International Journal of Power Electronics and Drive System (IJPEDS)
Vol. 17, No. 4, December 2026, pp. 2592~2602
ISSN: 2088-8694, DOI: 10.11591/ijpeds.v17.i4.pp2592-2602

Autonomous forklift through intelligent navigation algorithms

Juan C. Guachetá-Alba[1], Robinson Jiménez-Moreno[2], Anny Astrid Espitia-Cubillos[3]
[1]Mechanical Engineering Program, Engineering Faculty, Universidade Federal do Rio de Janeiro, Rio de Janeiro, Brazil
[2]Mechatronic Engineering Program, Engineering Faculty, Universidad Militar Nueva Granada, Bogotá, Colombia
[3]Industrial Engineering Program, Engineering Faculty, Universidad Militar Nueva Granada, Bogotá, Colombia

Article Info

Article history:

Received Jul 31, 2026
Revised Sep 19, 2026
Accepted Oct 4, 2026

Keywords:

Automated forklift
Autonomous navigation
Obstacle avoidance
Potential fields
Smart logistics

ABSTRACT

In the industrial logistics sector, forklift automation is an element for advancing towards intelligent management that improves processes across complex environments. This article details the design of an automated forklift movement system that features autonomous navigation to a destination, recognizing and avoiding static and dynamic obstacles along the way. The load is detected using a YOLO network, which classifies and locates pallets to be moved with 100% accuracy and an inference time of 0.404 s. Navigation utilizes artificial potential fields to generate safe trajectories with attractive forces towards the destination and repulsive forces away from obstacles. To control the forklift's movement, a mathematical model based on the kinematics of two-dimensional four-wheeled robots was used, grounded in the Ackermann steering system, to adjust the angular velocity of the rear wheels and the steering angle of the front wheels. System validation was carried out through simulation using MATLAB software for trajectory calculation and simulation, and CoppeliaSim for graphical representation and execution of tasks by a forklift in a 3D environment. Results show that the system performed consistently across simulated scenarios, suggesting its potential as a preliminary framework for autonomous forklift navigation in smart logistics, with a 100% success rate and no accidents, an average path length of 23.82 m covered in 29.77 s, a final positioning error of 0.18 m, a minimum obstacle clearance distance of 0.88 m, and a computation time of 0.28 s. The proposed simulation framework can support development and evaluation of solutions for automated warehouses prior to their physical implementation.

This is an open access article under the CC BY-SA license.

Corresponding Author:

Anny Astrid Espitia Cubillos
Industrial Engineering Program, Engineering Faculty, Universidad Militar Nueva Granada
Carrera 11 #101-80, Bogotá, Colombia
Email: anny.espitia@unimilitar.edu.co

1. INTRODUCTION

In an industrial logistics environment, the problem to be solved is the need to transport, efficiently and without accidents, boxes of products, arranged on wooden pallets, from the order picking area to the dispatch area; in this context, forklift automation is a key element if one wishes to move towards intelligent handling in complex environments with both static and dynamic obstacles. Key elements in industrial robot design [1] involve power considerations [2] that need to be integrated into navigation systems [3]. Conventional logistics systems require constant human supervision, resulting in higher costs and a greater likelihood of errors; this drives the objective of achieving autonomous operation within a simulated warehouse that closely mirrors real-world conditions.

To improve logistics processes, there are currently autonomy developments that offer a scenario where automation of forklifts for their fully autonomous and safe operation is feasible. Autonomous vehicles are under development [4], with those operating in production environments [5] or near humans [6] having a distinct development stage [7]. Operation of forklifts in industrial settings is a focus of research within this vehicle [8] for their autonomous operation [9]. This research has employed algorithms for forklift control [10], computer vision systems [11], low-cost electronic systems [12], and battery design for propulsion [13]. In general, forklift automation involves network control methods [14], remote control via cellular devices, such as one described in [15], and artificial intelligence systems for its operation [16].

Before physical implementation, simulation of forklift models [17] allows for validation in different aspects in a virtual environment [18], employing techniques such as deep learning algorithms [19] or potential fields [20], [21] for navigation. It is possible to integrate intelligent control algorithms with autonomous navigation [22] to generate optimal trajectories [23], typically involving the transport of pallets containing multiple stacked boxes, as seen in [24] and [25], and to validate the algorithms through computer simulation [26].

