autonomously in unstructured environments. The main improvement of the APR-02 prototype was the development of a path-planning and path-following procedure that has been widely used to develop complex operations such as gas leakage detection [19].

The new contribution of this paper is, then, the empirical evaluation of the path-tracking accuracy of the omnidirectional mobile robot APR-02 in terms of the maximum absolute error and the root mean square (RMS) error of the planned position of the mobile robot and of the planned angular orientation of the mobile robot, which is not usually evaluated. The particularity of the APR-02 is the application of a path-following procedure based on the information gathered by its precise onboard LIDAR. This information was also registered to obtain the ground truth trajectory of the mobile robot in order to estimate the path-tracking accuracy of each experiment conducted. The path-planning procedure implemented was based on the deterministic computation of the motion command required to reach the next planned mobile robot position and orientation. Finally, the accuracy of the resulting path-tracking performances was evaluated following standard target trajectories and moving at different translational velocities.

## 2. The Mobile Robot APR

Figure 1 shows an image of the mobile robot APR-02 ready to initiate an autonomous exploration. The APR-02 is an indoor omnidirectional mobile robot prototype developed by the Robotics Laboratory of the University of Lleida [17]. The mobile robot APR-02 is a tall and thin ( $1700 \times 55$  mm) indoor humanoid mobile robot with a touch screen monitor as a head, two simple thin arms with four degrees of freedom (DOFs) and a decorative hand. The mobile robot has a weight of 31 kg and a hexagonal base with a diameter of 540 mm, with structural pieces made of aluminum and non-structural pieces made of PLA and ABS using 3D printing. The omnidirectional motion system is composed of three omnidirectional wheels optimized to operate on flat floors [18]. This mobile robot has been designed to fulfill the requirement to operate in indoor collaborative domestic environments [20] and also in indoor industrial environments [21]. The mobile robot APR-02 has been optimized during several years to develop applications requiring autonomous navigation [19,22] based on the information gathered by an onboard precise LIDAR (UTM-30LN with  $270^\circ$ , 1.081 scan points, 40 Hz scan rate, 30 m range and individual scan precision from  $\pm 10$  to  $\pm 50$  mm) and SLAM [23]. At this moment, the next planned evolution of the APR prototypes is the development of outdoor applications based on the use of all-terrain omnidirectional wheels [24] and push-broom LIDARs [25] to improve the detection of obstacles in outdoor environments.



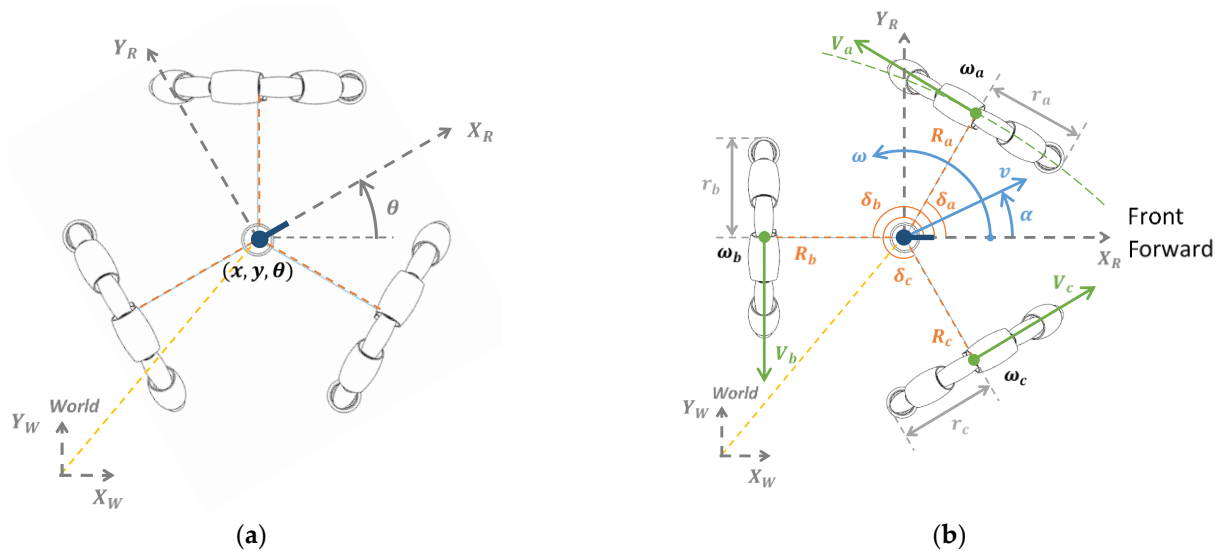
**Figure 1.** Image of one of the authors of the paper preparing the mobile robot APR-02 for an autonomous exploration under COVID-19 restrictions.

The omnidirectional motion system used in the family of APR mobile robots was described in Moreno et al. [18]. This motion system is based on three optimal omnidirectional wheels shifted  $120^\circ$  from each other (also known as a kiwi drive): the distance between the center of the robot and each wheel ( $R_a, R_b, R_c$ ) is 195 mm, and the radius of the wheels ( $r_a, r_b, r_c$ ) is 148 mm. The design of the omnidirectional wheels is considered optimal because the transversal rotating rollers have a minimized gap distance. The omnidirectional wheels are driven by geared brushed DC motors with a low-cost magnetic rotary encoder attached. The estimation of the angular rotational speed of the DC motors is performed by measuring and processing the pulse length of the digital signal provided by the encoder [8]. The omnidirectional motion system can move the robot up to 1.0 m/s in any direction, although 0.3 m/s is the nominal translational velocity used in most of its applications [19].

## 3. Kinematics of the Omnidirectional Mobile Robot APR

Figure 2a,b illustrate the reference frames and parameters of the omnidirectional motion system of the mobile robot APR used to define the kinematics of the mobile robot. Figure 2a shows the position and orientation of the center of the omnidirectional mobile robot in the world reference frame ( $X_W, Y_W$ ), referenced as  $(x, y, \theta)$ . The value of  $(x, y)$  represents the translation of the mobile robot reference frame ( $X_R, Y_R$ ) relative to the fixed

world reference frame  $(X_W, Y_W)$ , whereas  $\theta$  is the angular rotation of the reference frame of the mobile robot  $(X_R, Y_R)$  relative to the fixed world reference frame  $(X_W, Y_W)$ . Figure 2b represents the structural parameters  $(R_a, R_b, R_c)$ , which are the radial distance (in m) of the wheels relative to the center of the mobile robot, and  $(\delta_a, \delta_b, \delta_c)$  are the angular orientation (in arc degrees) of each omnidirectional wheel, relative to the mobile robot reference frame  $(X_R, Y_R)$ .



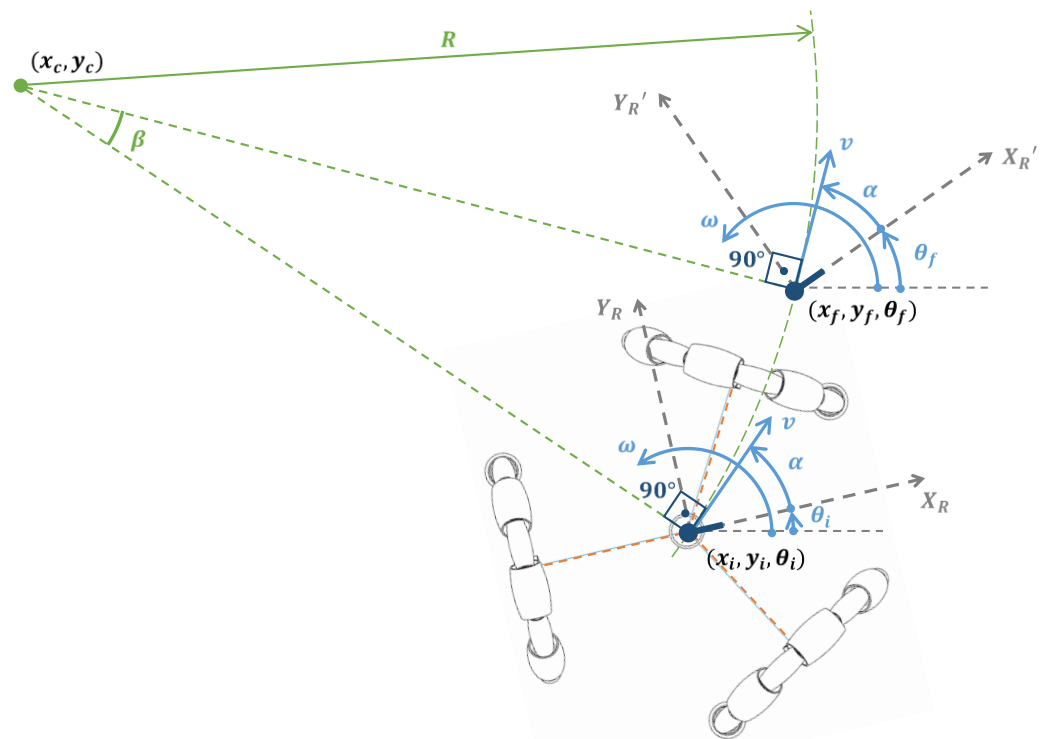
**Figure 2.** (a) Representation of the location of the center of the mobile robot  $(x, y)$  and absolute angular orientation  $(\theta)$  of the omnidirectional mobile robot relative to the world reference frame  $(X_W, Y_W)$ . The reference frame  $(X_R, Y_R)$  is the mobile frame of the mobile robot, where the axis  $X_R$  depicts the front of the mobile robot. (b) Representation of the parameters of a motion vector  $(v, \alpha, \omega)$ .  $(R_a, R_b, R_c)$  are the radial distances of each omnidirectional wheel relative to the center of the mobile robot,  $(\delta_a, \delta_b, \delta_c)$  are the angular orientations of the wheels relative to the mobile robot reference frame  $(X_R, Y_R)$ , and  $(\omega_a, \omega_b, \omega_c)$  are the representation of the angular rotational speed of the wheels.

