Research Article/Mechanical, Mechatronics, and Materials Engineering http://doi.org/10.15446/ing.investig.111403

Path Tracking for a Differential-Drive Robot Applied in the Automatic Disinfection of Hospital Environments

Seguimiento de ruta para un robot de guiado diferencial aplicado en la desinfección automática de entornos hospitalarios

Diana L. Gómez [1], Juan S. Cardozo [2], David A. Ramírez-Gámez [3], and Mauricio F. Jaramillo-Morales [4]

ABSTRACT

Since the end of the COVID-19 emergency was declared, the development and implementation of robotic systems aimed at mitigating exposure to the virus or to similar diseases has increased. This work proposed a simulation algorithm aimed at automatic route tracking for a differential-drive mobile robot in the disinfection of hospital environments, using scanners as proximity sensors. Furthermore, to validate the route-tracking algorithm, scenarios were created in the robot operating system (ROS), which were based on the floor plans of two healthcare institutions in the Tolima region (Colombia), i.e., the Emergency Room and the Coronary Intensive Care Unit of the Tolima Clinic and the Neonatal Intensive Care Unit of the Maternal and Child Health Unit of Tolima. For validation, route tracking was first performed automatically using the proposed algorithm. Then, the computer keyboard was used to perform manual tracking, simulating a joystick within a real robotic implementation. Five tests were conducted for each scenario, with better results obtained through automatic tracking, demonstrating the efficiency of this methodology.

Keywords: hospital disinfection, mobile robotics, path tracking, robot positioning with scanners, ROS

RESUMEN

Desde que se declaró el fin de la emergencia por COVID-19, se ha incrementado el desarrollo y la implementación de sistemas robóticos enfocados a mitigar la exposición al virus o a enfermedades similares. En este trabajo se propuso un algoritmo de simulación orientado al seguimiento automático de rutas para un robot móvil de guiado diferencial en la desinfección de ambientes hospitalarios, utilizando el escáner como sensor de proximidad. Además, para validar el algoritmo de seguimiento de rutas, se crearon escenarios en el sistema operativo robótico (ROS), con base en los planos de dos instituciones de salud de la región del Tolima (Colombia), i.e., Urgencias y la Unidad Coronaria de Cuidados Intensivos de la Clínica Tolima y la Unidad de Cuidados Intensivos Neonatal de la Unidad Materno Infantil del Tolima. Para la validación, el seguimiento de las rutas se realizó primero de forma automática con el algoritmo propuesto. Luego se empleó el teclado del computador para realizar un seguimiento manual, simulando un joystick en una implementación robótica real. Se realizaron cinco pruebas para cada escenario, obteniendo mejores resultados mediante el seguimiento automático, lo que comprueba la eficiencia de esta metodología.

Palabras clave: desinfección hospitalaria, robótica móvil, seguimiento de ruta, posicionamiento robótico con escáner, ROS

Received: August 29th, 2023
Accepted: August 29th, 2025

Highlights

- This work presents a methodology for disinfection path following by a robot in hospital environments
- Scanners were used to follow the defined paths
- 3D maps of two healthcare facilities in Tolima (Colombia) were elaborated in the ROS simulator called *Stage*

Introduction

Robotics has become a solution to many challenges across fields such as industry, medicine, services, and entertainment. In healthcare, its growth is especially evident, with innovations like robotic rehabilitation devices [1], surgical robots such as the Da Vinci system [2], hospital disinfection robots [3], medicine delivery systems [4], and telemedicine support robots [5].

On January 30th, 2020, the World Health Organization (WHO) declared the SARS-CoV-2 (COVID-19) outbreak a global emergency [6]. This led to an urgent need for effective cleaning and disinfection techniques [7]. According to the WHO and the Center for Disease Control and Prevention (CDC), frequent cleaning and disinfection are among the

[1] Biomedical engineer, Universidad Autónoma de Manizales, Caldas, Colombia. Member of the Research Group in Automatics, Universidad Autónoma de Manizales. Email: diana.gomezl@autonoma.edu.co

[2] Biomedical engineer, Universidad Autónoma de Manizales, Caldas, Colombia. Member of the Research Group in Automatics, Universidad Autónoma de Manizales. Email: juan.cardozor@autonoma.edu.co

[3] Biomedical Ing, Universidad Autónoma de Manizales (UAM), Colombia. Member of the automatic research group, Universidad Autónoma de Manizales, Caldas, Colombia. Email: davida.ramirezg@autonoma.edu.co

[4] Electronics engineer, Universidad Nacional de Colombia. Researcher and full professor, Department of Electronics Engineering and Automation, Universidad Autónoma de Manizales. Email: mfjaramillo@autonoma.edu.co

most effective measures for preventing viral transmission, particularly in hospitals. As a result, robotic disinfection systems were rapidly developed in the face of this pandemic [8], [9], especially in areas with high viral loads.

Developing such robots involves challenges like improving localization precision through environmental characterization, obstacle detection, and path planning in order to ensure minimal error during navigation. This work proposes an automatic path-tracking method for a differential-drive mobile robot using scanner-based localization. The goal is to disinfect high-risk areas, minimizing exposure for healthcare and cleaning staff.

Simulations were conducted in Stage, a tool within the Robot Operating System (ROS), using maps of two health facilities in Ibagué, Tolima. The environment was static, without dynamic obstacles or slopes. The ROS is an open-source framework that is widely used in both academia and industry [10]. For instance, [11] used ROS to control a Toyota robot for disinfecting door handles, albeit failing to provide simulation details or implementation algorithms. [12] simulated a mobile robot in Gazebo with a disinfection arm but did not employ the path-tracking method. Similarly, [13] controlled multi-robot systems in Stage using predefined maps, while [14] developed navigation algorithms in Stage but used simplified environments lacking real-world obstacle modeling. [15] compared path-planning methods in MATLAB, which lacks sensor integration (*e.g.*, laser scanners). Lastly, [16] proposed a dual disinfection system (UV and aerosol) with a solid architecture but without validation in realistic simulation environments.

In this study, three hospital maps were recreated in Stage: the Emergency Room and Coronary ICU of the Tolima Clinic, and the Neonatal ICU of the Maternal and Child Health Unit of Tolima. A differential-drive robot was assigned a disinfection route for each environment. The robot followed the route automatically, using on-off and proportional control strategies based on scanner feedback for linear and rotational motion. An initial analysis of orientation behavior and scanner range data was conducted to adjust the tracking code in Stage. The robot was expected to reach each disinfection point while avoiding collisions with walls and objects. To validate the method, tests were performed across the three environments. The error between the actual and target positions was measured and compared against manual route tracking. The automatic method showed improved accuracy.