Traditional routing methods are difficult to integrate into dynamic work environments that also require effective load recognition, environment reconstruction from images, the identification of obstacles and destinations, processes that lead to the generation of efficient, safe trajectories, all virtually validated prior to implementation in a physical environment. Identified innovations align with the objectives of smart logistics and Industry 4.0, where autonomy plays a significant role by contributing to increased efficiency in goods handling, a reduction in likelihood of human error, and a minimization of need for constant supervision in warehouses.

2. METHOD

Implementation of advanced navigation and control techniques enables safe movement of a wooden pallet loaded with stacked boxes from the order-picking area to the dispatch area of a warehouse. The first algorithm identifies and avoids obstacles. Path calculation and simulation were performed using MATLAB® R2025a software. Forklift's movements are simulated in a 3D environment using CoppeliaSim V4.10., allowing for a comparison of performance across different scenarios. Both software applications are synchronized using an API via the script ‘simRemoteApi.start(19999)’ and ‘clientID=sim.simxStart('127.0.0.1', 19999, true, true, 5000, 5)’, which establishes a connection to CoppeliaSim remote server at IP address ‘127.0.0.1’ (localhost) and port ‘19999’. If the connection is successful, ‘clientID’ will be greater than -1; however, any existing connection to CoppeliaSim must be closed beforehand to ensure there are no previously open connections.

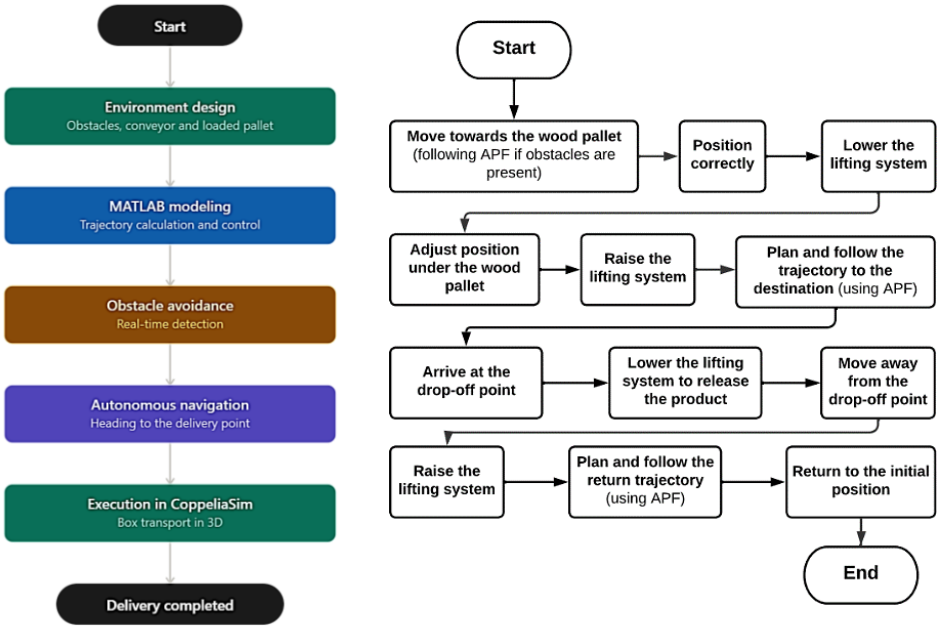
In CoppeliaSim, the process begins by defining the specifications of the forklift. A mast height of 2.5 m, a maximum lift of 1.0 m, a length of 2.5 m, a width of 1 m, a turning radius of 1.5 m, 1.0 m forks, and a load capacity of 3000 kg were selected. To execute its movement, a mathematical model based on the kinematics of four-wheeled robots with bidirectional wheels must be developed. This model determines the angular velocity of rear wheels using the relationship between linear velocity and wheel radius. Next, the turning radius equation is used to calculate the steering angle required to follow the desired trajectory. Finally, the algorithm adjusts control variables to ensure smooth and precise vehicle movement. To do this, a routine is implemented that combines inverse kinematics with the desired trajectory vector and finally transcribes it into the speeds and direction of the wheels.

As a second step, the design and model of boxes, as well as a wooden pallet, are created. The dimensions of the wooden pallet are 1200×1000 mm and a total height of 144 mm, with a dynamic load capacity of 1500 kg. Pallet detection is performed using deep learning algorithms employing transfer learning. In this case, a Yolo version 4 network [27] was used due to its high performance and low computational cost.