The motion and trajectory of an omnidirectional mobile robot is defined by a basic motion vector,  $M = (v, \alpha, \omega)$ , where  $v$  is the translational velocity of the displacement (in m/s),  $\alpha$  is the angular orientation of the displacement (in arc degrees) relative to the robot reference frame  $(X_R, Y_R)$ , and  $\omega$  is the angular rotational speed (in rad/s) applied to the center of the omnidirectional mobile robot. This basic motion vector is implemented in the APR mobile robots as a motion command using  $M = (v, \alpha, \omega, t_r)$  or  $M = (v, \alpha, \omega, d_r)$ , where  $t_r$  is the relative duration of the displacement (in s), and  $d_r$  is the relative distance of the displacement (in m), with  $v = d_r/t_r$ .

Figure 2b represents graphically the motion vector  $M = (v, \alpha, \omega)$  that originates the translation and rotation of the mobile robot reference frame  $(X_R, Y_R)$ . The translational velocity  $v$ , the angular orientation of the mobile robot  $\alpha$  and the angular rotational speed of the center of the mobile robot  $\omega$  are linked to the angular rotational speed of the wheels  $(\omega_a, \omega_b, \omega_c)$  (in rad/s), the radius of each wheel  $(r_a, r_b, r_c)$  (in m), the computed translational velocities of the wheels  $(V_a, V_b, V_c)$  (in m/s), the radial distance from the wheels to the center of rotation of the mobile robot  $(R_a, R_b, R_c)$  and the relative orientation of the wheels  $(\delta_a, \delta_b, \delta_c)$ . The conventional assumption is that the triplet of the structural parameters of the mobile robot does not change during its normal operation:  $r_a = r_b = r_c = 148$  mm,  $R_a = R_b = R_c = 195$  mm,  $\delta_a = 60^\circ$ ,  $\delta_b = 180^\circ$ , and  $\delta_c = 300^\circ$ . Then, the implementation of one specific motion  $(v_k > 0, \alpha_k, \omega_k)$  requires a unique combination of angular rotational speeds of the three wheels  $(\omega_{ak}, \omega_{bk}, \omega_{ck})$ , and one specific combination of angular rotational speeds of the three wheels  $(\omega_{ak}, \omega_{bk}, \omega_{ck})$  generates a unique motion in the mobile robot  $(v_k, \alpha_k, \omega_k)$ .

### 3.1. Motion Originated by the Execution of a Single Motion Command $M = (v, \alpha, \omega, t_r)$

Figure 3 is a representation of the kinematics of an ideal omnidirectional mobile robot with known initial position and orientation  $(x_i, y_i, \theta_i)$  and a known single motion command  $M = (v, \alpha, \omega, t_r)$  applied to the mobile robot. As a consequence of executing this motion command, an ideal omnidirectional mobile robot will move for a time,  $t_r$ , with a translational velocity,  $v$ , starting the displacement in the relative angular direction defined by  $\alpha + \theta_i$ , and with the center of the mobile robot rotating depending on the angular rotational velocity,  $\omega$ , defined.



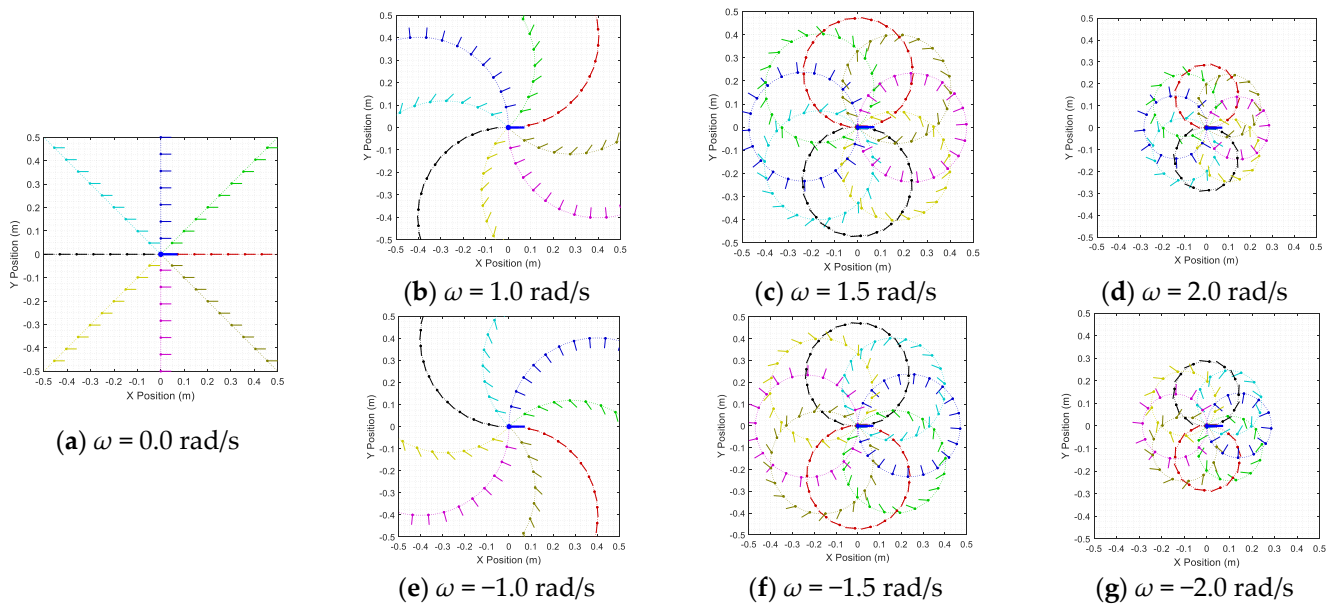
**Figure 3.** Representation of the displacement of an omnidirectional mobile robot located at  $(x_i, y_i, \theta_i)$  when executing a motion command  $M = (v, \alpha, \omega, t_r)$ .  $R$  is the radius of the circular trajectory,  $(x_c, y_c)$  is the location of the center of the circular trajectory,  $\beta$  is the angular displacement of the robot along the circular path, and  $(x_f, y_f, \theta_f)$  is the final position of the robot.

The final position of the mobile robot  $(x_f, y_f, \theta_f)$  in the world frame after implementing a motion command  $M = (v, \alpha, \omega, t_r)$  during a time  $t_r$  is deterministic. The distance traveled with this motion command is:

$$d = v \cdot t_r \quad (1)$$

In the case of  $\omega \neq 0$ , during the time  $t_r$  required to execute the motion command  $M = (v, \alpha, \omega, t_r)$ , the displacement of an omnidirectional mobile robot describes always a circular path (see Figure 4) whose radius  $R$  is computed using:

$$R = \frac{v}{|\omega|} \quad (2)$$



**Figure 4.** Simulation of the trajectories generated by motion commands  $M = (v, \alpha, \omega, t_r = 10s)$ . The starting point is  $(x = 0, y = 0, \theta = 0)$ , and the velocity  $v = 0.3$  m/s. Showing 8 different angular orientations  $\alpha$ :  $0^\circ$  (red),  $45^\circ$  (green),  $90^\circ$  (blue),  $135^\circ$  (cyan),  $180^\circ$  (black),  $225^\circ$  (yellow),  $270^\circ$  (magenta),  $315^\circ$  (olive), and different angular rotational speeds: (a)  $\omega = 0.0$  rad/s, (b)  $\omega = 1.0$  rad/s, (c)  $\omega = 1.5$  rad/s, (d)  $\omega = 2.0$  rad/s, (e)  $\omega = -1.0$  rad/s, (f)  $\omega = -1.5$  rad/s, (g)  $\omega = -2.0$  rad/s.

The location of the center of this circular path (valid during the time  $t_r$ ) is computed using:

$$x_c = -\frac{v}{\omega} \cdot \sin(\alpha + \theta_i) + x_i \quad (3)$$

$$y_c = \frac{v}{\omega} \cdot \cos(\alpha + \theta_i) + y_i \quad (4)$$

The localization of the center of the circular path depends on the counterclockwise ( $\omega > 0$ ) or clockwise ( $\omega < 0$ ) direction of the angular rotational speed applied to the center of the mobile robot.