Direct kinematic model

The differential drive configuration for mobile robots is one of the most common systems for navigation in confined working environments (*e.g.*, hospitals). The movements performed by the robot to achieve path following are linear movement and rotational movement on its own axis in the XY plane [17]. Fig. 1 shows the diagram used in this work for direct kinematic modeling.

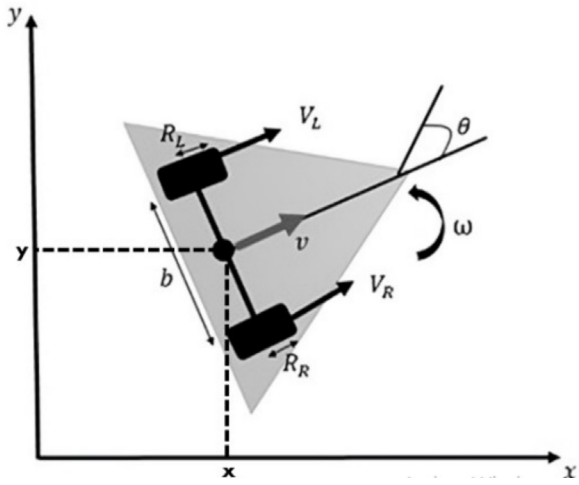

Figure 1. Kinematic model diagram
Source: Authors

Table 1 presents the variables corresponding to the kinematic model diagram shown in Fig. 1.

Table I. Variables of the differential-drive mobile robot

Variable	Description
VL, VR	Angular velocity of the left and right wheels of the robot
x, y	Coordinates of the center of the robot in a fixed reference coordinated frame X–Y
RL, RR	Radius of the driving wheels (Left and Right)
V	Linear velocity of the robot
ω	Angular velocity of the robot
θ	Angular position of the robot
B	Distance between the wheels

Source: Authors

Here, *linear velocity* is defined as the average of the left and right wheel velocities and is proportional to the wheel radius *R*. If the radii of the left and right driving wheels are equal, then $R = RR = RL$, as expressed in Eqs. (1) and (2).

$$v = R * \frac{V_R + V_L}{2} \tag{1}$$

Angular velocity is defined as the difference between the wheels' velocity over the distance between them. The angular velocity is proportional to *R*.

$$\omega = R * \frac{V_R - V_L}{b} \tag{2}$$

Linear and rotational movements

It is necessary for the robot to calculate linear forward movements and rotations about its own axis, so that it can follow the required disinfection paths.

Forward linear movement

Regarding forward movement, the kinematic condition requires that the angular velocities of the wheels be positive and equal, as shown in Eqs. (3) to (6). Therefore $V_T = V_R = V_L$.

$$v = R * \frac{V_R + V_L}{2} \quad \text{if} \quad V_L = V_R \tag{3}$$

$$v = R * \frac{V_R + V_R}{2} \tag{4}$$

$$v = R * V_R \tag{5}$$

$$v = R * V_T \tag{6}$$

Rotational movement about the robot's axis

As for the robot's rotations, the wheels' angular velocity must be equal and of opposite sign, so that $-V_L = V_R$. Eqs. (7) to (12) next show the corresponding substitutions.

$$\boldsymbol{w = R * \frac{V_R - V_L}{b}} \quad \text{if} \quad \boldsymbol{V_L = -V_R} \tag{7}$$

$$w = R * \frac{V_R - (-V_R)}{b} \tag{8}$$

$$w = \frac{2 * R}{b} * V_R \quad \text{if} \quad \boldsymbol{V_R = -V_L} \tag{9}$$

$$\boldsymbol{w = R * \frac{-V_L - V_L}{b}} \tag{10}$$

$$w = R * \frac{-2 * V_L}{b} \tag{11}$$

$$w = -\frac{2 * R}{b} * V_L \tag{12}$$

The above implies that the relation between the robot's angular velocity and the wheel angular velocity is equal to twice the quotient between the wheel radius and the distance between the wheels. The positive sign in the equation implies a counterclockwise rotational movement—clockwise rotation is symbolized by a negative sign.

Stage simulator (ROS)

Stage is widely used to develop and validate robot navigation systems via virtual tools, which include obstacles of different heights, sensors, and algorithms, among others [18]. One of the main advantages of this simulator is its ability to create *do-it-yourself* (DIY) worlds, although there are predefined worlds like Willow (Fig. 2).

The Stage scanner provides proximity data on the environment in exploration applications. The generated algorithms can be easily adapted for a real robot if its microprocessor incorporates ROS, as some commercial robots do [19], [20].

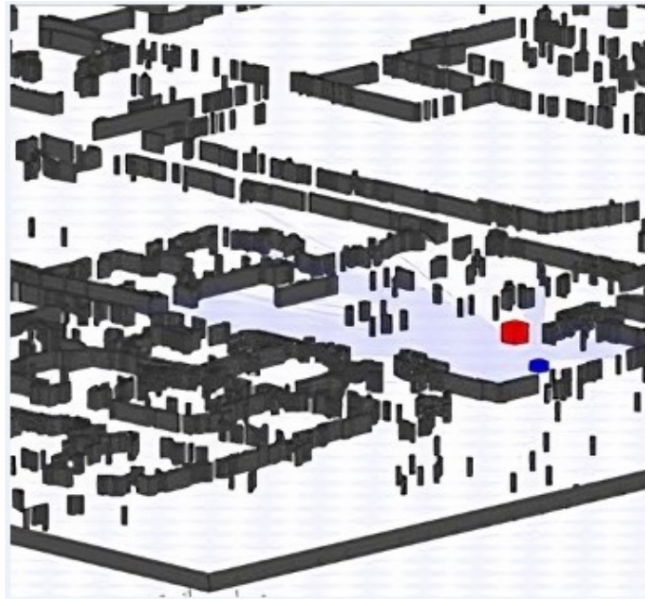

Figure 2. 3D view of the Willow world
Source: Stage - ROS Wiki [21]

Scanner description in ROS

A scanner is an electronic device commonly used to obtain distance measurements of objects in indoor environments. These sensors can be easily integrated into a mobile robot platform [22], [23].

The scanner used is composed of an arrangement of lasers located in different angular positions, receiving proximity information on the elements in the scene. To simplify the sensor data, a sector of the scanner is selected depending on the robot's movement. The data obtained are used to guide the robot in exploring an unknown environment, avoiding obstacles, or automatically following a path.

