www.editada.org

International Journal of Combinatorial Optimization Problems and Informatics, 17(5), 2026, 153-167. ISSN: 2007-1558.

https://doi.org/10.61467/2007.1558.2026.v17i5.1531

Fuzzy Logic and Computer Vision-Based Control Strategy for Line Following in an Omnidirectional Autonomous Mobile Robot

Jacinto González-Aguilar, Oscar Castillo, Prometeo Cortés-Antonio*

División of Graduate Studies, Tijuana Institute of Technology-Tecnológico Nacional de México, Tijuana, México

jacinto.aguilar@tectijuana.mx, ocastillo@tectijuana.mx, prometeo.cortes@tectijuana.edu.mx

✉Corresponding author: Prometeo Cortés-Antonio

Abstract. This article presents the design of a vision-based line-following control strategy for an omnidirectional mobile robot operating on surfaces marked by lines. The proposed approach combines computer vision techniques with a Mamdani-type fuzzy controller to generate motion commands directly from visual feedback. Image acquisition is performed using an onboard webcam with a resolution of 320 × 240 pixels, enabling continuous real-time capture. The Prewitt filter is applied to detect edges and extract the position of the line from a defined region of interest. Based on the processed images, the lateral deviation of the robot with respect to the image centre is computed and used as input to the fuzzy controller. The controller generates forward linear and angular velocity commands following a vision-based fuzzy feedback control. Simulation results obtained using an omnidirectional mobile robot model demonstrate stable and smooth line-following behaviour, showing that the proposed system is capable of tracking the trajectory without losing the reference line under the tested conditions.

Keywords: Computer vision, Prewitt filter, fuzzy control, line tracking, mobile robotics.

Article information
Received: Jun 3, 2026
Accepted: Aug 23, 2026

1 Introduction

Computer vision has become a highly relevant approach for line-following tasks in the field of mobile robotics (Wada & Mori,1996; Masmoudi et al., 2016). For this reason, this paper addresses the problem of line following in mobile robots, since this capability is widely employed in industrial, educational, and research-oriented applications (Oltean et al., 2010). In previous studies, line-following systems have commonly been implemented using classical control techniques or discrete sensors, such as PID controllers (Kalmár-Nagy et al., 2002; Sekhar et al., 2026; González-Aguilar et al., 2020; Castillo et al., 2020). However, these strategies often present limitations when facing variations in lighting conditions, visual noise, or trajectories with sharp curves.

In recent years, computer vision techniques have been increasingly applied to overcome these limitations (Sarwade et al., 2019; Casadiego López, 2020; Rubio et al., 2018). In this context, vision-based systems provide richer and more reliable information about the environment. At the same time, fuzzy control offers a robust control alternative, as it does not require an exact mathematical model of the system (Dirik et al., 2021; Caballero Julián, et al., 2023; Ma'arif, et al., 2020; Asfora, 2021; Yang et al., 2016; Esquivel-Godoy et al., 2022; Castillo & Kumar, 2023). Therefore, this work proposes the combination of edge detection using the Prewitt filter with a Mamdani-type fuzzy controller to achieve stable and effective line-following performance.

The motivation for this study is the development of a vision-based fuzzy control strategy capable of operating without an explicit mathematical model of the robot dynamics. By using a camera as the primary sensor, the proposed approach directly extracts relevant information from the environment and adapts the robot motion in real time based on the detected line position.

This design allows the system to handle visual disturbances and trajectory changes while maintaining stable behaviour during the line-following task.

The main contributions of this work are: (i) the integration of a Prewitt-based line detection method with a Mamdani fuzzy controller for an omnidirectional mobile robot, (ii) a vision-based control architecture that generates velocity commands directly from image information, and (iii) a systematic evaluation using different trajectories.

The remainder of this paper is organized as follows. Section 2 describes the kinematic model of the omnidirectional mobile robot used in this research. Section 3 presents the architecture of the Vision-Based Control, including its input and output variables. Section 4 explains the image processing method employed for line detection. Section 5 details the components of the designed fuzzy control system, including membership functions, error definitions, and output variables. Section 6 presents the experimental results obtained in this study. Finally, Section 7 provides the conclusions and outlines directions for future work.

2 Mobile Robots and Problem Definition

An omnidirectional mobile robot is characterized by having three degrees of freedom and at least three independently actuated wheels, which allows it to perform translational and rotational motion in any direction (Wada & Mori, 1996). By properly adjusting the angular velocity of each wheel, the robot can move forward, backward, laterally, or rotate about its own axis. The combined motion of the three wheels enables full manoeuvrability on a planar Surface.

In this work, an omnidirectional mobile robot is used as the experimental platform for the implementation and evaluation of the proposed fuzzy controller (Masmoudi et al., 2016; Oltean et al., 2010; Kalmár-Nagy et al., 2002; Kalmár-Nagy et al., 2002). This type of robot is particularly suitable for testing control strategies due to its high mobility and nonlinear kinematic behaviour.

Figure 1 illustrates the kinematic model of the robot. The angle ϕ represents the robot's orientation, defined as the counterclockwise rotation with respect to a global reference frame fixed to the floor. The local coordinate system (x, y) is located at the centre of the robot, as shown in the figure.

The variables used in the kinematic model are defined as follows: R is the distance from the centre of each wheel to the centre of the robot; r: radius of the wheels; $\dot{x}$, $\dot{y}$ are the linear velocities of the robot in the global x and y directions; $\dot{\phi}$ is the angular velocity of the robot (yaw rate); $\dot{\theta}_1, \dot{\theta}_2, \dot{\theta}_3$ are the angular velocities of wheels 1, 2, and 3, respectively; φ_i denotes the fixed angular orientation of the arm that holds the i-th wheel with respect to the forward direction of the robot platform.