The final position of the mobile robot in the world frame  $(x_f, y_f, \theta_f)$  is computed based on the angular displacement  $\beta$  along the circular path using:

$$\beta = (\omega \cdot t_r) \omega \frac{360^\circ}{2\pi} \quad (5)$$

$$x_f = \frac{v}{\omega} \cdot \sin(\theta_i + \alpha + \beta) + x_c \quad (6)$$

$$y_f = -\frac{v}{\omega} \cdot \cos(\theta_i + \alpha + \beta) + y_c \quad (7)$$

$$\theta_f = \theta_i + \omega \cdot t_r \cdot \frac{360^\circ}{2\pi} \quad (8)$$

Alternatively, in the case of  $\omega = 0$  (no angular rotational speed), the radius,  $R$ , of the circular path becomes infinite and the mobile robot moves only straight (see Figure 4). Then, the final position of the mobile robot in the world frame  $(x_f, y_f, \theta_f)$  can be computed using:

$$x_f = x_i + v \cdot t_r \cdot \cos(\theta_i + \alpha) \quad (9)$$

$$y_f = y_i + v \cdot t_r \cdot \sin(\theta_i + \alpha) \quad (10)$$

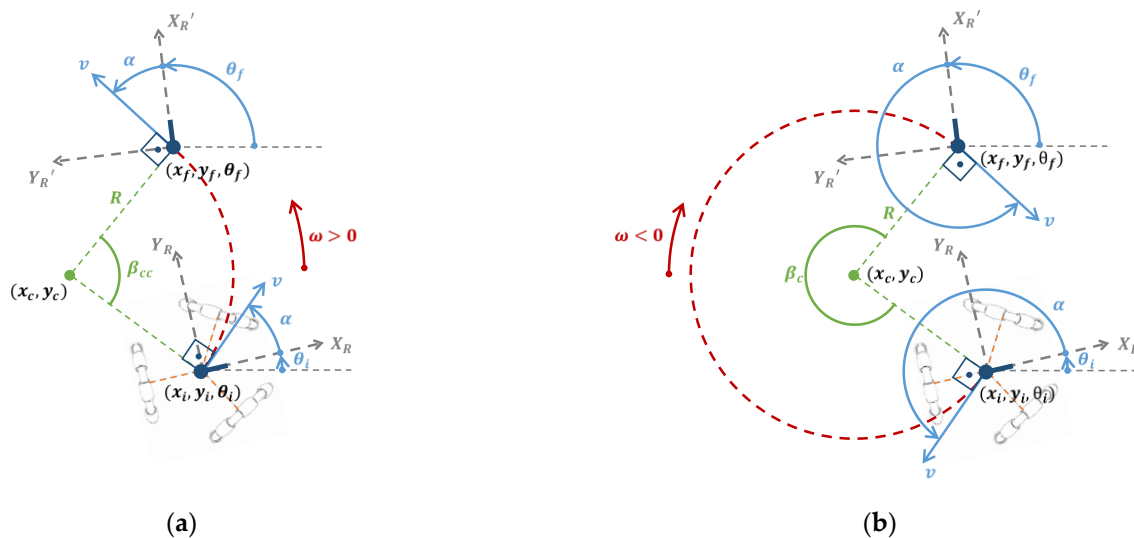
$$\theta_f = \theta_i \quad (11)$$

Figure 4 shows a simulation of the trajectories of an ideal omnidirectional mobile robot executing one specific motion command  $M = (v, \alpha, \omega, t_r)$  during a long execution time:

$t_r = 10$  s. Each trajectory of the mobile robot is represented with a thin line while the bold point and bold line are sparse representations of the position and orientation of the mobile robot during the displacement. Each particular trajectory is depicted with an identifying color. In all the motion commands simulated, the starting position of the mobile robot is the same ( $x_i = 0, y_i = 0, \theta_i = 0$ ), and the duration of the displacement  $t_r$  and the translational velocity  $v$  of the motion are also the same in all cases. The motion commands represented in each plot have angular orientations  $\alpha$  from  $0$  to  $315^\circ$  in increments of  $45^\circ$  (labeled with the same identifying color). Each plot represents a different angular rotational speed  $\omega$  from  $0$  to  $2.0$  rad/s and from  $0$  to  $-2.0$  rad/s. In Figure 4a, case  $\omega = 0$ , the mobile robot performs straight displacements in the direction established by the value of the angular orientation  $\alpha$ . The other cases are for  $\omega \neq 0$ , and then the mobile robot describes a circular trajectory defined by the value of the radius  $R$ , which is lower as  $\omega$  increases. A common characteristic of the basic trajectories shown in Figure 4 is that the orientation of the mobile robot relative to the tangent of the trajectory is constant during the motion, a typical feature of omnidirectional mobile robots.

### 3.2. Estimation of the Motion Command to Reach a Target Position $(x_f, y_f, \theta_f)$ for a Known $v$

Alternatively to the previous section, Figure 5 is a representation of the kinematics of an omnidirectional mobile robot with a known initial position  $(x_i, y_i, \theta_i)$ , known final destination  $(x_f, y_f, \theta_f)$  and known target translational velocity  $v$  of the motion. The unknown parameters of the motion command  $M = (v, \alpha, \omega, t_r)$  required to reach the final destination are  $\alpha, \omega$  and  $t_r$ , and the determination of these parameters is also deterministic.



**Figure 5.** Representation of the trajectory of the mobile robot: (a) case with translation and rotation in the counterclockwise direction ( $\omega > 0$ ) and (b) case with translation and rotation in the clockwise direction ( $\omega < 0$ ).

There are four possible case trajectories depending on the value of the target angular rotational speed  $\omega$ : (1) translation and rotation in the counterclockwise direction ( $\omega > 0$ ) (Figure 5a), (2) translation and rotation in the clockwise direction ( $\omega < 0$ ) (Figure 5b), (3) translation without rotation ( $v \neq 0, \omega = 0$ ) and (4) rotation without translation ( $v = 0, \omega \neq 0$ ).

#### 3.2.1. Translation and Rotation in the Counterclockwise Direction ( $\omega > 0$ ) for a Known $v$

Figure 5a depicts the trajectory of an ideal omnidirectional mobile robot (red dotted line) with a known initial position  $(x_i, y_i, \theta_i)$ , known final position, known translational velocity  $v$  and a target counterclockwise trajectory condition ( $\omega > 0$ ). The unknown parameters of the motion are the values of  $\alpha, \omega$  and  $t_r$ . The counterclockwise condition

( $\omega > 0$ ) means that the angular orientation of the mobile robot will increase from  $\theta_i$  to  $\theta_f$ . In the case of  $\theta_i > \theta_f$ , the trajectory must go from  $\theta_i$  to  $\theta_f + 360^\circ$ .

In this case, the angle covered by the circular trajectory  $\beta_{cc}$  and the radius  $R$  of the circular trajectory are computed as:

$$\begin{aligned} & \text{if } (\theta_f > \theta_i) \text{ then } \beta_{cc} = \theta_f - \theta_i \\ & \text{if } (\theta_f < \theta_i) \text{ then } \beta_{cc} = (\theta_f - \theta_i) + 360^\circ \end{aligned} \quad (12)$$

$$R = \sqrt{\frac{(X_f - X_i)^2 + (Y_f - Y_i)^2}{2 \cdot (1 - \cos(\beta_{cc}))}} \quad (13)$$

The angular velocity required to complete this counterclockwise motion is:

$$\omega = \frac{v}{R} \quad (14)$$

The angular orientation of the velocity vector required is:

$$\alpha = \frac{180 - \beta_{cc}}{2} + \text{atan}\left(\frac{Y_f - Y_i}{X_f - X_i}\right) - \theta_i - 90^\circ \quad (15)$$

Finally, the exact time required to complete this displacement is:

$$tr = \frac{\beta_{cc}}{\omega} \quad (16)$$

### 3.2.2. Translation and Rotation in the Clockwise Direction ( $\omega < 0$ ) for a Known $v$

Alternatively to the previous case, Figure 5b depicts the trajectory of an ideal omnidirectional mobile robot (red dotted line) with a known initial position  $(x_i, y_i, \theta_i)$ , known final destination  $(x_f, y_f, \theta_f)$ , known target translational velocity  $v$  and a target clockwise trajectory condition ( $\omega < 0$ ). Then the unknown parameters of the motion are the values of  $\alpha$ ,  $\omega$  and  $t_r$ .