In ROS, we chose a visual range of 270° (Fig. 3), as well as a maximum distance of 20 m. These parameters were selected because the LMS100 LIDAR scanner, which is often used in real robotics applications, has similar values [24].

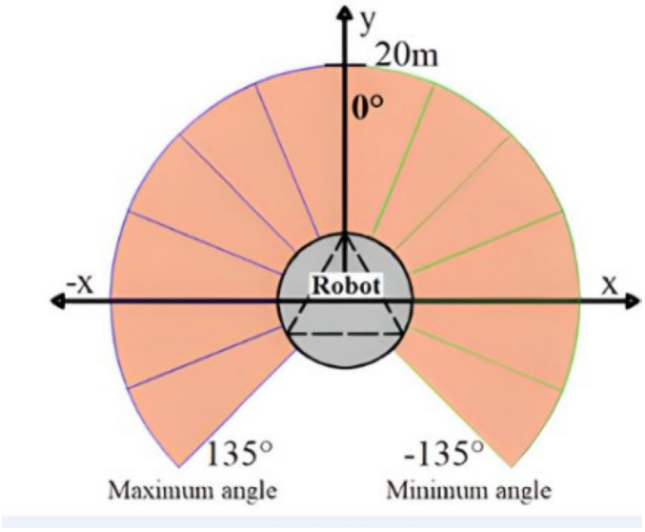

Figure 3. Scanner distribution diagram
Source: Authors

Disinfection device

As for the disinfection methods suitable for real-world implementations, several related works have used ultraviolet light as an alternative to traditional methods with chemical disinfectants [25].

Currently, disinfection devices use type-C ultraviolet radiation (UVC), a short-wave light with wavelengths between 200 and 270 nm [26]. Unlike conventional ultraviolet light, UVC does not pose a risk to human health [27], making it suitable for use in areas frequented by people. UVC can destroy molecular bonds and break the DNA or RNA of microorganisms such as bacteria and viruses. Recent studies have shown that SARS-CoV-2 and other similar viruses are susceptible to this type of radiation when exposed to doses of over 1000 J/cm^2 [28] for a recommended duration of 15 to 93 minutes and from a distance of approximately 2 m [29]. This neutralizes their ability to infect a host.

Methodology

Scenarios used for validation

Stage allows simulating an environment defined by a world file, considering scanner parameters such as range and coverage. Obstacles can be added, and environment settings such as XY locations, directions, and constraints can be modified. Although Stage includes some default navigation maps that are available in the literature, this project focused on floor plans provided by two healthcare facilities in the department of Tolima: the Emergency Room and Coronary ICU of the Tolima Clinic, and the Neonatal ICU (NICU) of the Maternal and Child Health Unit of Tolima (UMIT).

In this case, the virtual hospital environment included real-world obstacles such as chairs, stretchers, sinks, and walls. Therefore, the floor plans were first created in AutoCAD to generate realistic layouts. Since Stage requires pgm image files in grayscale, a PDF was exported from AutoCAD and then converted into the correct format. The resulting image was saved in a folder alongside the world file, and it was loaded as shown in Code 1.

Code 1: load an environment bitmap

```
1:  floorplan
2:  (
3:    name “willow”
4:    bitmap “Paredes.pgm”
5:    size [45.15  56  17.5]
6:    pose [10.000 10.000 0.000 90.000]
7:    color “gray100”
8:  )
```

Source: Authors

However, when using a single image containing the complete map, there is no height differentiation between objects, as they all share the same vertical dimension—set in line 5 of Code 1, in the last position of the size variable. Height is a crucial factor, since the robot uses object detection for spatial orientation, and some elements may lie outside the scanner's detection range and should not return data. To address this issue, the full map image was divided into four layers, grouping objects of similar height, and assigning a distinct height value for each layer in the world file. Fig. 4 shows the four layers into which the Coronary ICU environment was divided. The layers were organized so that (a) had the lowest height and (d) the highest.

This approach allowed considering objects of different heights, creating a scenario that more accurately represented a real hospital environment.

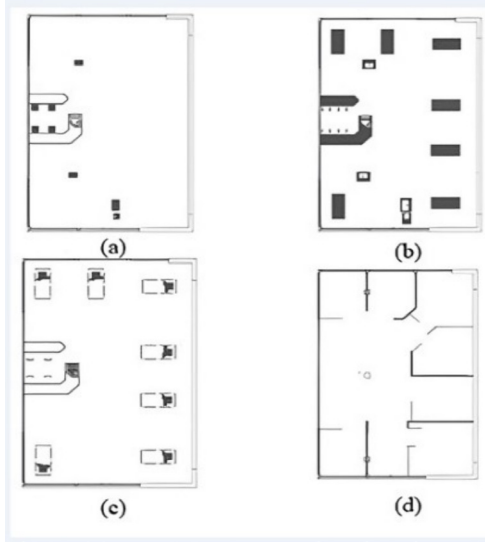

Figure 4. a) Layer chairs and sinks, b) layer beds and countertops, c) layer pillows and railings, and d) layer walls
Source: Authors

Disinfection routes

The routes contained a charging point PC and two disinfection points PA and PB, which depended on whether the room had been isolated and filled with a patient or with some other highly infectious pathology, requiring a higher level of disinfection. The robot started at PC and followed the route provided towards PA, then to PB, and ended by returning to PC.

The routes were elaborated manually, with steep changes in direction to force the robot to perform 90° rotations about its own axis—these movements are easier to control than curved ones. Figs. 5 and 6 show examples of the proposed routes. The Neonatal ICU environment is route 1, the Emergency Room is route 2 and the Coronary ICU is route 3.