Thirdly, a 3D simulation environment is designed in CoppeliaSim to simulate a realistic movement scenario. The size of the environment is determined to be a square of 30 meters on each side, with space for the movement of a wooden pallet with boxes by forklift; the warehouse also includes shelves, a conveyor belt, and fixed obstacles such as tables and cubes. A dataset of 240 images for pallet detection was created in CoppeliaSim and labeled in MATLAB; images were split into 70% for training, 15% for validation, and 15% for testing. The wooden pallet to be transported is designed in a virtual environment with dimensions of 1200×1000 mm and a total height of 144 mm, simulating a dynamic load capacity of 1500 kg.

As the fourth step, the same 2D scenario is established in MATLAB software, using a video camera with a resolution of 1024×1024 pixels, located at the top of the cellar to capture a bird's-eye view. An API is used to connect CoppeliaSim and MATLAB, and the camera image is shared in real time for a dynamic scan. As pre-processing, MATLAB converts the image to HSV format, eliminating irrelevant areas and focusing the analysis on the region of interest, thus enabling the identification of obstacles and open spaces. To convert the image to binary format, a filter based on image intensity is used, with a threshold of 0.8471 and a tolerance of 0.1.

In the fifth step, to generate unobstructed trajectories, the artificial potential field (APF) [28] is used, a strategy that calculates an attractive force towards the target, repulsive forces against obstacles, and the total direction of the field (α). Finally, previously defined algorithms are integrated, and the algorithm is validated using a routine to transport boxes of products in a collision-avoidance environment within the workspace, ensuring precise positioning at each stage of the process. Thirty tests were conducted by varying the start and end points and obstacle locations within virtual scenarios, enabling calculation of performance metrics such as success rate, collision rate, arrival point accuracy, and efficiency. Figure 1(a) summarizes the methodological process, and Figure 1(b) shows the flowchart of the process of transporting an object with a forklift.

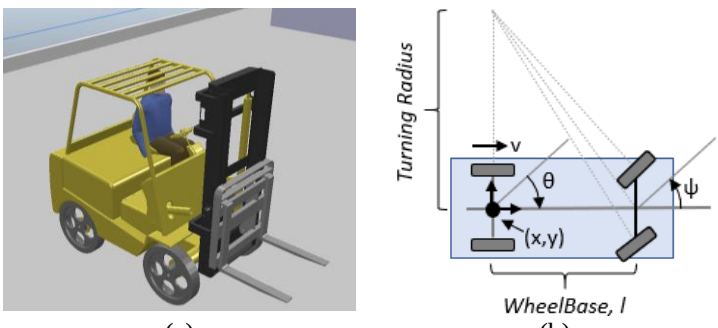


(a) (b)

Figure 1. Methodology flowchart: (a) general flow and (b) navigation flow for pallet transport

3. RESULTS AND DISCUSSION

Figure 2(a) presents a detailed model of forklift in CoppeliaSim, showing all main components, such as chassis, forks, wheels and lifting system, among others, that make up the structure of the vehicle and its operational capacity. The mathematical model based on kinematics of four-wheeled robots with bidirectional wheels takes as its reference Ackermann steering system [29], in which only front wheels rotate, while rear wheels remain fixed. This system generates efficient turning by orienting the front wheels, and axes of front and rear wheels interact to define turning trajectory. Figure 2(b) presents the kinematic diagrams of the four-wheeled robot with bidirectional wheels, illustrating how these components work together.



(a) (b)

Figure 2. Forklift model: (a) model in CoppeliaSim and (b) four-wheeled kinematic scheme

Kinematic model of a mobile robot with Ackermann steering requires determining the evolution of the robot's position and orientation, which depends on the angular velocity of the rear wheels and the steering angle of the front wheels. Variables include the robot's linear velocity v, radius of rear wheels r_r, angular velocity of rear wheels ω_r, and the robot's turning radius R. Additionally, the distance between the robot's axes L (between rear and front wheels), and steering angle of front wheels δ_f must be considered. Other relevant variables are the robot's angular velocity ω, the x and y coordinates that define its position in the plane, and orientation angle θ. Furthermore, derivatives of the positions $\dot{x}$ and $\dot{y}$, representing velocities along the x and y axes respectively, are incorporated, as well as the derivative of orientation $\dot{\theta}$, which corresponds to the robot's angular velocity.