In this case, the angle covered by the circular trajectory  $\beta_c$  and the radius  $R$  of the circular trajectory are computed as:

$$\begin{aligned} & \text{if } (\theta_f > \theta_i) \text{ then } \beta_c = (\theta_f - \theta_i) - 360^\circ \\ & \text{if } (\theta_f < \theta_i) \text{ then } \beta_c = \theta_f - \theta_i \end{aligned} \quad (17)$$

$$R = \sqrt{\frac{(X_f - X_i)^2 + (Y_f - Y_i)^2}{2 \cdot (1 - \cos(\beta_c))}} \quad (18)$$

The angular velocity required to complete this clockwise motion is:

$$\omega = -\frac{v}{R} \quad (19)$$

The angular orientation of the velocity vector required is:

$$\alpha = -\left(\frac{180 + \beta_c}{2}\right) - \text{atan}\left(\frac{X_i - X_f}{Y_i - Y_f}\right) - \theta_i \quad (20)$$

Finally, the time required to complete this displacement is again:

$$tr = \frac{\beta_c}{\omega} \quad (21)$$



### 3.2.3. Translation without Rotation ( $v \neq 0, \omega = 0$ ) for a Known $v$

When the initial and final angular orientations of the omnidirectional mobile robot are the same ( $\theta_i = \theta_f$ ), the angular rotational speed of the motion command is zero ( $\omega = 0$ ), and then the trajectory of the mobile robot defines a straight line. The unknown parameters of the motion command are the values of  $\alpha$  and  $t_r$ . In this special case, the distance traveled  $d$  during this translation is computed with:

$$d = \sqrt{(X_f - X_i)^2 + (Y_f - Y_i)^2} \quad (22)$$

and the angular orientation of the velocity  $\alpha$  required to complete this motion is:

$$\alpha = \text{atan}\left(\frac{Y_f - Y_i}{X_f - X_i}\right) - \theta_i \quad (23)$$

Then, the time required to complete this translation is:

$$t_r = \frac{d}{v} \quad (24)$$

### 3.2.4. Static Rotation without Translation ( $v = 0, \omega \neq 0$ )

The last alternative is a special case that defines a simple static rotation of the mobile robot without any additional translation, and thus the linear translational velocity  $v$  must be zero. In this static rotation, the angular orientation of the translational speed  $\alpha$  has no effect on the motion; therefore its value is indifferent. This static rotation depends on the angular rotational speed  $\omega$  and the target final angular orientation of the mobile robot  $\theta_f$ ; thus these parameters must be defined, and then the unknown parameter of the motion is only the value of  $t_r$ . In this static case, the increment of the angular position ( $\beta$ ) can be calculated as in Section 3.2.3 and Section 3.2.4: according to the direction of rotation (counterclockwise, ( $\omega > 0$ ), or clockwise, ( $\omega < 0$ )) and depending on the initial and final angular positions ( $\theta_i > \theta_f$  or  $\theta_i < \theta_f$ ).

Nevertheless, in practice, this special case can be highly simplified by defining the angular rotational speed of the rotation  $\omega$  and an additional parameter that is the relative increment of the angular orientation of the mobile robot  $\beta$ . Then, the sign of the angular rotation  $\beta$  directly defines the direction of rotation (instead of the sign of  $\omega$ ). Therefore, if  $\beta > 0$ , the robot must rotate in the counterclockwise direction (requiring  $\omega > 0$ ), and if  $\beta < 0$ , the robot must rotate in the clockwise direction (requiring  $\omega < 0$ ). Finally, the time required to complete the rotation based on this definition is:

$$t_r = \frac{|\beta|}{|\omega| \cdot \frac{360^\circ}{2\pi}} \quad (25)$$

The angular rotational speed of the angular rotation  $\omega$  applied internally by the control system of the mobile robot must be:

$$\omega = \text{sign}(\beta) \cdot |\omega| \quad (26)$$

### 3.3. Kinematic Model: Determination of $(\omega_a, \omega_b, \omega_c)$ from $(v, \alpha, \omega)$

The analysis of the kinematic model of the omnidirectional motion system allows the determination of the angular rotational speeds of the three omnidirectional wheels  $(\omega_a, \omega_b, \omega_c)$  required to implement a specific motion command  $M = (v, \alpha, \omega)$ . The kinematic model of the motion system of the mobile robot APR was described in Moreno et al. [18] and can also be summarized as follows [9] (see Figure 3).

$$v_x = v \cdot \cos(\alpha) \quad (27)$$

$$v_y = v \cdot \sin(\alpha) \quad (28)$$

$$IK = \begin{bmatrix} -\sin(\delta_a) & \cos(\delta_a) & R_a \\ -\sin(\delta_b) & \cos(\delta_b) & R_b \\ -\sin(\delta_c) & \cos(\delta_c) & R_c \end{bmatrix} \quad (29)$$

$$\begin{bmatrix} V_a \\ V_b \\ V_c \end{bmatrix} = IK \cdot \begin{bmatrix} v_x \\ v_y \\ \omega \end{bmatrix}_{Robot} \quad (30)$$

$$\begin{bmatrix} \omega_a \\ \omega_b \\ \omega_c \end{bmatrix} = \begin{bmatrix} V_a \\ V_b \\ V_c \end{bmatrix} \cdot \begin{bmatrix} \frac{1}{r_a} & \frac{1}{r_b} & \frac{1}{r_c} \end{bmatrix} \quad (31)$$

where  $(V_a, V_b, V_c)$  are the linear speed of each wheel (in m/s) and  $(\omega_a, \omega_b, \omega_c)$  the angular velocities of each wheel (in rad/s). The brushed DC motors that drive the wheels of the mobile robot have a gear ratio of 64:1, and the internal PID controllers of the motion control board use targets defined in rpm. Therefore, the target angular rotational speeds of the motors  $(\omega_{MA}, \omega_{MB}, \omega_{MC})$  (defined in rpm) are:

$$\begin{bmatrix} \omega_{MA} \\ \omega_{MB} \\ \omega_{MC} \end{bmatrix} = \begin{bmatrix} \omega_a \\ \omega_b \\ \omega_c \end{bmatrix} \cdot \frac{2\pi}{360^\circ} \cdot \frac{64}{1} \quad (32)$$

### 3.4. Kinematic Model: Determination of $(v, \alpha, \omega)$ from $(\omega_a, \omega_b, \omega_c)$

The analysis of the kinematic model of the omnidirectional motion system allows the determination of the motion  $(v, \alpha, \omega)$  based on the angular rotational speed of the three omnidirectional wheels  $(\omega_a, \omega_b, \omega_c)$ . The computation of this kinematic model was described in Moreno et al. [18]. The angular rotational speed of the wheels is deduced from the information gathered by the rotary encoders attached to the geared brushed DC motors. Therefore, the estimate of the angular rotational speed of the motors  $(\omega_{MA}, \omega_{MB}, \omega_{MC})$  can be converted to the angular rotational speed of the wheels  $(\omega_a, \omega_b, \omega_c)$  using:

$$\begin{bmatrix} \omega_a \\ \omega_b \\ \omega_c \end{bmatrix} = \begin{bmatrix} \omega_{MA} \\ \omega_{MB} \\ \omega_{MC} \end{bmatrix} \cdot \frac{360^\circ}{2\pi} \cdot \frac{1}{64} \quad (33)$$

Then, the translational velocity of the center of the mobile robot referred to the mobile robot reference frame  $(v_x, v_y)$  and its angular rotational speed  $\omega$  can be computed by inverting the inverse kinematic matrix of the mobile robot. In the case of omnidirectional wheels with the same radius  $r$ :

$$\begin{bmatrix} v_x \\ v_y \\ \omega \end{bmatrix}_{Robot} = IK^{-1} \cdot \begin{bmatrix} \omega_a \\ \omega_b \\ \omega_c \end{bmatrix} \cdot r \quad (34)$$

where the linear translational velocity,  $v$ , of the mobile robot and the angular orientation of the translational velocity,  $\alpha$ , referred to the mobile reference frame are computed using:

$$v = \sqrt{v_x^2 + v_y^2} \quad (35)$$

$$\alpha = \tan^{-1}\left(\frac{v_y}{v_x}\right) \quad (36)$$

Finally, in an ideal implementation of the omnidirectional mobile robot structure with an exact mechanical placement of the wheels ( $R = R_a = R_b = R_c$  and  $\delta_a = 60^\circ$ ,  $\delta_b = 180^\circ$ ,  $\delta_c = 300^\circ$ ) this matrix  $IK^{-1}$  is:

$$IK^{-1} = \begin{bmatrix} -\frac{1}{\sqrt{3}} & 0 & \frac{1}{\sqrt{3}} \\ \frac{1}{3} & -\frac{2}{3} & \frac{1}{3} \\ \frac{1}{3R} & \frac{1}{3R} & \frac{1}{3R} \end{bmatrix} \quad (37)$$

### 3.5. Odometry: Determination of $(\Delta x, \Delta y, \Delta\theta)$ from $(\omega_a, \omega_b, \omega_c)$

Odometry is the use of the information of the rotation of the wheels of a mobile robot to estimate their position relative to a starting location. This method is very effective, but it is also very sensitive to errors because it is based on the discrete integration of the information of the velocity of the wheels [26,27]. The odometry used in the omnidirectional mobile APR-02 is based on the kinematic analysis described in the previous Section 3.4. This kinematic model converts the information of the angular rotational speed of the wheels of the omnidirectional mobile robot  $(\omega_a, \omega_b, \omega_c)$  into an estimation of the motion of the mobile robot  $(v_x, v_y, \omega)$  relative to the robot reference frame. Then, this motion can be combined with the previous known position of the mobile robot  $(x_i, y_i, \theta_i)$  in order to update the current location of the mobile robot.