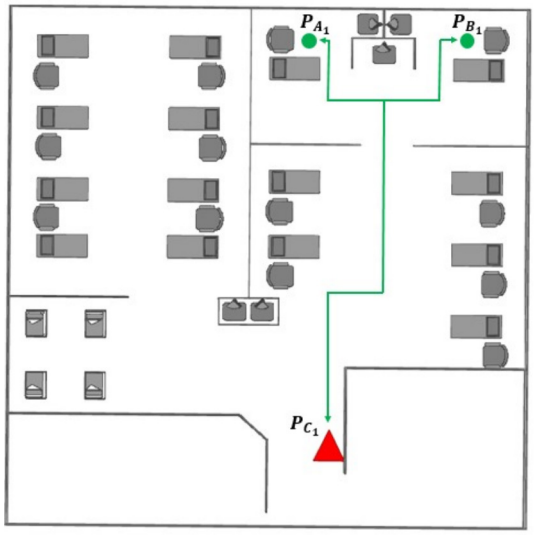

Figure 5. Route 1 proposed for the Neonatal ICU (UMIT)
Source: Authors

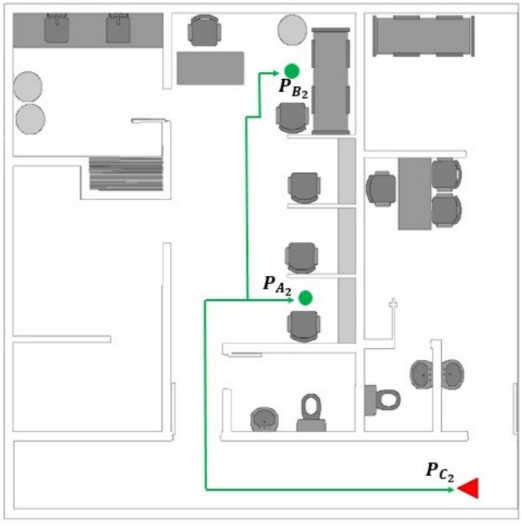

Figure 6. Route 2 planned for the Emergency Department (Tolima Clinic)
Source: Authors

Scanner ranges

Scanner ranges for linear movement

To execute the forward linear movement, the robot was assigned a linear default velocity (Vd) of 1 m/s, without lateral displacement. The robot maintains this movement while extracting the information provided by the scanner, from which the subset of lasers between -5 and 5° is used (Fig. 7). The reason for using such a small subset of lasers and averaging their information was to decrease the computational cost of using all the lasers in the scanner. Additionally, it is not convenient to use a single laser or a short range of lasers because one or more may fail at the same time. These velocities are consistent with those of commercial robots such as the Nomad Super Scout or the Pioneer P3DX [30].

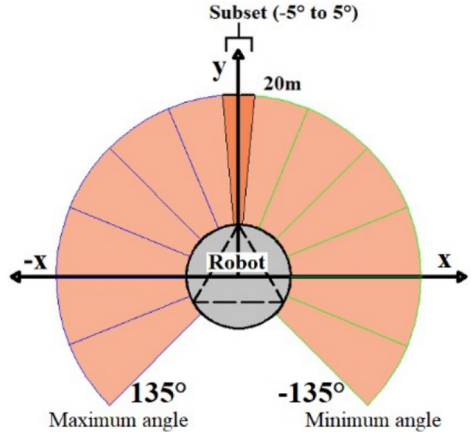

Figure 7. Laser ranges for linear movement
Source: Authors

The distance data extracted from each laser were averaged to obtain a single value that was compared against a minimum allowable distance between the robot and the elements in the scene, in order to determine the change from linear to angular movement. Fig. 8 shows the linear movement algorithm.

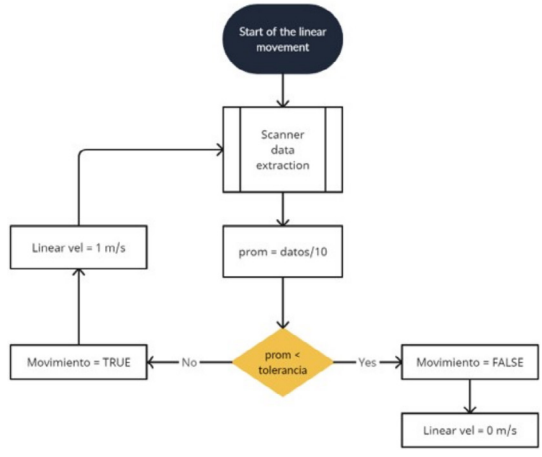

Figure 8. Linear movement algorithm
Source: Authors

Scanner ranges for angular movement

The robot moved angularly about its own axis with an angular default velocity (Wd) of 0.3 rad/s—with no linear velocity (v)—in order to orient itself towards the next linear section of the path. It reduced its angular velocity to 0.1 rad/s in the places where it must perform disinfection, since it had to rotate more slowly to comply with the disinfection time. These low angular velocities were selected for higher accuracy in reading the laser ranges. The scanner ranges used by the robot were between 125 and 135° and between -125 and -135° (Fig. 9).

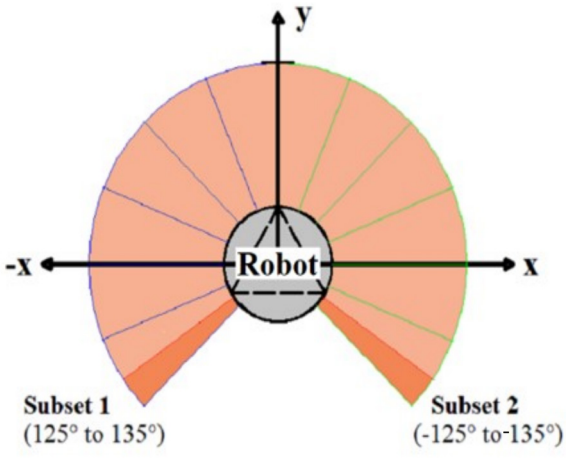

Figure 9. Laser ranges for angular movement
Source: Authors

Fig. 10 shows how the robot uses the scanner subsets to orient itself with the walls on its sides and continue with a linear movement along the desired path.

If the robot is surrounded by asymmetric elements, more adequate laser scanner subsets must be selected in order to generate the necessary rotation towards the next path segment.

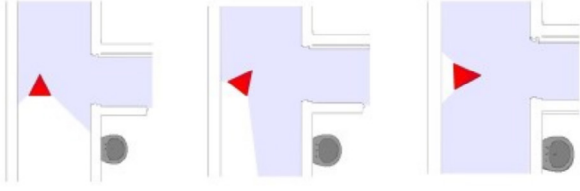

Figure 10. Robot using walls to orient itself
Source: Authors

Fig. 11 illustrates the rotational movement algorithm. In this case, two averages were obtained, which were subtracted. The resulting value was compared against the allowable minimum. If such comparison was satisfactory, the robot would stop rotating; otherwise, it maintained its angular velocity.