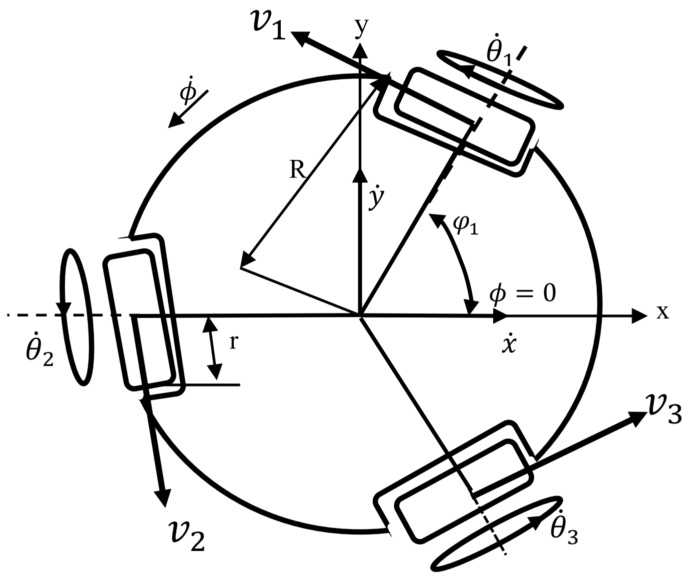


Fig. 1. Kinematic model of the omnidirectional mobile robot.

The kinematic relationship between the robot velocities and the wheel angular velocities is given by:

$$\begin{bmatrix}\dot{\theta}_1\\ \dot{\theta}_2\\ \dot{\theta}_3\end{bmatrix}=\frac{1}{\mathrm{r}}\begin{bmatrix}v_1\\ v_2\\ v_3\end{bmatrix}=\frac{1}{\mathrm{r}}\overbrace{\begin{bmatrix}-sen(\phi+\varphi_1) & cos(\phi+\varphi_1) & \mathrm{R}\\ -sen(\phi+\varphi_2) & cos(\phi+\varphi_2) & \mathrm{R}\\ -sen(\phi+\varphi_3) & cos(\phi+\varphi_3) & \mathrm{R}\end{bmatrix}}^{\mathrm{P}(\phi)}\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{\phi}\end{bmatrix} \quad (1)$$

Where (x, y, ϕ) define the robot's position and orientation in the global coordinate system, and $(\dot{x},\dot{y},\dot{\phi})$ represent its linear and angular velocities.

The robot velocities can also be obtained from the wheel angular velocities by inverting the matrix $\mathrm{P}(\phi)$, resulting in:

$$\begin{bmatrix}\dot{x}\\ \dot{y}\\ \dot{\phi}\end{bmatrix}=\mathrm{r}\mathrm{P}^{-1}(\phi)\begin{bmatrix}\dot{\theta}_1\\ \dot{\theta}_2\\ \dot{\theta}_3\end{bmatrix} \quad (2)$$

Through the relation expressed in Equation 2, it is possible to estimate the translational velocity along the x and y axes, as well as the rotational velocity $\dot{\phi}$,from the wheel velocities. Equation 2 constitutes the basis for odometry estimation and trajectory-tracking control of the omnidirectional mobile robot.

3 Proposed vision-based fuzzy feedback control

This section describes the proposed fuzzy control strategy used for the line-tracking task. The controller is designed to regulate the motion of an omnidirectional mobile robot using information obtained directly from a vision-based sensing system. The control scheme is based on two key input variables derived from image processing: the lateral error and the change in error.

The control architecture follows a feedback control structure, in which the controller generates velocity commands directly from the visual information without relying on a closed-loop position controller. In this approach, the camera acts as the main sensor, providing real-time feedback that is processed and transformed into motion commands. The fuzzy controller interprets the visual information and produces control actions that guide the robot along the desired trajectory.

Figure 2 illustrates the proposed control scheme. In this work, only the linear velocity in the forward direction v_x and the angular velocity v_a are controlled. The lateral velocity v_y is not considered in the control design and is set to zero. Although the robot is omnidirectional, preliminary experiments showed that lateral corrections produced irregular motion because the onboard camera is fixed in the forward direction, which resulted in unnatural trajectories and higher demands on the actuators. Moreover, the fuzzy inference structure became more complex without yielding significant improvements in tracking accuracy. For these reasons, the proposed design adopts a simplified non-holonomic-like operating mode that maintains stable and effective line-following behaviour

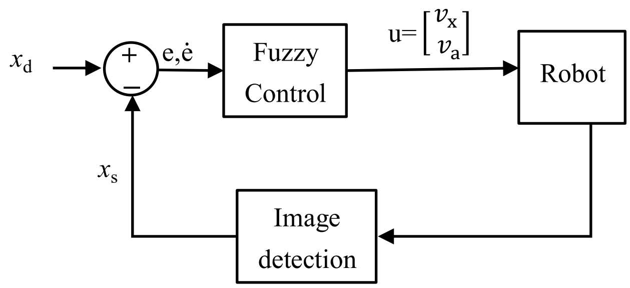


Fig. 2. Vision-Based Fuzzy feedback Control Scheme.

Figure 2, shows the variables involved in the control scheme: where x_s correspond to a sensor signal computed from the camera (measured line position); x_d=160 indicates the reference signal corresponding to the center of the image, v_x linear velocity along the forward direction of the robot; and v_a angular velocity of the robot.

For the line-tracking task, the error represents the horizontal deviation of the detected line with respect to the center of the image. The lateral error is computed as e=x_d-x_s. This error, together with its temporal variation, is used as input to the fuzzy controller to determine the appropriate velocity commands.

4 Image processing

The image-processing method used to detect the line and obtain the information required by the control system is shown in Fig. 3. The objective of this stage is to estimate the horizontal position of the line from the captured images and convert it into numerical data suitable for control.

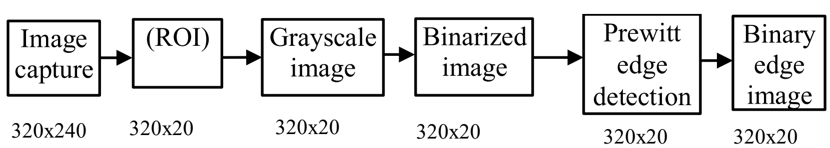


Fig. 3. Vision-based image processing pipeline for line position estimation

4.1 Image Acquisition and Preprocessing

Image acquisition is performed using a webcam with a resolution of 320×240 pixels, which is integrated into the robot platform. Images are captured continuously in real time during robot operation.

To reduce computational cost and improve robustness, a region of interest (ROI) is defined. The ROI corresponds to a horizontal strip located at the bottom of the image, with a size of 320×20 pixels. This region is selected because it is the area where the line and the robot are in close proximity.

The captured RGB image is converted to grayscale in order to reduce processing complexity and simplify subsequent operations. The Grayscale conversion used in this work was Y=0.299R+0.587G+0.114B.

4.2 Image Binarization

After grayscale conversion, the image is binarized to distinguish the line from the background. This process applies an intensity threshold that classifies each pixel into one of two categories: background or line. The resulting binary image highlights the regions corresponding to the detected line.