Movement of the forklift is driven by two rear wheels, which maintain the same direction. Speed of forklift v depends on the angular velocity of the rear wheels ω_r and wheel radius r, through the relationship expressed in (1), which shows that the robot's linear speed is directly proportional to wheel radius and angular speed of the wheels. Thus, an increase in angular speed of rear wheels causes an increase in the robot's forward speed. Turning occurs around a point known as the instantaneous center of rotation (ICR). The distance between this center and the vehicle's axis is called the turning radius R, which depends on wheelbase L and steering angle δ_f of front wheels, calculated using (2). Conversely, as δ_f increases, turning radius decreases, allowing for tighter turns. When the vehicle follows a curved path, that is, when $\delta_f \neq 0$, the robot's angular velocity ω around the instantaneous center of rotation (ICR) is obtained as the ratio of the linear velocity v to the turning radius R. Angular velocity ω can be expressed in (3). This expression indicates that the robot's angular velocity depends directly on forward speed and direction angle δ_f.

$$v = r \cdot \omega_r \tag{1}$$

$$R = \frac{L}{\tan \delta_f} \tag{2}$$

$$\omega = \frac{v}{R} = \frac{r \cdot \omega_r}{L} \tan \delta_f \tag{3}$$

Change in robot's position in global (x, y) plane is determined by the robot's linear velocity v and orientation θ. Evolution of the x and y coordinates is described by (4) and (5), which indicate that displacement along the x and y axes depends on the robot's orientation. For example, if $\theta = 0$, movement occurs only along the x-axis, while $\theta = 90°$, the movement occurs only along the y-axis.

$$\dot{x} = v \cos \theta \tag{4}$$

$$\dot{y} = v \sin \theta \tag{5}$$

On the other hand, the evolution of the robot's orientation θ is determined by angular velocity ω. The (6) describes how the change in the robot's orientation is proportional to the angular velocity of the rear wheels and the steering angle of the front wheels. Inverse kinematics analysis seeks to find the angular velocity of the rear wheels (ω_r) and steering angle of front wheels (δ_f) from a desired linear velocity (v) and an angular velocity (ω) of the robot. Given a set of desired velocities in the global ($\dot{x}$,$\dot{y}$) plane, the linear velocity of the robot can be calculated using (7). The direction angle required to achieve the desired angular velocity can be calculated, which is expressed in (8). Using these equations, the relationship between the direction of the wheels and their angular velocity was established in relation to the linear and angular velocities of the robot.

$$\dot{\theta} = \frac{r \cdot \omega_r}{L} \tan \delta_f \tag{6}$$

$$v = \sqrt{\dot{x}^2 + \dot{y}^2} \tag{7}$$

$$\delta_f = \tan^{-1}\left(\frac{L\omega}{v}\right) \tag{8}$$

Implementing routine combines inverse kinematics with desired trajectory vector, which is translated into wheel speeds and directions. The algorithm uses points (x, y) and calculates the velocities required for the forklift to follow the trajectory. In each iteration, it determines linear velocity v based on movement between consecutive points and the forklift's orientation using a tangential function. Then, it estimates angular velocity ω according to changes in orientation between steps. Ackermann inverse kinematics algorithm is used to

establish ω_r and δ_f. Finally, the results are stored and returned as output, allowing vehicle control based on the desired trajectory.

For validation, a routine is implemented for transporting boxes of pre-sorted and organized products, ensuring their movement in an environment that prevents collisions. Detection of the stack is carried out by training a YOLO v4 network [27] with a camera located on the forklift front; Figure 3 illustrates a sample of the training database taken in the CoppeliaSim environment, edited, and labeled later in MATLAB [30]. The YOLOv4 network was trained to detect a single class corresponding to the pallet to be transported; Figure 3 illustrates the recall training curve. The dataset was generated using the CoppeliaSim environment and labeled in MATLAB, including variations in angle, distance, scale, lighting, background, and pallet position, to increase the diversity of the simulated images. Table 1 summarizes the main training and evaluation parameters.