Typically, the odometry of a mobile robot provides a new update of the estimation of the position of the robot  $(x, y, \theta)$  in the world reference frame after a fixed time lapse  $\Delta T$ . This new position of the robot can be computed using the expressions provided in Section 3.1 or directly applying a compact transformation matrix. In this second case, the velocity of the center of the robot  $(v_x, v_y)$  relative to the mobile robot reference frame is converted into the world reference frame according to the previous angular orientation of the mobile robot,  $\theta_i$ :

$$\begin{bmatrix} v_X \\ v_Y \\ \omega \end{bmatrix}_{World} = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) & 0 \\ \sin(\theta_i) & \cos(\theta_i) & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} v_x \\ v_y \\ \omega \end{bmatrix}_{Robot} \quad (38)$$

Since the next position of the robot is calculated after a time lapse  $\Delta T$ , this new estimate of  $(v_x, v_y, \omega)$  is an average value computed during the time lapse  $\Delta T$ . Then, the general assumption is that  $\Delta T$  is small enough and the information provided by the rotary encoders is precise enough in order to consider the new value of  $(v_x, v_y, \omega)$  as representative of the motion of the mobile robot. Finally, the displacement and new position of the mobile robot relative to the world reference frame is computed using:

$$\begin{bmatrix} \Delta x \\ \Delta y \\ \Delta\theta \end{bmatrix}_{World} = \begin{bmatrix} v_X \\ v_Y \\ \omega \end{bmatrix}_{World} \cdot \Delta T \quad (39)$$

$$\begin{bmatrix} x_f \\ y_f \\ \theta_f \end{bmatrix}_{World} = \begin{bmatrix} x_i \\ y_i \\ \theta_i \end{bmatrix}_{World} + \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta\theta \end{bmatrix}_{World} \quad (40)$$

## 4. Path Planning and Path Following

### 4.1. Rough Trajectory Definition through Waypoints

The definition of the trajectory of the mobile robot APR from a starting point  $(x_i, y_i, \theta_i)$  to an ending point  $(x_f, y_f, \theta_f)$  is based on the definition of one or several intermediate trajectory waypoints  $(x_w, y_w)$ . These waypoints can be manually generated over a map of the application scenario [19], for example, by direct indication of the destination  $(x_f, y_f)$ , by direct indication of some intermediate destinations  $(x_w, y_w)$  or indicating the intermediate destination and the desired mobile robot orientation  $(x_w, y_w, \theta_w)$ . In the cases of having

only a final destination  $(x_f, y_f)$ , the sequence of intermediate destinations  $(x_w, y_w)$  can be automatically obtained by using the A\* (A-star) algorithm [19,28] or by the application of an artificial potential field algorithm [16].

#### 4.2. Path Planning: Linearizing and Smoothing the Trajectory

The path-planning procedure used in the mobile robot APR-02 consists of linearizing and smoothing the trajectory defined by the waypoints  $(x_w, y_w)$  with the application of splines using a constant distance interval [29–32]. The result of this spline interpolation is a fine sequence of intermediate trajectory points  $(x_k, y_k, \theta_k)$  that the robot must follow in order to move from the starting point  $(x_i, y_i, \theta_i)$  to the final destination  $(x_f, y_f, \theta_f)$ :

$$(x_i, y_i, \theta_i) \rightarrow \dots \rightarrow (x_k, y_k, \theta_k) \rightarrow (x_{k+1}, y_{k+1}, \theta_{k+1}) \rightarrow \dots \rightarrow (x_f, y_f, \theta_f) \quad (41)$$

The motion capabilities of an omnidirectional motion system allow an uncorrelated or independent definition of the intermediate trajectory positions  $(x_k, y_k)$  and the mobile robot orientations  $(\theta_k)$ :

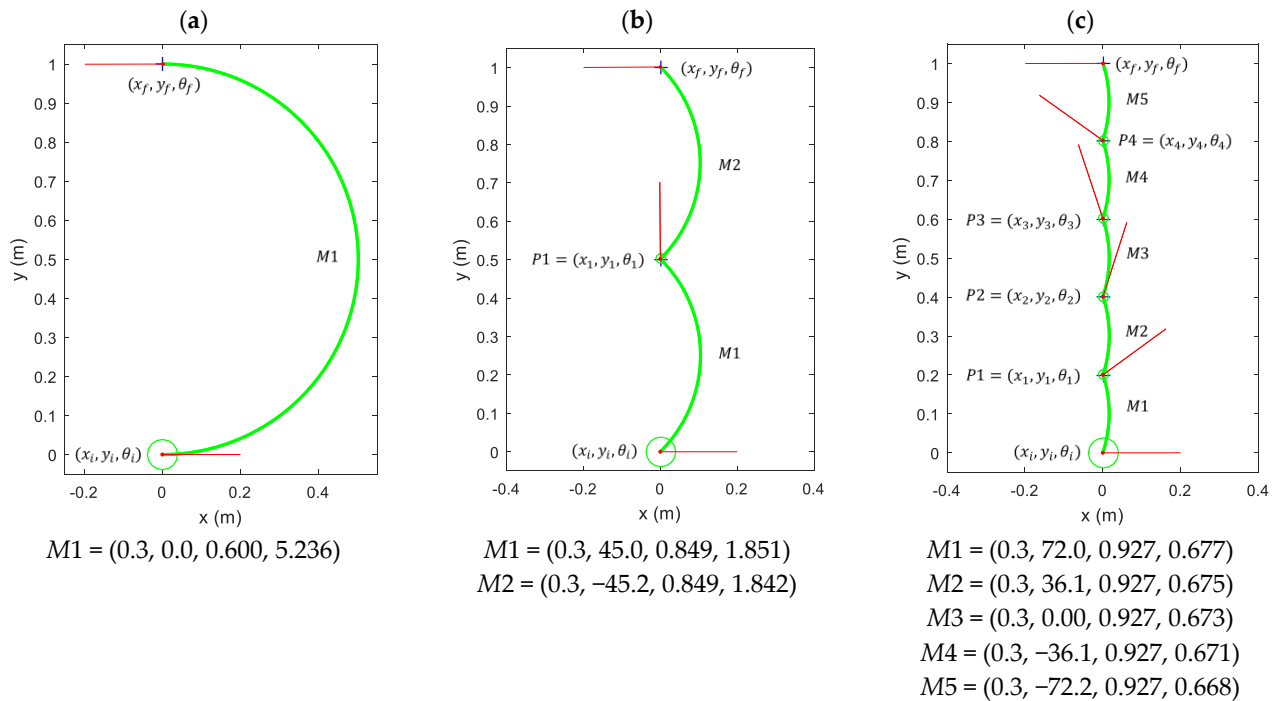
$$\begin{aligned} (x_i, y_i) &\rightarrow \dots \rightarrow (x_k, y_k) \rightarrow (x_{k+1}, y_{k+1}) \rightarrow \dots \rightarrow (x_f, y_f) \\ (\theta_i) &\rightarrow \dots \rightarrow (\theta_k) \rightarrow (\theta_{k+1}) \rightarrow \dots \rightarrow (\theta_f) \end{aligned} \quad (42)$$

Therefore, an omnidirectional mobile robot is able to keep the same orientation during the whole displacement  $(\theta_f = \theta_i)$ , reach a specific final angular orientation  $(\theta_f = 90^\circ)$ , rotate the mobile robot during the displacement  $(\theta_f = \theta_i + N \cdot 360^\circ)$  or maintain an orientation tangent to the planned trajectory  $(\theta_k = \tan((y_{k+1} - y_k) / (x_{k+1} - x_k)))$  in order to define a humanlike smooth motion that is expected to be more socially accepted [33].

This linearizing and smoothing strategy is based on the assumption that the motion command required to move the omnidirectional mobile robot between two positions  $(x_k, y_k, \theta_k) \rightarrow (x_{k+1}, y_{k+1}, \theta_{k+1})$  with a known target translational velocity  $v$  can be analytically obtained using the procedures described in Section 3.2.

Figure 6 illustrates the application of this path-planning procedure. Figure 6 shows an omnidirectional mobile robot that has to move from a starting point  $P_i = (x_i, y_i, \theta_i)$  located at  $(x_i = 0, y_i = 0, \theta_i = 0)$  to a final destination point located at  $(x_f = 0, y_f = 1, \theta_f = 180^\circ)$  with an expected transversal linear velocity  $v$  fixed to 0.3 m/s. Figure 6a shows the implementation of this displacement using only a single motion command  $M1$  (computed as defined in Section 3.2). In this example, the angular orientation of the mobile robot changes  $180^\circ$   $(\theta_i = 0^\circ, \theta_f = 180^\circ)$ , and thus this single motion  $M1$  will require a certain angular velocity  $(\omega \neq 0)$  to rotate the mobile robot that will generate a curved trajectory displacement (see Figure 6a). In this case, this single motion  $M1$  needs an execution time  $t_r$  of 5.236 s in order to reach the destination. Alternatively, Figure 6b shows the same displacement using one intermediate trajectory point  $P1$  interpolated in the middle of the displacement. In this case, the first motion command  $M1$  is required to reach the intermediate trajectory point, and then the second motion command  $M2$  is required to reach the destination. This trajectory is then composed of two consecutive circular displacements, but the inclusion of the intermediate point  $P1$  has reduced significantly the arc of the circular trajectory of the mobile robot. Finally, Figure 6c shows the effect of using four interpolated trajectory points  $P1, P2, P3$  and  $P4$  in a planned direct displacement from  $P_i$  to  $P_f$ . In this case, the arc of the trajectory is almost inexistent, and the execution of this displacement is visually perceived as a single straight displacement in which the mobile robot is rotating. Therefore, increasing the number of intermediate interpolated trajectory points between  $P_i$  and  $P_f$ , the trajectory of the mobile robot becomes continuous without trajectory discontinuities. In the APR-02, the minimum value of the separation distance between intermediate interpolated trajectory points was obtained by trial and error as 55

mm. This value depends largely on the setting time of the PID that controls the velocity of the wheels. This setting time is 0.5 s in the case of the APR-02. The example case shown in Figure 6 defines a displacement between  $P_i$  and  $P_f$ , but the intermediate interpolated points can define a direct trajectory or any arbitrary trajectory.