4.3 Edge Detection Using the Prewitt Filter

Edge detection is performed using the Prewitt operator, which estimates intensity gradients in both horizontal and vertical directions. The Prewitt masks used in this work are defined as:

$$F_x=\begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \qquad F_y=\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \tag{3}$$

The gradient components are computed by the convolution of the grayscale ROI image, I_{ROI} with the corresponding filters:

$$G_x(x, y)=I_{ROI} * F_x \qquad G_y(x, y)=I_{ROI} * F_y \tag{4}$$

The edge magnitude is computed by combining both gradient components as:

$$Edge(x,y)=\sqrt{\left(G_x(x,y)\right)^2+\left(G_y(x,y)\right)^2} \tag{5}$$

The resulting edge image is binarized using a fixed threshold T, where T=50 in this work:

$$bw(x, y)=\begin{cases} 0, & \text{if} \quad Edge(x,y) \geq T \\ 1, & \text{if} \quad Edge(x,y) < T \end{cases} \tag{6}$$

Table 1 illustrates the binary edge image bw(x,y) for two representative rows of the region of interest. Pixels with a value of one correspond to detected line pixels, while zeros represent background regions. For each valid row, the horizontal position of the line is computed as the centroid of the detected pixels, as defined in the line position estimation. Rows without detected line pixels are excluded from the calculation.

Table 1. Example of binary edge extraction used for centroid-based line position estimation.

bw(x,y)	**…**	**146**	**147**	**148**	**…**	**170**	**172**	**…**	**319**	**320**
y:1	0	0	0	1	1	1	1	0	0	0
y:20	0	0	0	1	1	1	1	0	0	0

4.4 Line Detection and Position Estimation

Once the binary edge image is obtained, the position of the line is estimated using a row-wise centroid calculation. For each row y_j of the ROI, the number of detected line pixels is defined as:

$$S_j=\sum_{i=1}^{320} bw\left(x_i,y_j\right) \tag{7}$$

If $S_{j_i} > 0$, the horizontal position of the line for that row is computed as:

$$line_j=\frac{\sum_{i=1}^{320} bw\left(x_i,y_j\right) i}{S_j} \tag{8}$$

Rows for which no line pixels are detected ($S_{j_i} = 0$) are not considered in the estimation process. Let J denote the set of rows in the ROI for which a valid line position is obtained:

$$J=\left\{j\in[1,20] \mid S_{j_i}>0\right\} \tag{9}$$

If at least one valid row exists (|J|>0), the overall detected line position is computed as the average of the valid row estimates:

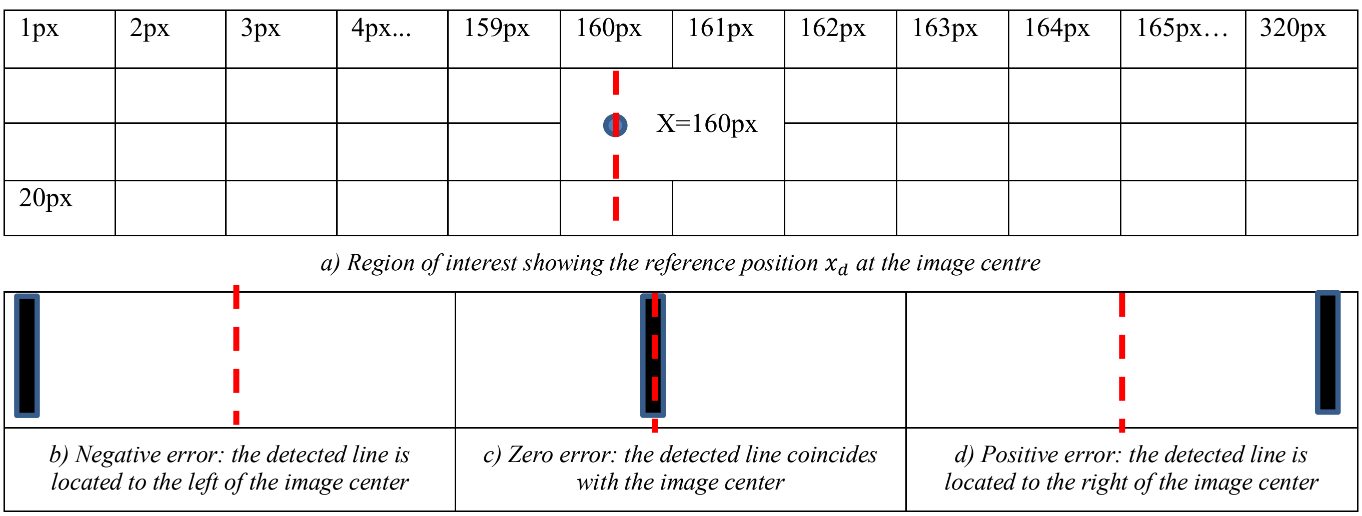
$$\text{line}_{\text{det}}=\frac{1}{|J|}\sum_{j=J}\text{line}_j \tag{10}$$

In the absence of detected line pixels in the region of interest, the algorithm maintains the last valid line position estimate. This mechanism prevents undefined values and contributes to stable behaviour during brief detection failures.

Finally, the error signal is provided to the fuzzy controller using line_{det} as follows:

$$\text{Error} = x_d - \text{line}_{\text{det}} \tag{11}$$

Figure 4 provides a graphical interpretation of the detected line position within the ROI and its relation to the reference position. The dashed vertical line represents the estimated line position, while the image centre defines the reference value. A negative error indicates that the line is located to the left of the image centre, a zero error corresponds to a centred line, and a positive error indicates that the line is located to the right. This error signal is subsequently used as an input to the fuzzy controller described in the next section.



a) Region of interest showing the reference position x_d at the image centre

b) Negative error: the detected line is located to the left of the image center

c) Zero error: the detected line coincides with the image center

d) Positive error: the detected line is located to the right of the image center

Fig. 4. Visual interpretation of line position estimation and error sign used for fuzzy control.