Under these controlled simulated conditions, the network achieved a detection accuracy of 100% and an average inference time of 0.404 s. These values are contrasted with an average accuracy of 96.8% using a Yolo v5n network and 99.4% with a Yolo v5x network for automatic inspection of pallet racks reported in [31], and highest value achieved in [32] of 93.64% using the CAPI model for automatic recognition of pallet loads where they also relate average classification times of 0.315 seconds. The results demonstrate reliable identification of pallets in a simulated environment; however, further validation using real images is required to establish proof of generalization in warehouses. Figure 4 shows the CoppeliaSim environment, where integration of different components and obstacles that are part of the simulation is displayed.

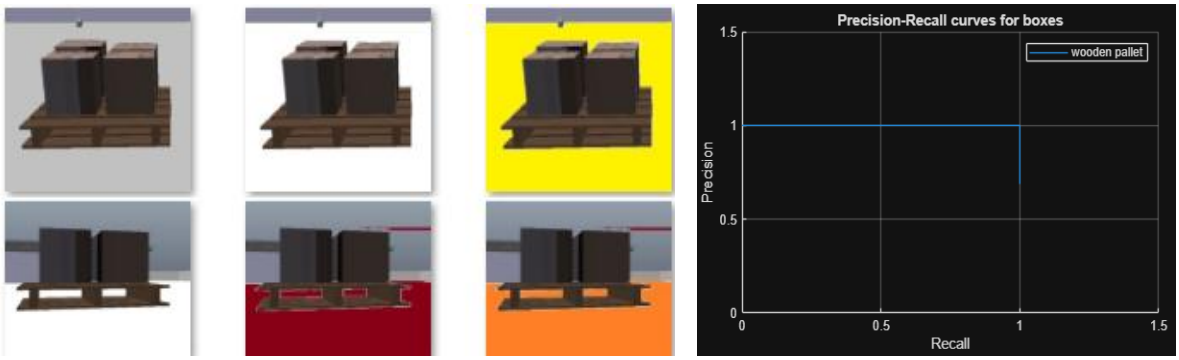


Figure 3. YOLO training database example and recall training curve

Table 1. YOLOv4 training and evaluation parameters

Element	Description	Parameter	Value
Number of classes	1 — pallet	Input	416×416×3
Learning rate	0.001	Optimizer	Adam
Learning parameters	6.9 M	Number of epochs	80
Dataset source	CoppeliaSim simulated images	Average classification time	0.404 s
Labeling tool	MATLAB	Detection accuracy	100%

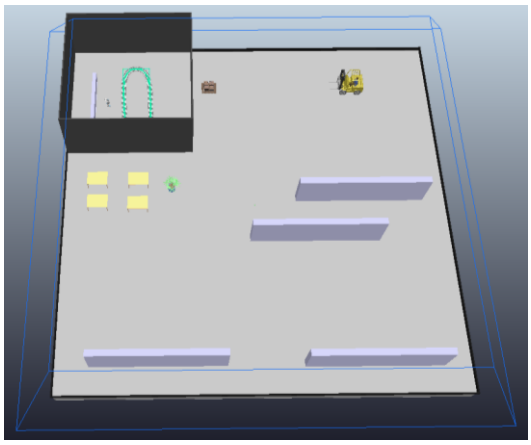


Figure 4. Simulation environment in CoppeliaSim

It is essential to develop this environment in MATLAB. Forklift's trajectory requires recognizing environment in MATLAB using video camera, image is converted to HSV format, as seen in Figure 5(a). Subsequently, irrelevant areas are removed to focus analysis on the region of interest. Next, filtering based on image intensity is applied, with a threshold of 0.8471 and a tolerance of 0.1, allowing image to be converted to binary format. Furthermore, image is resized to 100×100 pixels. The result is shown in Figure 5(b), where yellow area represents traversable region, and blue areas correspond to obstacles, objects, and key elements such as the forklift and wooden pallet. Forklift and wooden pallet are also detected in the binary image. Since dimensions of space are known, a grid function is used to assign each pixel its corresponding position in real space. Pixels corresponding to forklift and wooden pallet are identified and labeled. Result of this process is shown in Figure 5(c), where desired initial and final positions on trajectory are highlighted with circles, considering actual location of forklift and obstacles. Algorithm receives image sent from Coppelia along with positions of forklift and wooden pallet, transforming it into an adjusted binary image through a resizing process.