**Figure 6.** Representation of the motion command  $M = (v, \alpha, \omega, t_r)$  and the mobile robot trajectory (green line) required to move from a starting point  $P_i$  (green circle) to a planned destination  $P_f$  depending on the number of intermediate waypoints defined using an interpolation procedure: (a) direct trajectory with only one motion command and no intermediate waypoints; (b) trajectory with one intermediate waypoint  $P_1$  that requires the computation of two motion commands; (c) trajectory with four intermediate waypoints  $P_1, P_2, P_3$  and  $P_4$  that require the computation of five motion commands.

#### 4.3. Path-Following Procedure

The path-following procedure applied in the mobile robot APR-02 uses the current mobile robot position and orientation  $(x_p, y_p, \theta_p)$  updated by the SLAM procedure and close to the current target position  $(x_k, y_k, \theta_k)$  in order to compute the next motion command  $M = (v, \alpha, \omega, t_r)$  required to reach the next planned intermediate trajectory point  $(x_{k+1}, y_{k+1}, \theta_{k+1})$ . This computation is repeated until reaching the final destination point. Therefore, the goal of this procedure is always to have an updated motion command in order to go to the next planned intermediate trajectory point instead of trying to pass precisely over an intermediate trajectory point.

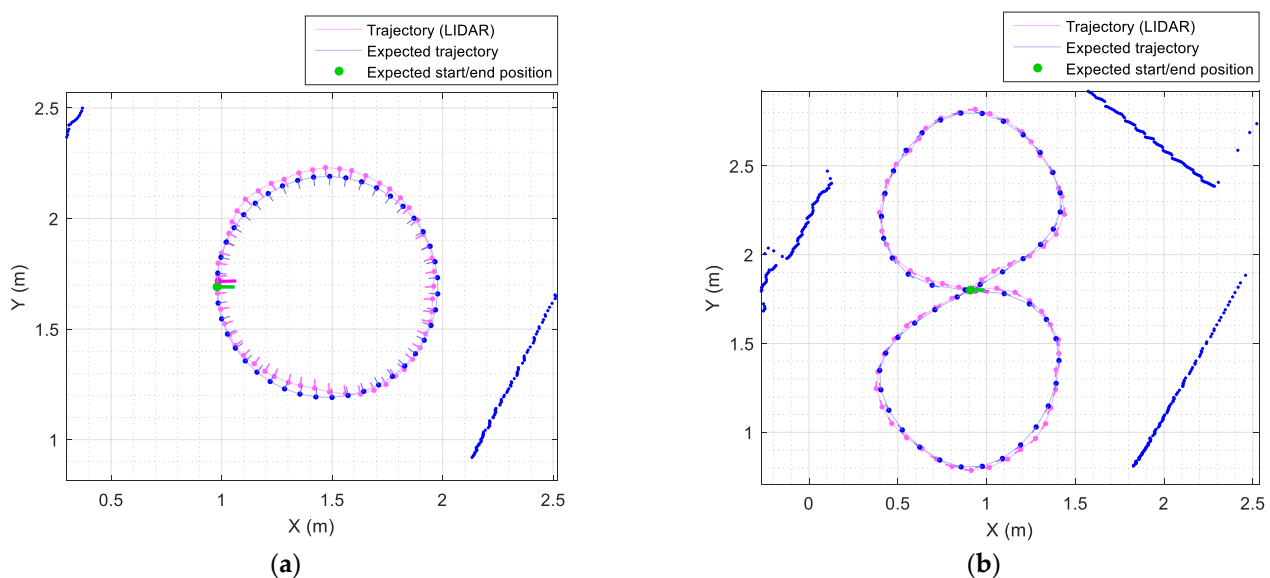
The implementation of this path-following procedure in the mobile robot APR-02 is based on the precise position feedback provided by the onboard LIDAR rather than on the use of the prediction horizon proposed in model predictive control (MPC) approaches (see [34] for a comprehensive review). The use of the information of the odometry as position feedback in this path-following procedure was also tested but it had to be discarded because of the well-known effect of the cumulated errors generated in the odometry during a large displacement.

Finally, the main advantage of the proposed path-following procedure is the automatic compensation of the motion errors caused by wheel slippage without requiring additional specific compensation procedures [13]. Another advantage of this path-following procedure is the ability to maintain a constant translational velocity  $v$  during the whole displacement, a feature that is expected to increase the social acceptance of a mobile robot operating in a shared space with humans [33].

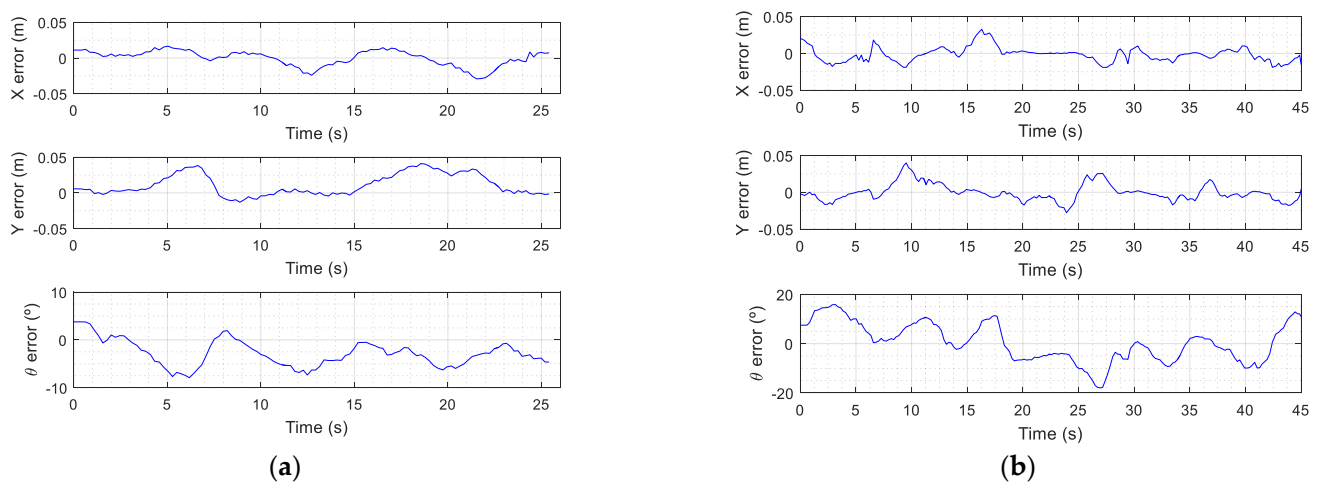
## 5. Experimental Evaluation of the Path-Tracking Accuracy

This section empirically evaluates the path-tracking accuracy of the path-planning and path-following procedures implemented in the mobile robot APR-02. As cited previously, this paper is inspired by the work of Li et al. [13] that proposed the evaluation of the path-tracking performances of an omnidirectional mobile robot using multiple Mecanum wheels completing circular and eight-shaped target trajectories. This evaluation was a pending task in the case of the mobile robot APR-02. The specific trajectories selected by Li et al. [13] are especially interesting. In the case of performing a circular trajectory, an omnidirectional mobile robot is able to complete this path by using only one motion command  $M = (v, \alpha, \omega)$  (see Figure 4), and thus this trajectory represents an easy path to follow. Alternatively, in the case of performing an eight-shaped trajectory, an omnidirectional mobile robot must continuously update the motion command  $M = (v, \alpha, \omega, t_r)$ , and thus this is a very challenging trajectory prone to control errors.

Figure 7 shows the circular and eight-shaped target trajectories used to evaluate the path-tracking accuracy of the omnidirectional mobile robot APR-02. Figure 7a shows a circular target trajectory (blue color) with a radius of 1 m. Figure 7a includes a sparse representation of the intermediate trajectory points  $(x_k, y_k, \theta_k)$  (blue point and blue line) used to plan this trajectory. Figure 7a also shows the ground truth trajectory (magenta color) obtained when the mobile robot APR-02 completes this trajectory with a linear translational velocity  $v$  of 0.15 m/s, which is a very low velocity for a human-sized mobile robot. This ground truth trajectory was obtained from the information registered by the onboard LIDAR. Figure 7a also depicts a sparse representation of the real mobile robot position and orientation (magenta point and magenta line) that are correlated with the intermediate trajectory points planned. The detailed evolution of the location and orientation errors  $(x_e, y_e, \theta_e)$  obtained in this experiment are also plotted in Figure 8a. Similarly, Figure 7b shows an eight-shaped target trajectory (blue color) with a distance between focus of 1 m. Figure 7b includes a sparse representation of the intermediate trajectory points  $(x_k, y_k, \theta_k)$  (blue point and blue line) used to plan this trajectory. Figure 7b also shows the ground truth trajectory (magenta color) obtained when the mobile robot APR-02 completes this trajectory with a linear translational velocity  $v$  of 0.15 m/s. The detailed evolution of the location and orientation errors  $(x_e, y_e, \theta_e)$  obtained in this experiment are plotted in Figure 8b.