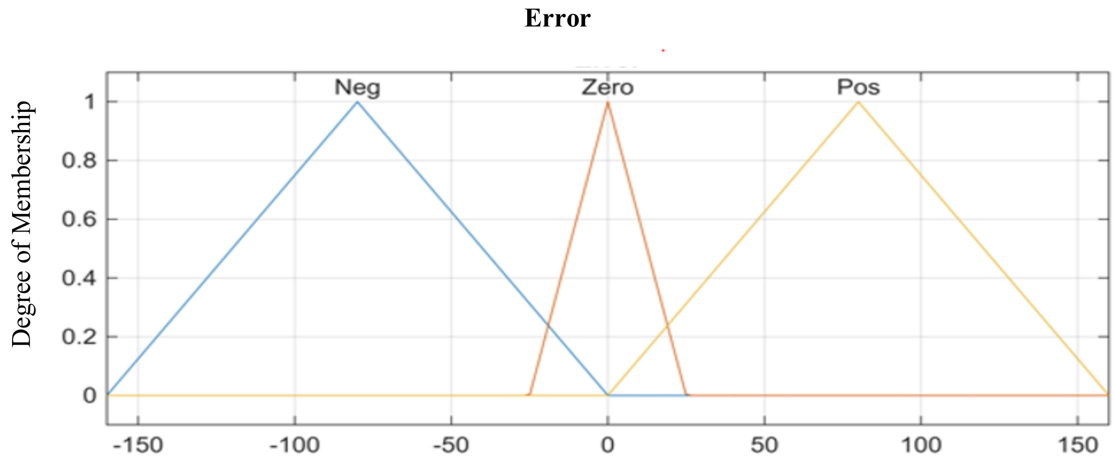
5 Design of the Fuzzy Control System

A fuzzy control system is used to generate the robot motion commands for the line-following task. A Mamdani-type fuzzy inference system is employed due to its intuitive rule-based structure and suitability for control applications.

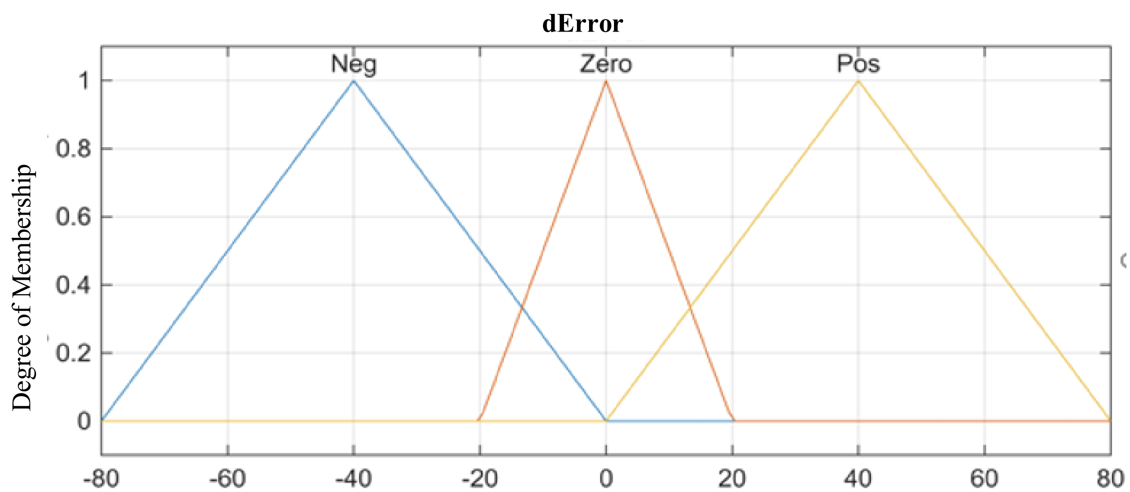
The fuzzy controller has two input variables: the error and the change in error, both obtained from the image-processing stage described in the previous section. The controller produces two output commands: the forward linear velocity v_x and the angular velocity v_a. These outputs are directly applied to the robot motion controller.

Figure 5 shows the membership functions associated with the input variables. The error variable is defined using three linguistic terms: Negative, Zero, and Positive. Similarly, the change in error variable is described using the same three linguistic terms.

Figure 6 presents the membership functions for the output variables. The forward linear velocity v_x is defined using five linguistic terms: Ultra slow, Very slow, Slow, Fast, and Very fast. The angular velocity v_a is defined using three linguistic terms: Negative, Zero, and Positive.