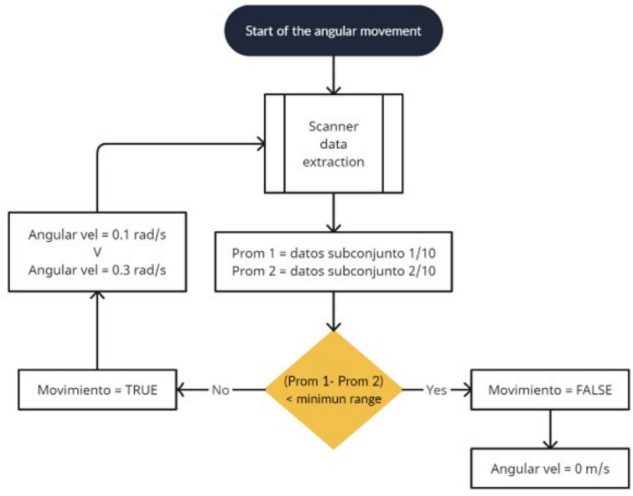

Figure 11. Rotational movement algorithm
Sources: Authors

Tracking algorithm

With the desired route and the scanning strategy, it was possible to elaborate an algorithm for automatically tracking the disinfection route.

Problem statement

The tracking algorithm uses the following input parameters to determine the robot's linear and angular output speeds. Since the scenario is predefined, the user manually provides the vectors that describe the desired path. Additionally, the algorithm receives input from the scanner's detection range, which spans multiple directions. The processed vector inputs are presented below.

Vector inputs

S[i]. This vector represents the average distances of a specific subset of laser scanner data, depending on whether the robot is moving with linear or angular velocity, in accordance with the switching behavior used to follow the desired path:

- When the robot moves with linear velocity, the average is computed using data from the subset between -5 and 5°.
- When the robot moves with angular velocity, the value is calculated as the difference between the averages of two subsets, *i.e.*, 125 to 135° and -125 to -135° (Fig. 11).

SD[i]. The **switch distance vector** defines critical thresholds used by the robot's path-tracking algorithm to determine when to switch between **linear** and **angular** motion. Each element in this vector plays a different role depending on the robot's current mode of movement:

- During linear motion (linear speed, or LS), SD[i] represents the minimum allowable distance between the robot and the obstacle directly in front of it at the end of the current path segment. This value is typically extracted from the hospital's environment map and ensures a safe progression without collision.
- During angular motion (angular speed, or AS), SD[i] is calculated as the difference between two defined groups of average distance readings (Fig. 11). This indicates when the robot should initiate reorientation maneuvers—typically 90 or 180° turns—to align with the next segment of the path. These values are often determined through a manual preliminary exploration of the environment by the robot, in order to account for irregularities in the reference objects or landmarks used for orientation. When the robot aligns itself using a wall (a regular surface), the SD value is close to zero since Prom1 ≈ Prom2 (Fig. 10).

Once the real-time laser scanner readings approach the predefined SD[i] values, the path-tracking algorithm triggers a transition to the appropriate movement mode (linear or angular), ensuring that the robot maintains an accurate path-following behavior.

T[i]. A vector that defines the tolerance distance at which the robot transitions from linear to angular velocity. This parameter is typically close to zero and indicates when the distance readings of vector **S[i]** are sufficiently close to the distance vector **SD[i]**, allowing the robot to initiate its movement transition.

M[i]. This vector defines the robot's predefined iterative movement behavior, set by the user to achieve path following. Each element in the vector can take one of the following values:

- R = Right rotation (90° negative rotation)
- L = Left rotation (90° positive rotation)
- F = Forward movement
- ROA = 180° Rotational movement about its Own Axis (the direction—positive or negative—of the rotation does not affect the tracking algorithm)

In this context, is a parameter that denotes the number of traversals through each segment, covering the path from the initial point to the disinfection points and back to the initial point (reoriented to the initial orientation if required). Thus, it represents the total number of iterations for the input vectors. Accordingly, the iteration index **i** takes values in the range from 1 to **N**.

Output vector

VO[i]. This vector defines the robot's velocity based on the values of M[i] and the switch input vectors. Depending on the required motion, it behaves as follows:

- If the robot needs to move with linear speed, the command sent is LS = Vd, AS = 0 rad/s.
- If the robot needs to rotate in place with angular positive speed, the command sent is LS = 0 m/s, AS = Wd. If it needs to rotate in place with angular negative speed, the command sent is LS =0 m/s, AS = -Wd.

Tracking algorithm flowchart

Once the problem had been clearly defined, we elaborated a tracking algorithm flowchart to visualize the logic required in order for the robot to accurately follow the assigned path. This flowchart is shown in Fig. 12.

Note that **M[i]** determines the movement of the robot at the end of each path segment. The difference between **S[i]** and **SD[i]** defines the time during which the robot maintains a specific behavior. Finally, when the **counter** *i*—which increases when the robot changes its movement—reaches the value of **N**, the algorithm stops, and the mobile robot completes the path-following and disinfection process. Fig. 13 illustrates the path-following stages, where the mobile robot uses a scanner to orient itself towards the next segment of the path. This allows it to accurately reach the disinfection points via the tracking algorithm.

Results

Hospital validation environments

Three scenarios were used in the ROS Stage simulator, which represent different hospital environments. Scenario 1 (Fig. 13) corresponds to the NICU of the Maternal and Child Health Unit of Tolima, scenario 2 (Fig. 14) takes place in the Emergency Room of the Tolima Clinic, and scenario 3 (Fig. 15) involves the Coronary ICU of said clinic.

Validation tests

Once the hospital environments had been simulated and the codes that allowed the robot to follow the disinfection route in Stage had been elaborated, validation tests were conducted to determine the efficiency of the algorithm's proportional control, using the scanner as a sensor. In addition, the proposed automatic tracking procedure and manual route tracking were compared, calculating the error in the robot's position upon reaching the disinfection points.

The XY coordinates (in meters) of the disinfection points determined for each hospital environment are presented below.