**Figure 7.** Comparison between the planned (blue line) and real (magenta) trajectories of the mobile robot moving at a constant speed of 0.15 m/s. The green line depicts the expected initial and final positions and orientations of the mobile robot: (a) describing a circular trajectory with the robot facing inward and (b) describing an eight-shaped trajectory with the robot facing forward.



**Figure 8.** Error location ( $x_e, y_e, \theta_e$ ) when the mobile robot moves at a constant translational velocity of 0.15 m/s: (a) following a circular trajectory with the robot facing inward and (b) following an eight-shaped trajectory with the robot facing forward.

Finally, Tables 1 and 2 summarize the root mean square (RMS) error (RMSE) and maximum absolute error of the Euclidean distance between the expected and real mobile robot locations and between the expected and real mobile robot angular orientations. Table 1 shows the results obtained when performing a circular target trajectory and Table 2 the results obtained when performing an eight-shaped target trajectory. These two tables summarize the errors obtained with different translational velocities  $v$  where 0.10 m/s is visually perceived as a very slow velocity, 0.30 m/s is the nominal velocity that is visually perceived as normal or adequate for the mobile robot APR-02, and 0.5 m/s is externally visually perceived as a very fast (and maybe annoying) velocity. A video prepared by the authors showing the mobile robot APR-02 completing these two target trajectories at a translational velocity of 0.3 m/s is available on YouTube [35].

**Table 1.** Path-tracking errors obtained in the case of performing a circular target trajectory (Figure 7a).

Speed (m/s)	Distance (m)		Angular Orientation (°)	
	RMSE	Absolute Maximum Error	RMSE	Absolute Maximum Error
0.10	0.017203	0.042762	6.7002	13.4761
0.15	0.021478	0.043392	3.9457	7.9026
0.20	0.023732	0.050270	5.6000	12.8638
0.25	0.033889	0.060157	5.3974	10.8083
0.30	0.032467	0.077929	6.2730	12.6040
0.35	0.051998	0.125830	7.3219	15.2344
0.40	0.038150	0.080402	6.3755	12.9896
0.45	0.040882	0.101140	6.0992	11.1705
0.50	0.032762	0.070848	8.0095	17.6987

**Table 2.** Path-tracking errors obtained in the case of performing an eight-shaped target trajectory (Figure 7b).

Speed (m/s)	Distance (m)		Angular Orientation (°)	
	RMSE	Absolute Maximum Error	RMSE	Absolute Maximum Error
0.10	0.017036	0.045863	8.99090	20.8563
0.15	0.015341	0.044073	7.80650	17.9776

Table 2. Cont.

Speed (m/s)	Distance (m)		Angular Orientation (°)	
	RMSE	Absolute Maximum Error	RMSE	Absolute Maximum Error
0.20	0.017418	0.039401	7.08460	17.6608
0.25	0.025817	0.068124	6.53130	17.2021
0.30	0.039706	0.088557	7.76150	21.5102
0.35	0.059036	0.123140	7.64930	19.2291
0.40	0.065989	0.151920	9.48780	20.7115
0.45	0.087276	0.220810	11.5129	24.3387
0.50	0.100260	0.265240	12.2232	22.5929

## 6. Discussion and Conclusions

This paper presents the empirical evaluation of the path-tracking accuracy of the omnidirectional mobile robot APR-02 designed to develop services as a personal assistant. This mobile robot uses three omnidirectional wheels driven by geared brushed DC motors with magnetic rotary encoders attached and is able to move in any direction while simultaneously changing its orientation. This paper describes the kinematics and path-planning procedure implemented in the mobile robot and empirically evaluates its path-tracking accuracy. The mobile robot uses a path-following procedure based on the self-location capabilities provided by a precise onboard LIDAR. The ground truth trajectory of the mobile robot during the experiments has been obtained by registering the information gathered by the onboard LIDAR. The experimentation area used to conduct the experiments included large plain walls in order to maximize the precision of the self-locating procedure [25].

Results of Table 1 show that the RMS error of the distance measured when performing a circular trajectory has its minimum value (0.017 m) for a translational velocity of 0.1 m/s, which is a very low velocity for a humanlike mobile robot. The RMS distance error has a slightly increasing tendency for translational velocities in the range from 0.25 to 0.5 m/s: between 0.033 m and 0.051 m. In this translational velocity range, the maximum distance error is in a range from 0.06 m to 0.12 m. This distance error can be interpreted as low for a mobile robot with a base diameter of 0.54 m and 31 kg. Results of Table 1 also show that the RMS of the angular orientation error of the mobile robot is always close to 6.5° and the absolute value of the maximum error is close to 13° in all the velocity ranges analyzed. We observed that this angular difference is due to the fact that the mobile robot always aligns its orientation with its trajectory, and thus a correction in the trajectory suddenly increases the angular error of the mobile robot. This current implementation produces smooth visual trajectories, but this effect will be analyzed in depth in future works in order to effectively reduce the angular error of the mobile robot. A video sequence showing the mobile robot APR-02 completing a circular trajectory while rotating is available in [35].

Results of Table 2 show that the RMS error in the distance measured when performing an eight-shaped trajectory monotonically increases as the translational velocity increases. This trajectory is very challenging for any type of mobile robot because the speed of the wheels must be continuously adapted during this motion. The RMS distance error increased one order of magnitude in the velocity range from 0.1 to 0.5 m/s with values from 0.017 to 0.100 m. At low velocities, the absolute maximum error is around 0.44 m, at the nominal velocity of 0.3 m/s, the maximum error is 0.088 m, and the difficulty of this trajectory is evidenced at the velocity of 0.50 m/s with an instantaneous maximum distance error of 0.265 m. Again, the trajectory of the mobile robot is perceived as smooth, and the trajectory corrections are not visually detected. Results of Table 2 also show that the RMS of the angular orientation error of the mobile robot is around 9° or 10° with absolute maximum



angular errors around  $20^\circ$  in all velocity ranges because this angular error is caused when correcting the trajectory of the mobile robot. A video sequence showing the mobile robot APR-02 completing an eight-shaped trajectory is available at [35].

The comparison of the path-tracking errors with the results of the scientific literature was complicated because of the different metrics used, the different sizes and weights of the mobile robots and the different motorization used. The comparative proposal of Li et al. [13] evaluated the path-following performances of omnidirectional mobile robots using four Mecanum wheels, evaluating the relative error of the displacement as a metric but without specifying the error in the angular orientation of the mobile robot. In [13], a circular trajectory with a radius of 1 m at a velocity of 0.2 m/s produced a relative error of the displacement of  $-15.23\%$  in the X-axis and  $-2.85\%$  in the Y-axis; this error was reduced to  $-1.99\%$  in the X-axis and  $-1.50\%$  in the Y-axis when applying a velocity compensation coefficient, a strategy that was not applied in this work. In [13], an eight-shaped trajectory at a velocity of 0.2 m/s produced relative errors of  $-13.91\%$  and  $-1.17\%$  and  $-1.95\%$  and  $-0.04\%$  when applying a tuned velocity compensation coefficient.

In general, the results obtained in this paper are in the same range as the results obtained by Li et al. [13] using a translational velocity of 0.2 m/s: absolute distance error of 0.050 m in the circular trajectory and 0.039 m in the eight-shaped trajectory. Then, the new contribution of this paper is the evaluation of the path-tracking accuracy at different translational velocities and the analysis of the error in the angular orientation of the mobile robot, which is fundamental information for an omnidirectional mobile robot. Another improvement relative to the proposal of Li et al. [13] is the avoidance of remote-control procedures applied to control the path of the mobile robot analyzed. However, a future pending work is the validation of the effect of tuned velocity compensation coefficients in path-tracking accuracy.

In this paper, the evaluation results of the path-tracking accuracy of the three-wheeled omnidirectional mobile robot APR-02 obtained when performing a circular trajectory at the nominal translational velocity of 0.3 m/s are: RMS distance error of 0.032 m, absolute distance error of 0.077 m, RMS angular orientation error of  $6.27^\circ$  and absolute angular orientation error of  $12.60^\circ$ . In the case of a more challenging eight-shaped trajectory, these values are: RMS distance error of 0.039 m, absolute distance error of 0.088 m, RMS angular orientation error of  $7.76^\circ$  and absolute angular orientation error of  $21.51^\circ$ . These small trajectory errors summarize the good visual impression generated by the displacement of the mobile robot APR-02.

Future research will be focused on the evaluation of the performance of the path-planning procedure of the mobile robot passing through open doors and maneuvering in a crowded environment. One of the goals will be the analysis of the affinity generated by the displacement of the mobile robot.