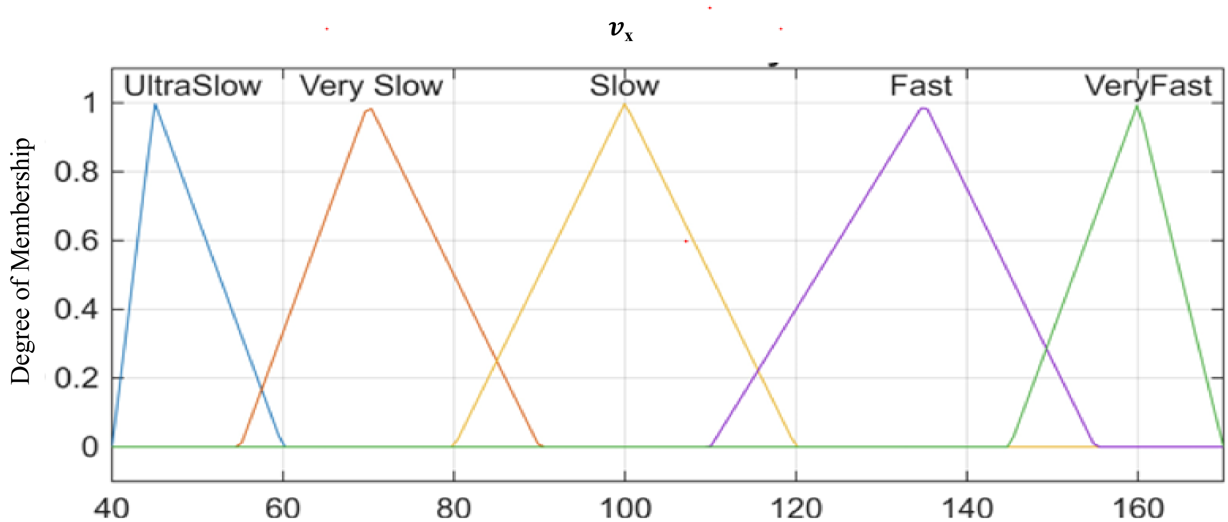


a) Error

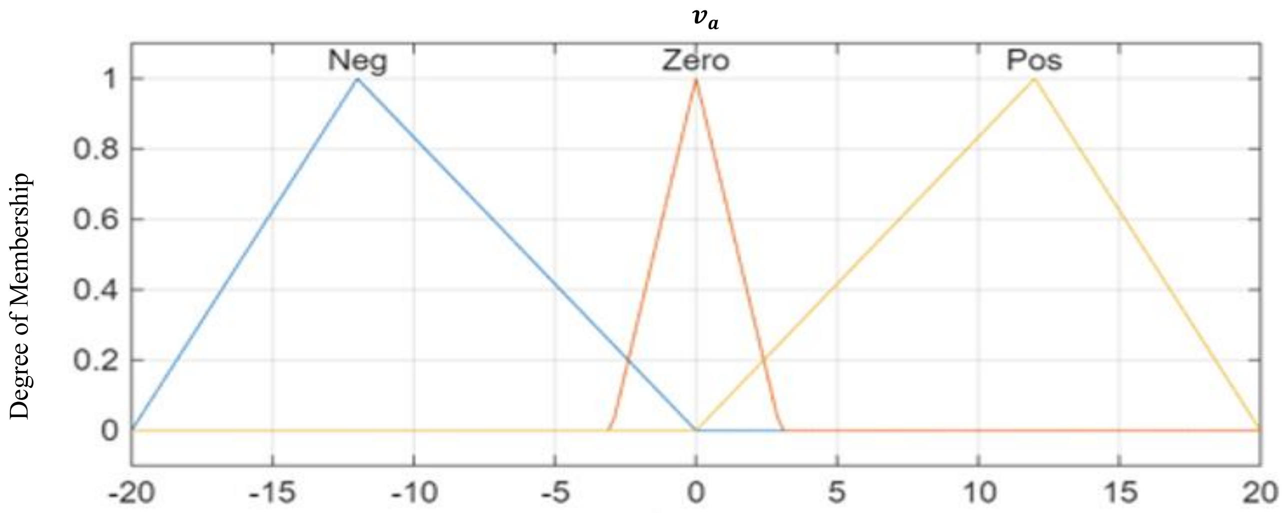


b) Error change

Fig. 5. Fuzzy input variables: (a) error and (b) error change.



a) line velocity in the x-direction



b) angular velocity

Fig. 6. Fuzzy output variables: (a) linear velocity in the x-direction and (b) angular velocity.

The membership functions of the fuzzy controller are defined as triangular sets, with parameters listed in Table 2 for its reproducibility of the fuzzy inference process.

Table 2. Fuzzy inference rule base for line-following control.

Variable	**Linguistic term**	**Parameters**
Error	Negative, Zero, Positive	[−160, −80,0], [−25,0,25], [0 80,160]
Error change	Negative, Zero, Positive	[−80, -40,0], [-20,0,20], [0,40,80]
v_x (linear velocity)	Ultra slow, Very slow, Slow, Fast, Very fast	[40,45,60], [55,70,90], [80,100,120], [110,135,155], [145,160,170]
v_a (angular velocity)	Negative, Zero, Positive	[−20, -12, 0], [-3,0,3], [0,12,20]

The behaviour of the fuzzy controller is defined by a set of nine rules, summarized in Table 3. These rules relate the input linguistic variables to the corresponding velocity commands, allowing the robot to correct its trajectory according to the position of the detected line.

Table 3. Fuzzy inference rule base for line-following control.

Rule	**Error**	**Error Change**	v_x (linear velocity)	v_a (angular velocity)
1	Negative	Negative	Ultraslow	Negative
2	Negative	Zero	Ultraslow	Negative
3	Negative	Positive	Ultraslow	Zero
4	Zero	Negative	Slow	Negative
5	Zero	Zero	Fast	Zero
6	Zero	Positive	Very Fast	Negative

7	Positive	Negative	Very Slow	Zero
8	Positive	Zero	Very Slow	Positive
9	Positive	Positive	Very Slow	Positive

The parameters of the fuzzy inference system are summarized as follows: for rule evaluation (min for AND, max for OR), implication (min), aggregation (max), and centroid defuzzification.

6 Analysis of results

The main objective of this research is the design a vision-based fuzzy control strategy for line following in an omnidirectional mobile robot operating on line-marked surfaces. The proposed approach was evaluated using Robotino® SIM, which provides a virtual Robotino® platform and predefined navigation circuits. Figure 8 shows the two test circuits used in this study.

The onboard Robotino® camera operates at 30 frames per second. The experiments were conducted exclusively in the Robotino® SIM environment. Each frame was processed using grayscale conversion, binarization, Prewitt edge detection, and fuzzy inference, and the resulting error signal was updated at the same rate. Velocity commands were transmitted to the robot controller once per frame. Thus, the feedback pathway operates at the camera frame rate, with a total control cycle of approximately 33–50 ms, as summarized in Table 4.

Table 4. Feedback pathway parameters in experiments.

Parameter	**Value**	**Description**
Camera frame rate	30 fps	Manufacturer's specification (Festo Robotino®).
Frame period	~33 ms	Time between consecutive frames.
Image processing delay	20–30 ms	Grayscale conversion, binarization, Prewitt edge detection.
Fuzzy inference delay	10–15 ms	Fuzzy controller.
Transmission delay	2-5 ms	Command sent to robot controller.
Total control cycle	~33-50 ms	Effective period including acquisition + processing + command transmission.

To evaluate the controller performance, the image-based error, the linear velocity v_x the angular velocity v_a and the robot trajectory were analysed during navigation. The image-based error is measured in pixels and represents the deviation of the detected line from the centre of the image. A value of zero indicates that the line is perfectly centred, whereas positive and negative values indicate deviations to the right and left of the reference position, respectively. The linear velocity v_x varies between 40 and 170 mm/s, while the angular velocity v_a ranges from −20 to 20 deg/s according to the fuzzy control actions.