PA1 = (-12, 16)	**PB1 =** (-12, 26)	*Scenario 1. NICU*
PA2 = (13.5, 13)	**PB2 =** (-9, 10)	*Scenario 2. ER*
PA3 = (-18, -3,7)	**PB3 =** (30.5, 24)	*Scenario 3. Coronary ICU*

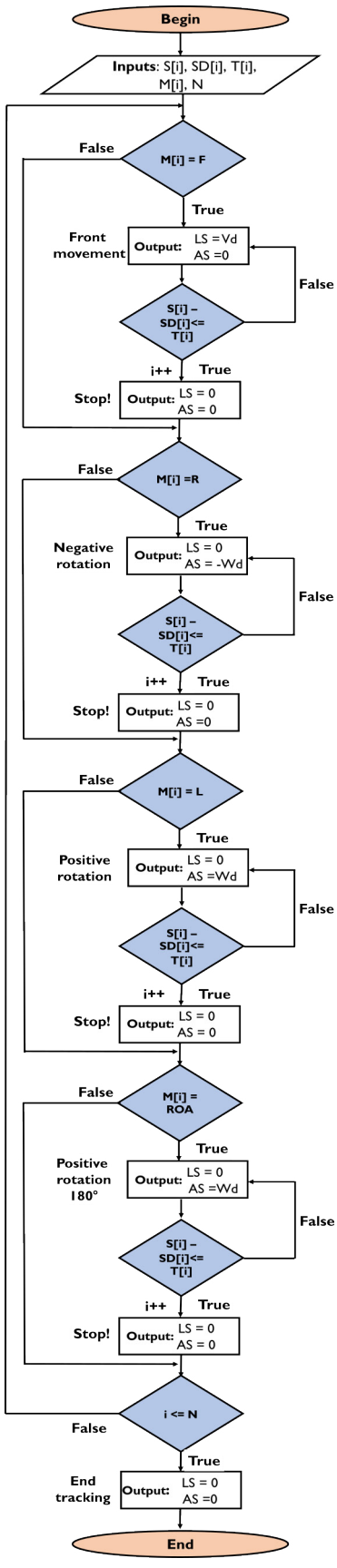

Figure 12. Tracking algorithm flowchart
Sources: Authors

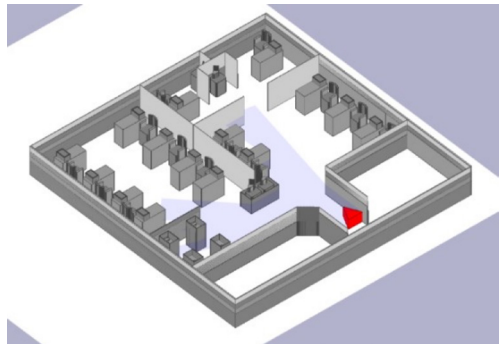

Figure 13. Stage, neonatal ICU, 3D view
Source: Authors

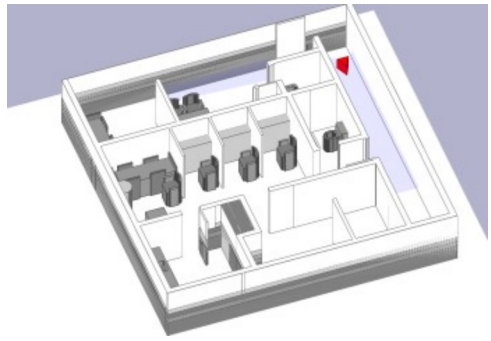

Figure 14. Stage, emergency room, 3D view
Source: Authors

Figure 15. Robot in different phases of the route in scenario 3
Source: Authors

Tables I, II, and III show the results regarding the position of the robot upon reaching the disinfection points for both automatic and manual route tracking.

Table I. Data obtained via automatic and manual tracking in scenario 1

NEONATAL ICU							
Automatic tracking				**Manual tracking**			
PA1		PB1		PA1		PB1	
X	Y	X	Y	X	Y	X	Y
-12.07	16.20	-12.14	25.79	-12.01	16.11	-12.13	26.01
-12.07	16.20	-12.14	25.79	-12.50	16.23	-12.37	26.05
-12.07	16.20	-12.14	25.79	-12.73	16.34	-12.13	25.88
-12.07	16.20	-12.14	25.79	-12.84	16.12	-12.29	25.99
-12.07	16.20	-12.14	25.79	-12.62	15.93	-12.32	25.90

Source: Authors

Table II. Data obtained via automatic and manual tracking in scenario 2

EMERGENCY ROOM							
Automatic tracking				**Manual tracking**			
PA2		PB2		PA2		PB2	
X	Y	X	Y	X	Y	X	Y
13.54	12.71	-9.45	10.11	13.28	12.76	-9.73	10.29
13.54	12.71	-9.45	10.11	13.29	12.22	-9.78	10.51
13.54	12.71	-9.45	10.11	13.09	12.32	-9.74	10.52
13.54	12.71	-9.45	10.11	13.29	11.98	-9.34	10.49
13.54	12.71	-9.45	10.11	13.22	12.33	-10.00	9.78

Source: Authors

Table III. Data obtained via automatic and manual tracking in scenario 3

CORONARY ICU							
Automatic tracking				Manual tracking			
PA3		PB3		PA3		PB3	
X	Y	X	Y	X	Y	X	Y
-17.61	-3.76	30.47	24.07	-16.48	-4.07	30.01	22.90
-17.61	-3.76	30.47	24.07	-17.01	-4.04	30.06	23.22
-17.61	-3.76	30.47	24.07	-16.60	-4.12	29.70	23.52
-17.61	-3.76	30.47	24.07	-16.59	-4.30	30.23	23.34
-17.61	-3.76	30.47	24.07	-16.53	-4.24	30.21	23.55

Source: Authors

Manual tracking was employed to simulate the teleoperation of the robot by the hospital staff. Stage has a function that allows this kind of simulation by means of the computer keyboard. Five tests were run for each route tracking method and scenario, with the purpose of obtaining the error percentages with respect to the theoretical coordinates. The results are shown in Fig. 16.

With the error percentages, the average error was calculated, resulting in values of 1.87% for automatic tracking and 4.82% for manual tracking. It should be highlighted that we recorded no collision with the elements in the hospital environments during automatic tracking.

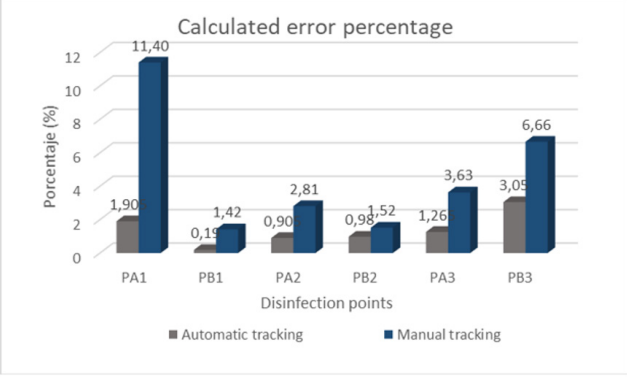

Figure 16. Error rates for automatic and manual route tracking
Source: Authors

Discussion

As mentioned above, the position error obtained through automatic tracking was 1.87%, which is comparable to—and in some cases better than—the results of other studies on mobile robot path following. For example, [31] reported an average tracking error of 4.95% when comparing different path planners in the Gazebo simulator (ROS). [32] achieved a 1.5% error with static obstacles and 10% with dynamic ones while using fuzzy logic methods in MATLAB-simulated environments. [33] implemented path-planning on the commercial Robotino robot, reporting position errors between 1 and 7%. However, these studies used simplified environments with few obstacles and did not address route-tracking algorithms.