**Author Contributions:** Conceptualization and writing—review and editing, J.P. and E.R.; investigation and writing, E.R.; resources, E.C. and D.M.; methodology, J.P. and E.R. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Duda, S.; Dudek, O.; Gembalczyk, G. Developing a Test Site for Testing the Suspension of Vehicles with Omnidirectional Wheels. *Vib. Phys. Syst.* **2020**, *31*, 2020304.
2. Hou, L.; Zhou, F.; Kim, K.; Zhang, L. Practical Model for Energy Consumption Analysis of Omnidirectional Mobile Robot. *Sensors* **2021**, *21*, 1800. [[CrossRef](#)] [[PubMed](#)]
3. Kao, S.-T.; Ho, M.-T. Ball-Catching System Using Image Processing and an Omni-Directional Wheeled Mobile Robot. *Sensors* **2021**, *21*, 3208. [[CrossRef](#)] [[PubMed](#)]
4. Levratti, A.; De Vuono, A.; Fantuzzi, C.; Secchi, C. TIREBOT: A novel Tire Workshop Assistant Robot. In Proceedings of the International Conference on Advanced Intelligent Mechatronics (AIM), Banff, AB, Canada, 12–15 July 2016. [[CrossRef](#)]
5. Bogue, R. Domestic robots: Has their time finally come? *Ind. Robot Int. J.* **2017**, *44*, 129–136. [[CrossRef](#)]

6. Tagliavini, L.; Botta, A.; Cavallone, P.; Carbonari, L.; Quaglia, G. On the Suspension Design of Paquitop, a Novel Service Robot for Home Assistance Applications. *Machines* **2021**, *9*, 52. [[CrossRef](#)]
7. Saadatzi, M.N.; Abubakar, S.; Das, S.K.; Saadatzi, M.H.; Popa, D. Neuroadaptive Controller for Physical Interaction with an Omni-Directional Mobile Nurse Assistant Robot. In Proceedings of the ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Virtual, Online, 17–19 August 2020. [[CrossRef](#)]
8. Palacín, J.; Martínez, D. Improving the Angular Velocity Measured with a Low-Cost Magnetic Rotary Encoder Attached to a Brushed DC Motor by Compensating Magnet and Hall-Effect Sensor Misalignments. *Sensors* **2021**, *21*, 4763. [[CrossRef](#)] [[PubMed](#)]
9. Tsai, C.-C.; Jiang, L.-B.; Wang, T.-Y.; Wang, T.-S. Kinematics Control of an Omnidirectional Mobile Robot. In Proceedings of the CACS Automatic Control Conference, Tainan, China, 18–19 November 2005.
10. Tri, R.; Arifinatio, D.; Bachtiar, F.; Intan, J. Holonomic Implementation of Three Wheels Omnidirectional Mobile Robot using DC Motors. *J. Robot. Control JRC* **2021**, *2*, 65–71.
11. Lin, P.; Liu, D.; Yang, D.; Zou, Q.; Du, Y.; Cong, M. Calibration for Odometry of Omnidirectional Mobile Robots Based on Kinematic Correction. In Proceedings of the International Conference on Computer Science & Education (ICCSE), Toronto, ON, Canada, 19–21 August 2019. [[CrossRef](#)]
12. Maddahi, Y.; Maddahi, A.; Sepehri, N. Callibration of omnidirectional wheeled mobile robots: Method and experiments. *Robotica* **2013**, *31*, 969–980. [[CrossRef](#)]
13. Li, Y.; Ge, S.; Dai, S.; Zhao, L.; Yan, X.; Zheng, Y.; Shi, Y. Kinematic Modeling of a Combined System of Multiple Mecanum-Wheeled Robots with Velocity Compensation. *Sensors* **2020**, *20*, 75. [[CrossRef](#)] [[PubMed](#)]
14. Baede, T.A. Motion Control of An Omnidirectional Mobile Robot. In Proceedings of the Motion Control of an Omnidirectional Mobile Robot, Eindhoven, The Netherlands, 18 September 2006.
15. Wang, C.; Liu, X.; Yang, X.; Hu, F.; Jiang, A.; Yang, C. Trajectory Tracking of an Omni-Directional Wheeled Mobile Robot Using a Model Predictive Control Strategy. *Appl. Sci.* **2018**, *8*, 231. [[CrossRef](#)]
16. Leng, C.; Cao, Q.; Huang, Y. A Motion Planning Method for Omnidirectional Mobile Robot Based on the Anisotropic Characteristics. *Int. J. Adv. Robot. Syst.* **2008**, *5*, 45. [[CrossRef](#)]
17. Clotet, E.; Martínez, D.; Moreno, J.; Tresanchez, M.; Palacín, J. Assistant Personal Robot (APR): Conception and Application of a Tele-Operated Assisted Living Robot. *Sensors* **2016**, *16*, 610. [[CrossRef](#)]
18. Moreno, J.; Clotet, E.; Lupiañez, R.; Tresanchez, M.; Martínez, D.; Pallejà, T.; Casanovas, J.; Palacín, J. Design, Implementation and Validation of the Three-Wheel Holonomic Motion System of the Assistant Personal Robot (APR). *Sensors* **2016**, *16*, 1658. [[CrossRef](#)]
19. Palacín, J.; Martínez, D.; Clotet, E.; Pallejà, T.; Burgués, J.; Fonollosa, J.; Pardo, A.; Marco, S. Application of an Array of Metal-Oxide Semiconductor Gas Sensors in an Assistant Personal Robot for Early Gas Leak Detection. *Sensors* **2019**, *19*, 1957. [[CrossRef](#)]
20. Slovák, J.; Melicher, M.; Šimovec, M.; Vachálek, J. Vision and RTLS Safety Implementation in an Experimental Human—Robot Collaboration Scenario. *Sensors* **2021**, *21*, 2419. [[CrossRef](#)]
21. Bonci, A.; Cen Cheng, P.D.; Indri, M.; Nabissi, G.; Sibona, F. Human-Robot Perception in Industrial Environments: A Survey. *Sensors* **2021**, *21*, 1571. [[CrossRef](#)]
22. Palacín, J.; Clotet, E.; Martínez, D.; Martínez, D.; Moreno, J. Extending the Application of an Assistant Personal Robot as a Walk-Helper Tool. *Robotics* **2019**, *8*, 27. [[CrossRef](#)]
23. Lluvia, I.; Lazkano, E.; Ansuategi, A. Active Mapping and Robot Exploration: A Survey. *Sensors* **2021**, *21*, 2445. [[CrossRef](#)]
24. Palacín, J.; Martínez, D.; Rubies, E.; Clotet, E. Suboptimal Omnidirectional Wheel Design and Implementation. *Sensors* **2021**, *21*, 865. [[CrossRef](#)]
25. Palacín, J.; Martínez, D.; Rubies, E.; Clotet, E. Mobile Robot Self-Localization with 2D Push-Broom LIDAR in a 2D Map. *Sensors* **2020**, *20*, 2500. [[CrossRef](#)]
26. Inthiam, J.; Deelertpaiboon, C. Self-Localization and Navigation of Holonomic Mobile Robot Using Omni-Directional Wheel Odometry. In Proceedings of the TENCON 2014—2014 IEEE Region 10 Conference, Bangkok, Thailand, 22–25 October 2014; pp. 1–5. [[CrossRef](#)]
27. Rijalusalam, D.U.; Iswanto, I. Implementation Kinematics Modeling and Odometry of Four Omni Wheel Mobile Robot on The Trajectory Planning and Motion Control Based Microcontroller. *JRC* **2021**, *2*, 448–455. [[CrossRef](#)]
28. Li, Y.; Dai, S.; Shi, Y.; Zhao, L.; Ding, M. Navigation Simulation of a Mecanum Wheel Mobile Robot Based on an Improved A\* Algorithm in Unity3D. *Sensors* **2019**, *19*, 2976. [[CrossRef](#)]
29. Lau, B.; Sprunk, C.; Burgard, W. Kinodynamic Motion Planning for Mobile Robots Using Splines. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, St. Louis, MO, USA, 10–15 October 2009. [[CrossRef](#)]
30. Sprunk, C.; Lau, B.; Pfaffz, P.; Burgard, W. Online Generation of Kinodynamic Trajectories for Non-Circular Omnidirectional Robots. In Proceedings of the IEEE International Conference on Robotics and Automation, Shanghai, China, 9–13 May 2011. [[CrossRef](#)]
31. Kuenemund, F.; Kirsch, C.; Hess, D.; Roehrig, C. Fast and Accurate Trajectory Generation for Non-Circular Omnidirectional Robots in Industrial Applications. In Proceedings of the ROBOTIK: German Conference on Robotics, Munich, Germany, 21–22 May 2012.
32. Cao, Z.; Bryant, D.; Molteno, T.C.A.; Fox, C.; Parry, M. V-Spline: An Adaptive Smoothing Spline for Trajectory Reconstruction. *Sensors* **2021**, *21*, 3215. [[CrossRef](#)] [[PubMed](#)]

33. Guillén Ruiz, S.; Calderita, L.V.; Hidalgo-Paniagua, A.; Bandera Rubio, J.P. Measuring Smoothness as a Factor for Efficient and Socially Accepted Robot Motion. *Sensors* **2020**, *20*, 6822. [[CrossRef](#)]
34. Nascimento, T.; Dórea, C.; Gonçalves, L. Nonholonomic mobile robots' trajectory tracking model predictive control: A survey. *Robotica* **2018**, *36*, 676–696. [[CrossRef](#)]
35. Palacin, J.; Clotet, E. *RoboticaUdL*. APR-02 Eight-Shaped and Circular Trajectories. Youtube. 2021. Available online: [https://youtu.be/vRLM-kc2\\_UM](https://youtu.be/vRLM-kc2_UM) (accessed on 12 September 2021).