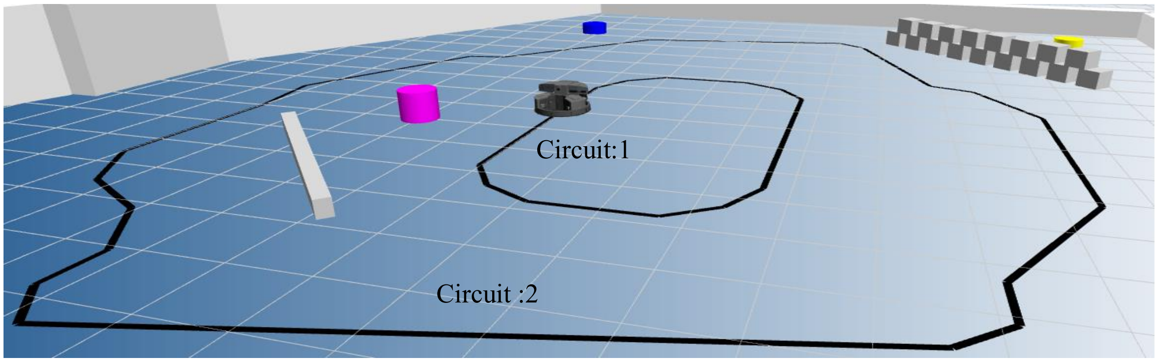
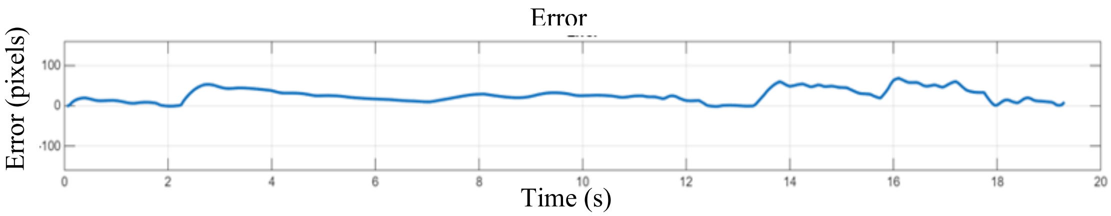
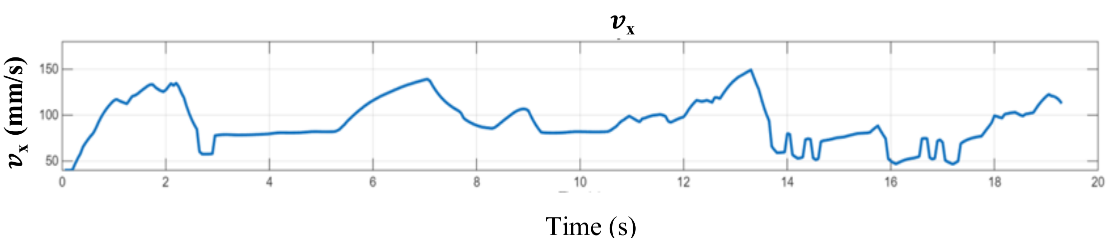


Fig. 7. Test circuits used for evaluating the proposed line-following controller.

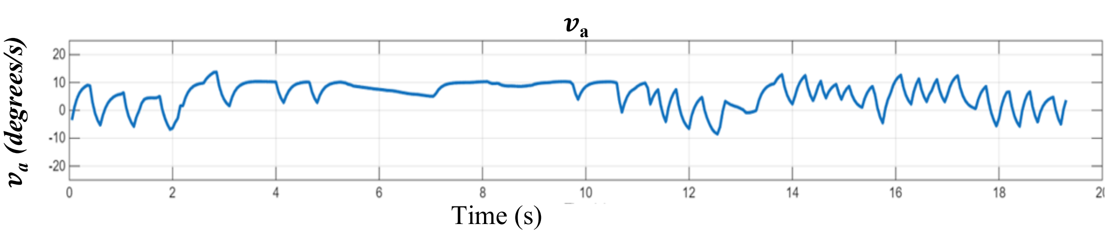
Figures 8 and 9 present the results obtained for Circuits 1 and 2, respectively. Each figure includes the image-based error, the linear velocity, the angular velocity, and the trajectory estimated from the robot odometry. To further evaluate the proposed controller, both circuits were traversed in clockwise and counterclockwise directions.



a) Error



b) Linear velocity



c) Angular velocity

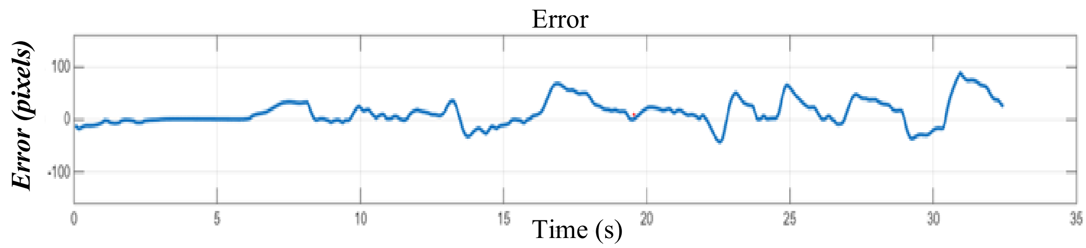
d) Robot path

Fig. 8. Error, velocity responses, and robot trajectory for Circuit 1.

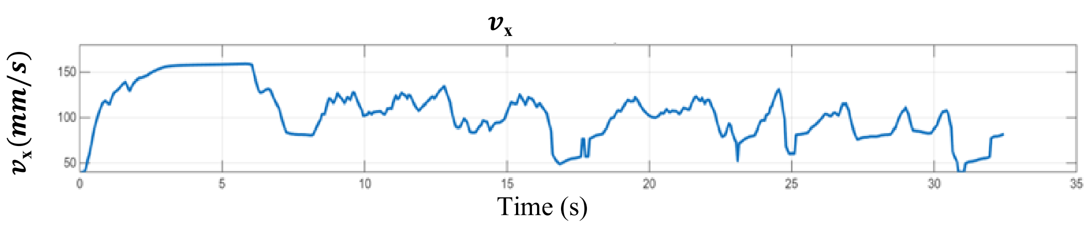
Figure 8(a) shows that the image-based error remains bounded throughout the experiment. As expected, larger deviations occur when the robot encounters sections with higher curvature; however, the controller continuously reduces the error and drives the detected line back toward the centre of the image. This behaviour indicates that the proposed control strategy is capable of maintaining stable line tracking with-out losing the reference trajectory.

The evolution of the linear velocity v_x shown in Fig. 8(b), demonstrates that the robot does not move at a constant speed. Instead, the velocity varies according to the fuzzy rules and the current navigation conditions. Lower velocities are generally observed when larger corrections are required, particularly in curved sections of the path, whereas higher velocities are achieved in straighter sections where the tracking error is small.