To the authors' knowledge, no similar studies have proposed a path-tracking algorithm for differential-drive mobile robots using a scanner for orientation in custom-built environments within Stage (ROS). While scanners have been used in recent robotics research for navigating and mapping unknown environments [31], [34], [35], no specific methods for route tracking have been presented. Additionally, these applications often face difficulties related to storing and managing map point data. In this work, such data challenges were addressed by using specific subsets of the scanner's range, in order to orient the robot through a known 3D map.

The proposed methodology can be adapted to other applications, such as medicine transport in hospitals, warehouse automation, and building security. However, future work must consider dynamic obstacles for real-world implementation.

Conclusions

This work proposed an algorithm for the automatic tracking of disinfection paths in hospital environments. This method was validated using models of the Tolima Clinic and the Maternal and Child Health Unit of Tolima (Colombia) within the Stage simulator (ROS). For the proportional control that allows for tracking, scanners were used as proximity sensors. The characteristics of this work facilitate the transition from simulation to real application. The only thing that must be modified in the code is the robot topics used. In recent related works [32], [33], scanners have also been the preferred sensor in different robotics applications (LiDAR and its variations).

The automatic tracking system yielded a percent error of 1.87%, lower than the manual tracking method's 4.82%. This demonstrates the superior efficiency of the former, as it reduces human error. The precision achieved by the automatic system is particularly notable, considering that the disinfection device must be positioned at a specific distance due to its limited effective range. Although an additional estimation process is needed to calculate the values of the SD[i] vector, once adjusted, the robot can autonomously

follow the disinfection path iteratively by means of the tracking algorithm. This allows healthcare workers to reduce their exposure to pathogens and focus on other urgent tasks, rather than teleoperating the robot, especially in intensive care settings.

Although our path tracking algorithm was used for hospital disinfection, it can be applied to other scenarios and applications, such as surveillance, warehouse logistics, and more. As future work, the path-following algorithm could be enhanced by automating certain user inputs (*e.g.*, the definition of the desired path) based on environment maps.

Acknowledgements

We would like to thank Universidad Autónoma de Manizales and the Research Group in Automatics for the opportunity to carry out this project.

CRediT author statement

Author 1: investigation, formal analysis, software, validation, writing (original draft).
Author 2: formal analysis, software, validation.
Author 3. Manuscript translation
Author 3: conceptualization, investigation, methodology, software, supervision, writing (review and editing).

Access to research data

The datasets generated and/or analyzed during this study are available from the authors upon reasonable request.

Conflicts of interest

The authors declare that they have no conflict of interest.

Statement on artificial intelligent

The authors used ChatGPT to improve the grammar and clarity of certain paragraphs in the text. After using this tool, they reviewed and edited the content as required. The authors take full responsibility for the contents of this publication.

References

[1] R. Gassert and V. Dietz, "Rehabilitation robots for the treatment of sensorimotor deficits: A neurophysiological perspective," J. NeuroEng. Rehabil., vol. 15, no. 1, art. 46, 2018. https://doi.org/10.1186/s12984-018-0383-x

[2] D. K. Agarwal *et al.*, "Initial experience with da Vinci single-port robot-assisted radical prostatectomies," *Eur. Urol.*, vol. 77, no. 3, pp. 373–379, 2020. https://doi.org/10.1016/j.eururo.2019.04.001

[3] A. Begić, "Application of service robots for disinfection in medical institutions," in *Advanced Technologies, Systems, and Applications II*, M. Hadžikadić and S. Avdaković, S., Eds. Cham, Germany: Springer, 2017, pp. 1056–1065, 2017. https://doi.org/10.1007/978-3-319-71321-2_89

[4] K. L. N., D. N. M. Kumaran, G. R., H. Arshadh, I. D., and V. C., "Design and fabrication of medicine delivery robots for hospitals," *Proc. Int Conf. Recent Trends Comp. Comm. Net. Tech.* (*ICRTCCNT*), 2019, art. 3432156. https://doi.org/10.2139/ssrn.3432156

[5] T. Thinh and H. Nguyen, "Telemedicine mobile robot - Robots to assist in remote medical," *Int. J. Mech. Eng. Robot. Res.*, vol. 10, pp. 337–342, 2021. https://doi.org/10.18178/ijmerr.10.6.337-342

[6] Z. H. Khan, A. Siddique, and C. W. Lee, "Robotics utilization for healthcare digitization in global COVID-19 management," *Int. J. Environ. Res. Public Health*, vol. 17, no. 11, art. 3819, 2020. https://doi.org/10.3390/ijerph17113819

[7] World Health Organization, "Cleaning and disinfection of environmental surfaces in the context of COVID-19," 2020. [Online]. Available: https://www.who.int/publications/i/item/cleaning-and-disinfection-of-environmental-surfaces-inthe-context-of-covid-19

[8] G. Z. Yang, "Combating COVID-19—The role of robotics in managing public health and infectious diseases," Sci. Robot., vol. 5, no. 40, art. 5589. https://doi.org/10.1126/scirobotics.abb5589

[9] A. G. Sánchez, N. Bernhart, and W. D. Smart, "Improving UV disinfection of objects by a robot using human feedback," in *Proc. 33rd IEEE Int. Conf. Robot. Hum. Interact. Commun.* (*RO-MAN*), Pasadena, CA, USA, 2024, pp. 1697–1704. https://doi.org/10.1109/RO-MAN60168.2024.10731182

[10] A. M. Padilla Ericksen, "Implementación y comparativa de algoritmos de control y planificación local para robots móviles utilizando ROS," Bachelor's thesis, E.T.S.I. Industriales (UPM), Madrid, Spain, 2017. [Online]. Available: https://oa.upm.es/48146/

[11] B. Ramalingam *et al.*, "A human support robot for the cleaning and maintenance of door handles using a deep-learning framework," *Sensors*, vol. 20, no. 12, art. 3543, 2020. https://doi.org/10.3390/s20123543

[12] D. Hu, H. Zhong, S. Li, J. Tan, and Q. He, "Segmenting areas of potential contamination for adaptive robotic disinfection in built environments," *Build. Environ.*, vol. 184, art. 107226, 2020. https://doi.org/10.1016/j.buildenv.2020.107226