Having established the workspace and the positions of objects, the next objective is to define a trajectory generation strategy that avoids obstacles. The chosen technique for trajectory calculation is the APF. This navigation method in mobile robotics models the robot's environment with two types of forces: the attractive force toward the target, which generates a force field that guides the robot toward the target point, and the repulsive force from obstacles, which creates a force field that moves the robot away from obstacles to avoid collisions [25]. The attractive force is generated toward the goal and is calculated using a potential function that depends on the distance between the robot and the target. This attractive field guides the robot along the desired trajectory. Figure 6 shows an image representing the forces (attraction and repulsion) and their effect on the trajectory of the forklift to achieve a safe movement.

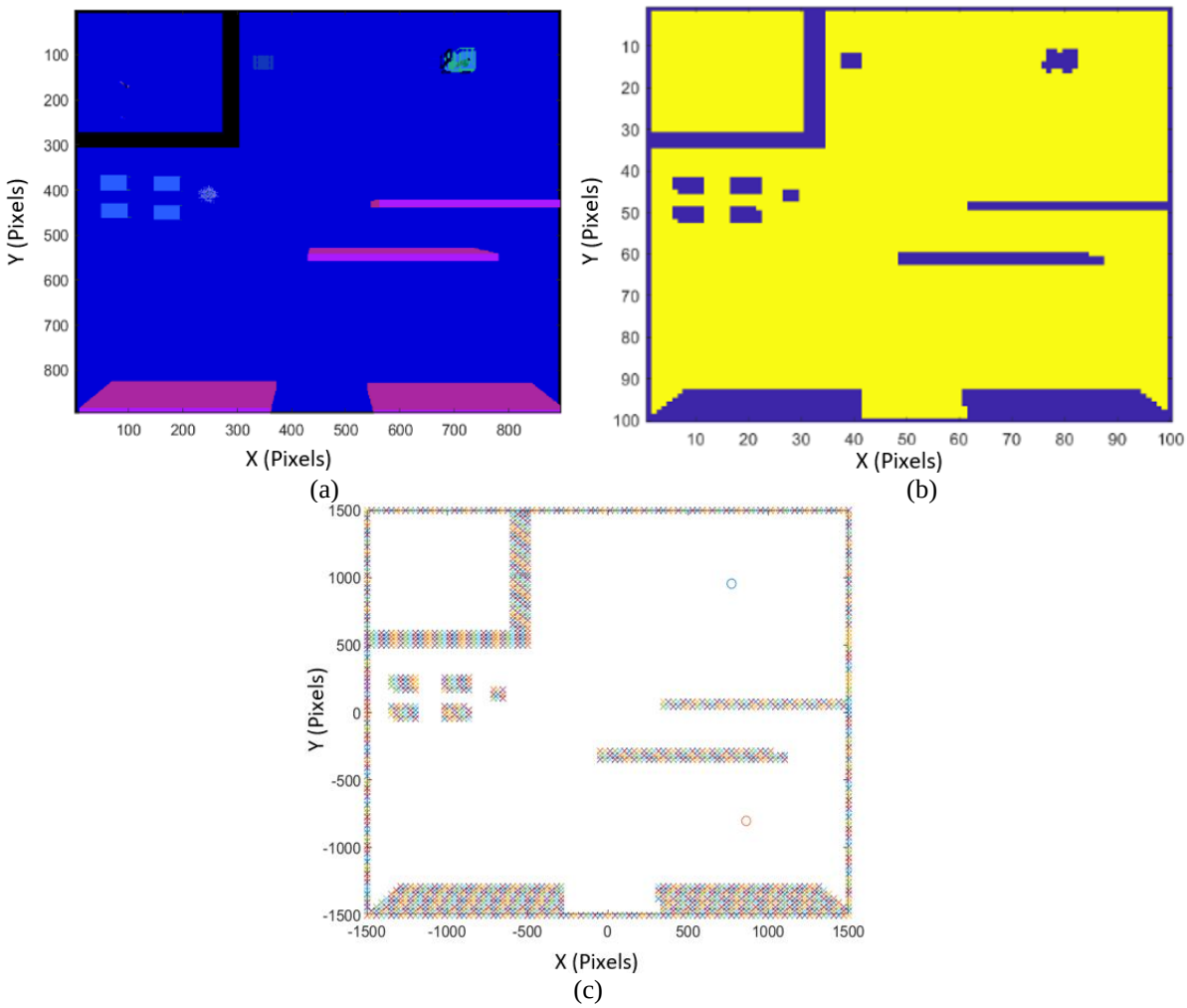


Figure 5. Image processing for environmental recognition in MATLAB: (a) conversion to HSV format, (b) binarized image with walkable areas and obstacles, and (c) position of the forklift and the pallet