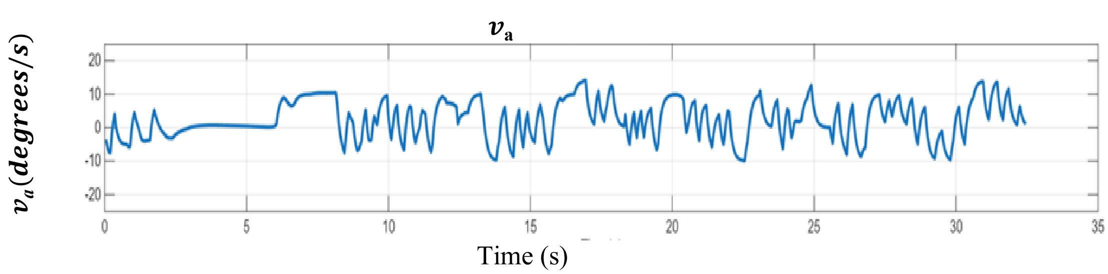
Figure 8(c) presents the angular velocity v_a. Continuous variations in this signal indicate that the controller performs constant heading corrections to maintain alignment with the detected line. The largest changes occur in regions with higher curvature, where more frequent steering actions are required. The corresponding robot trajectory is shown in Fig. 8(d), where the robot successfully follows the prescribed path throughout the experiment.



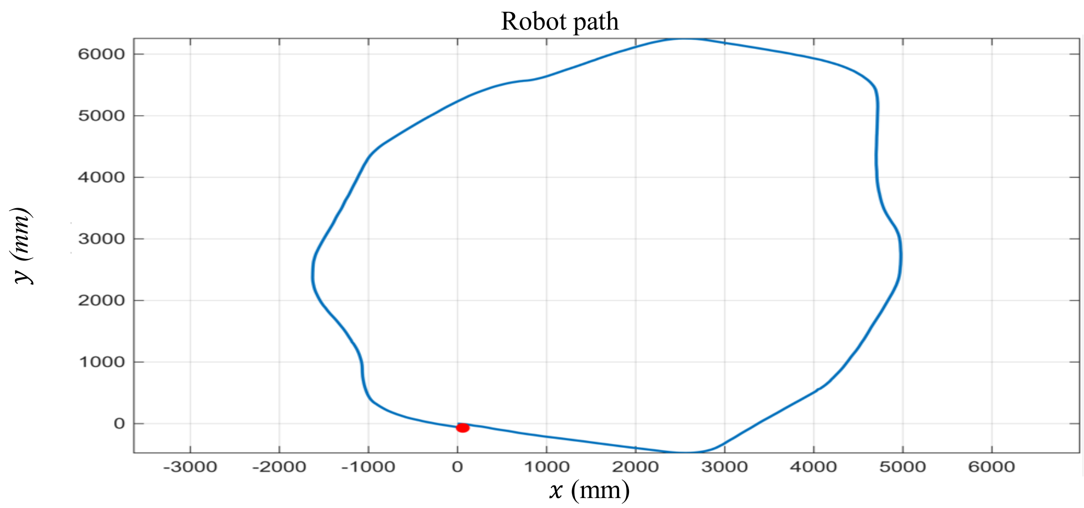
a) Error



b) Linear velocity



c) Angular velocity



d) Robot path

Fig. 9. Error, velocity responses, and robot trajectory for Circuit 2.

A similar behaviour can be observed in Fig. 9 for Circuit 2. In both test scenarios, the image-based error evolves smoothly and remains within acceptable limits, while the velocity commands adapt continuously to the path geometry. These results indicate that the combination of computer vision and fuzzy control provides stable line-following performance for an omnidirectional mobile robot.

Finally, a quantitative evaluation of the controller performance was performed using the Root Mean Square Error (RMSE), defined in (12).

$$\mathrm{RMSE}=\sqrt{\frac{1}{n}\sum_{i=1}^{n}(e_i)^2} \tag{12}$$

where e_i is image-based error measured in pixels at sample i, and n the total number of samples. This metric is computed from the image-based error measured with respect to the center of the camera image and provides an objective measure of the tracking accuracy achieved by the proposed control strategy.

It should be noted that Robotino® SIM incorporates sensor noise, communication delays, and processing-time variations. As a result, slight differences in performance may occur between consecutive runs of the same circuit. Table 5 summarizes the RMSE values obtained for the experiments performed on the test circuits shown in Fig. 8. The symbols (+) and (−) indicate counterclockwise and clockwise directions, respectively.

All RMSE values are expressed in pixels. The RMSE results indicate that the proposed controller achieved consistent performance across all experimental conditions. Although variations are observed between runs, the error values remain within a relatively narrow range for most tests, demonstrating repeatable behaviour despite the uncertainty introduced by the simulation environment.

Table 5. RMSE values obtained for the evaluated circuits under clockwise and counterclock-wise directions.

Test	Circuit 1 (+)	Circuit 1 (-)	Circuit 2 (+)	Circuit 2 (-)
1	24.22	23.70	26.73	20.40
2	23.85	22.81	35.08	19.76
3	22.86	23.83	24.04	20.70
4	22.92	20.73	33.48	20.94
5	23.76	20.73	26.55	21.54
6	22.73	23.04	26.55	21.13
7	24.91	22.39	26.48	22.07
8	22.14	23.03	18.58	20.74
9	21.75	24.04	19.80	21.08
10	23.14	24.79	19.80	21.93

Table 6 summarizes the statistical analysis of the RMSE obtained over ten experimental runs. The results show that the proposed controller maintains consistent line-tracking performance across circuits with different geometric characteristics. Evaluating each circuit in both clockwise and counterclockwise directions provide a more comprehensive assessment of the controller, demonstrating that the tracking behaviour is not biased toward a particular turning direction. Although some variability is observed among the tested scenarios, particularly in Circuit 2 (+), the average RMSE values and standard deviations remain within a relatively narrow range, indicating stable operation under different path conditions. Furthermore, Circuit 2 (−) achieved the lowest average RMSE and the smallest standard deviation, while Circuit 1 exhibited comparable performance in both directions. Overall, these results confirm that the proposed vision-based fuzzy control strategy is capable of maintaining reliable line-tracking performance over trajectories of varying complexity and in both directions.

Table 6. Descriptive statistics for the RMSE obtained by the fuzzy controller during line tracking in the evaluated circuits.