[13] M. V. Shikhman and S. V. Shidlovskiy, "Algorithms and models of multirobot systems and their implementation in ROS," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 696, art. 012015, 2019. https://doi.org/10.1088/1757-899X/696/1/012015

[14] Y. Song, T. Zhang, B. Li, and H. Huang, "A virtual experiment platform for 2d robot autonomous navigation algorithm system based on ROS," in *Proc. IEEE Int. Conf. Inf. Autom.* (*ICIA*), 2018, pp. 985–990. https://doi.org/10.1109/ICInfA.2018.8812455

[15] G. Álvarez and O. Flor, "Desempeño en métodos de navegación autónoma para robots móviles," *Minerva*, vol. 1, no. 2, pp. 19–29, 2020. https://doi.org/10.47460/minerva.v1i2.8

[16] Z. Yao, N. Ma, and Y. Chen, "An autonomous mobile combination disinfection system," *Sensors*, vol. 24, no. 1, art. 53, 2024. https://doi.org/10.3390/s24010053

[17] X. Yun and Y. Yamamoto, "Internal dynamics of a wheeled mobile robot," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.* (*IROS '93*), vol. 2, pp. 1288–1294, 1993. https://doi.org/10.1109/IROS.1993.583753

[18] B. Pappas, "Multi-robot frontier based map coverage using the ROS environment," M.S. thesis, Auburn Univ., Auburn, AL, USA, 2014. [Online]. Available: https://etd.auburn.edu/bitstream/handle/10415/4058/BrianThesis.pdf

[19] J. Kerr and K. Nickels, "Robot operating systems: Bridging the gap between human and robot," in *Proc. 44th Southeastern Symp. Syst. Theory* (*SSST*), 2012, pp. 99–104. https://doi.org/10.1109/SSST.2012.6195127

[20] A. Araújo, D. Portugal, M. S. Couceiro, J. Sales, and R. P. Rocha, "Desarrollo de un robot móvil compacto integrado en el middleware ROS," *Rev. Iberoam. Autom. Inform. Ind.* (*RIAI*), vol. 11, no. 3, pp. 315–326, 2014. https://doi.org/10.1016/j.riai.2014.02.009

[21] ROS Wiki, "SimulatingOneRobot," [Online]. Available: https://wiki.ros.org/stage/Tutorials/SimulatingOneRobot

[22] M. F. Jaramillo-Morales, S. Dogru, L. Marques, and J. B. Gomez-Mendoza, "Predictive power estimation for a differential drive mobile robot based on motor and robot dynamic models," in *Proc. 3rd IEEE Int. Conf. Robot. Comput.* (*IRC*), 2019, pp. 301–307. https://doi.org/10.1109/IRC.2019.00056

[23] M. F. Jaramillo-Morales, S. Dogru, and L. Marques, "Energy optimal speed profiles for a differential drive mobile robot with payload," *J. Optim. Theory Appl.*, vol. 204, art. 17, 2025. https://doi.org/10.1007/s10957-024-02590-46

[24] S. Gatesichapakorn, J. Takamatsu, and M. Ruchanurucks, "ROS-based autonomous mobile robot navigation using 2D LiDAR and RGB-D camera," in *Proc. ICA-SYMP.*, 2019, pp. 151–154. https://doi.org/10.1109/ICA-SYMP.2019.8645984

[25] C. Jinadatha *et al.*, "Is the pulsed xenon ultraviolet light no-touchdisinfection system effective on methicillin-resistant Staphylococcus aureus in the absence of manual cleaning?" *Am. J. Infect. Control*, vol. 43, no. 8, pp. 878–881, 2015. https://doi.org/10.1016/j.ajic.2015.04.005

[26] J. H. Yang, U. I. Wu, H. M. Tai, and W. H. Sheng, "Effectiveness of an ultraviolet-C disinfection system for reduction of healthcare-associated pathogens," *J. Microbiol. Immunol. Infect.*, vol. 52, no. 3, pp. 487–493, 2019. https://doi.org/10.1016/j.jmii.2017.08.017

[27] M. Buonanno, D. Welch, I. Shuryak, and D. J. Brenner, "Far-UVC light (222 nm) efficiently and safely inactivates airborne human coronaviruses," *Sci. Rep.*, vol. 10, art. 10285, 2020. https://doi.org/10.31984/oactiva.v5i3.501

[28] C. S. Heilingloh *et al.*, "Susceptibility of SARS-CoV-2 to UV irradiation," *Am. J. Infect. Control*, vol. 48, no. 10, pp. 1273–1275, 2020. https://doi.org/10.1016/j.ajic.2020.07.031

[29] B. C. Natali, Z. T. Miriam, F. Fabricio, and C. Katherine, "Luz ultravioleta para desinfección en áreas de salud, frente al covid-19. revisión de literatura," *Rev. Oactiva UC Cuenca*, vol. 5, no. 3, pp. 107–114, 2020. https://doi.org/10.31984/oactiva.v5i3.501

[30] Webots Documentation, "Adept's Pioneer 3-DX," 2020. [Online]. Available: https://cyberbotics.com/doc/guide/pioneer-3dx

[31] S. Kim Min, R. Delgado, and W. Choi Byoung, "Comparative study of ROS on embedded system for a mobile robot," *J. Autom. Mobile Robot. Intell. Syst.*, vol. 2012, no. 3, pp. 61–67, 2018. https://doi.org/10.14313/JAMRIS_3-2018/19

[32] H. Baez *et al.*, "Desarrollo y simulación de la evasión y navegación de robots móviles utilizando toolboxs de Matlab," in *Proc. LACCEI 2018*, 2018, art. 124. https://doi.org/10.18687/LACCEI2018.1.1.124

[33] A. I. Yandún Torres and N. G. Sotomayor, "Planeación y seguimiento de trayectorias para un robot móvil," Escuela Politécnica Nacional, 2012. [Online]. Available: http://bib-digital.epn.edu.ec/handle/15000/4913

[34] J. Zhao, S. Liu, and J. Li, "Research and implementation of autonomous navigation for mobile robots based on SLAM algorithm under ROS," *Sensors*, vol. 22, no. 11, art. 4172, 2022. https://doi.org/10.3390/s22114172

[35] L. Nwankwo, C. Fritze, K. Bartsch, and E. Rueckert, "ROMR: A ROS-based open-source mobile robot," 2025. [Online]. Available: https://doi.org/10.17605/OSF.IO/K83X7