Circuit	Average RMSE	Std.	Min.	Max.
Circuit 1 (+)	23.23	0.96	21.75	24.91
Circuit 1 (-)	22.91	1.34	20.73	24.79
Circuit 2 (+)	25.71	5.52	18.58	35.08
Circuit 2 (-)	21.03	0.70	19.76	22.07

7 Conclusions

In this work, a vision-based line-following strategy for an omnidirectional mobile robot was presented. By combining a Prewitt-based line detection method with a Mamdani-type fuzzy controller, the proposed system was able to generate navigation commands directly from visual information. The simulation results demonstrated reliable line-tracking performance across trajectories of different complexity and traversal directions, showing that the controller can maintain stable operation under diverse navigation scenarios.

Overall, the results support the feasibility of using computer vision and fuzzy logic as an effective framework for autonomous line-following in omnidirectional mobile robots. Future work will focus on real-world implementation, optimization of the fuzzy controller parameters, and the incorporation of Type-2 fuzzy logic techniques to enhance robustness under uncertainty and changing environmental conditions.

In addition, comparative studies with conventional baselines such as PID or rule-based controllers, together with ablation analyses of the change-in-error input and the Prewitt operator, are identified as promising directions to further strengthen and quantify the scientific contribution of the proposed approach.

References

Asfora, B. A. (2021). *Embedded computer vision system applied to a four-legged line follower robot* [Preprint]. arXiv. https://doi.org/10.48550/arXiv.2101.04804

Caballero Julián, F. G., Pérez Solano, M. Á., Hernández Sánchez, C., Ordoñez Gonzaga, L. Y., & Matus Merino, C. A. (2023). Robot manipulador con visión artificial. *Revista IPSUMTEC, 6*(6), 39–49. https://doi.org/10.61117/ipsumtec.v6i6.51

Casadiego López, J. D. (2020). *Implementación de sistema de detección de personas para un robot guía usando visión por computador* [Undergraduate thesis, Universidad de los Andes]. Séneca: Repositorio Institucional de la Universidad de los Andes. https://repositorio.uniandes.edu.co/entities/publication/cefacffa-5ac2-4719-b3ae-b6c34c926c9a

Castillo, O., & Kumar, A. (Eds.). (2023). *Recent trends on type-2 fuzzy logic systems: Theory, methodology and applications*. Springer. https://doi.org/10.1007/978-3-031-26332-3

Castillo, O., Cortés-Antonio, P., Melin, P., & Valdez, F. (2020). Type-2 fuzzy control for line following using line detection images. *Journal of Intelligent & Fuzzy Systems, 39*(5), 6089–6097. https://doi.org/10.3233/JIFS-189081

Dirik, M., Castillo, O., & Kocamaz, F. (2021). *Vision-based mobile robot control and path planning algorithms in obstacle environments using type-2 fuzzy logic*. Springer. https://doi.org/10.1007/978-3-030-69247-6

Esquivel-Godoy, D. J., Nuño-Maganda, M. A., Hernández-Mier, Y., & Polanco-Martagón, S. (2022). Módulos de supervisión y automatización de un sistema hidropónico mediante lógica difusa y visión por computadora. *Difu100ci@, Revista de Difusión Científica, Ingeniería y Tecnologías, 16*(3), 15–22.

González-Aguilar, J., Castillo, O., & Cortés-Antonio, P. (2020). Implementation a fuzzy system for trajectory tracking of an omnidirectional mobile autonomous robot. In O. Castillo & P. Melin (Eds.), *Hybrid intelligent systems in control, pattern recognition and medicine* (pp. 327–340). Springer. https://doi.org/10.1007/978-3-030-34135-0_23

Kalmár-Nagy, T., Ganguly, P., & D'Andrea, R. (2002). Real-time trajectory generation for omnidirectional vehicles. In *Proceedings of the 2002 American Control Conference* (Vol. 1, pp. 286–291). IEEE. https://doi.org/10.1109/ACC.2002.1024818

Ma'arif, A., Nuryono, A. A., & Iswanto. (2020). Vision-based line following robot in Webots. In *2020 FORTEI-International Conference on Electrical Engineering (FORTEI-ICEE)* (pp. 24–28). IEEE. https://doi.org/10.1109/FORTEI-ICEE50915.2020.9249943

Masmoudi, M. S., Krichen Masmoudi, N., Masmoudi, M., & Derbel, N. (2016). Fuzzy logic controllers design for omnidirectional mobile robot navigation. *Applied Soft Computing, 49*, 901–919. https://doi.org/10.1016/j.asoc.2016.08.057

Oltean, S. E., Dulău, M., & Puskas, R. (2010). Position control of Robotino mobile robot using fuzzy logic. In *2010 IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR)* (Vol. 1, pp. 366–371). IEEE. https://doi.org/10.1109/AQTR.2010.5520855

Rubio, Y., Picos, K., Orozco-Rosas, U., Sepúlveda, C., Ballinas, E., Montiel, O., Castillo, O., & Sepúlveda, R. (2018). Path following fuzzy system for a nonholonomic mobile robot based on frontal camera information. In O. Castillo, P. Melin, & J. Kacprzyk (Eds.), *Fuzzy logic augmentation of neural and optimization algorithms: Theoretical aspects and real applications* (pp. 223–240). Springer. https://doi.org/10.1007/978-3-319-71008-2_18

Sarwade, J., Shetty, S., Bhavsar, A., Mergu, M., & Talekar, A. (2019). Line following robot using image processing. In *2019 3rd International Conference on Computing Methodologies and Communication (ICCMC)* (pp. 1174–1179). IEEE. https://doi.org/10.1109/ICCMC.2019.8819826

Sekhar, K. Ch., Shanthi, J. M., Senthil Kumar, K. R., Gopikrishnan, M., Revathi, R., & Dhanraj, J. A. (2026). Design and implementation of an autonomous line-following mobile robot using Q-learning and computer vision. *EPJ Web of Conferences, 354*, Article 03002. https://doi.org/10.1051/epjconf/202635403002

Wada, M., & Mori, S. (1996). Holonomic and omnidirectional vehicle with conventional tires. In *Proceedings of the 1996 IEEE International Conference on Robotics and Automation* (Vol. 4, pp. 3671–3676). IEEE. https://doi.org/10.1109/ROBOT.1996.509272

Yang, H., Wang, Y., & Gao, L. (2016). A general line tracking algorithm based on computer vision. In *2016 Chinese Control and Decision Conference (CCDC)* (pp. 5365–5370). IEEE. https://doi.org/10.1109/CCDC.2016.7531957

Acknowledgments: SECIHTI supports this work.

Data Availability Statement: Not applicable.

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