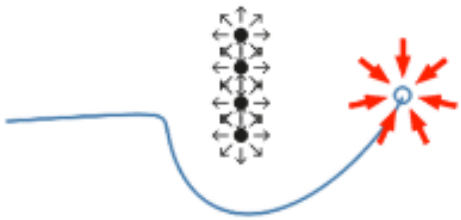


Figure 6. Representation of the forces of attraction and repulsion on the robot's trajectory

Attractive force towards goal is calculated from (9), where constant K_G defines the intensity of attraction and d_G distance from robot to target. On the other hand, repulsive force generated by obstacles is calculated with (10), where K_O is repulsion intensity and d_O is distance to obstacle, with an influence threshold $d_{O_{\max}}$. Finally, the resultant force on the robot, expressed in (11), is the sum of the attractive and repulsive forces, and the direction of movement is determined by angle α, calculated using the relationship between the components of the resultant force as in (12). The resultant force on the robot is the sum of attractive and repulsive forces generated by obstacles, resulting in the robot moving toward its target while avoiding collisions. Using d_O^3 in the denominator instead of Khatib's classic formulation causes the repulsion to decay more rapidly with distance, concentrating its effect in the immediate vicinity of the obstacle, relevant to subsequent sensitivity analysis.

$$F_a = -K_G \cdot \frac{(x-x_G, y-y_G)}{d_G} \tag{9}$$

$$F_r = -K_O \cdot \frac{(x-x_O, y-y_O)}{d_O^3} \tag{10}$$

$$F_x = F_{ax} + \sum_i F_{rxi}\,, F_y = F_{ay} + \sum_i F_{ryi} \tag{11}$$

$$\alpha = tan^{-1}\frac{F_y}{F_x} \tag{12}$$

Based on these equations, code was designed to implement the APF method for guiding a vehicle toward a target while avoiding obstacles. First, the attractive force toward the target is calculated, which is proportional to the distance r_G and directed toward the desired target. Then, the repulsive forces generated by obstacles are determined; these decrease with the cube of the distance $(1/r^3)$, meaning the effect is stronger when the robot is close to them. The equations governing movement are continuously updated using the kinematic model to ensure uninterrupted navigation to the destination and to avoid collisions. The calculation of the forklift's trajectory factors in the starting point, the destination, and the location of obstacles to generate a smooth path. Tests in the simulated environment evaluate trajectory generation based on four starting points and a single destination; the trajectories are represented in different colors in Figure 7.

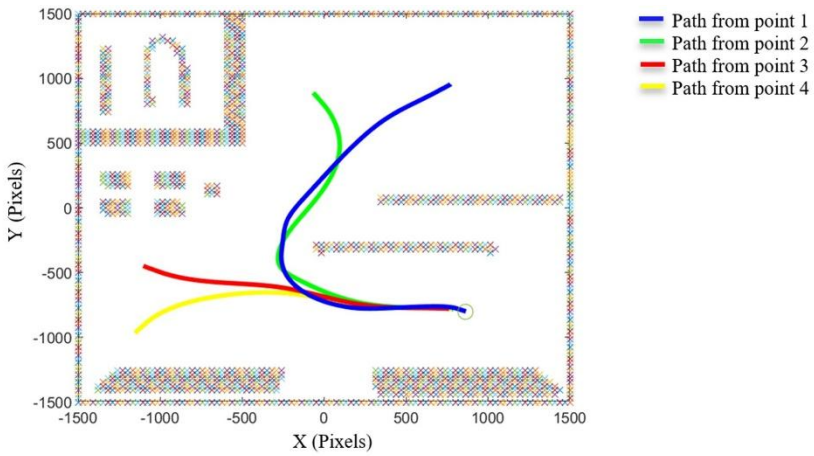


Figure 7. Trajectories generated by the navigation algorithm in MATLAB

Table 2 presents results of 30 runs of navigation algorithm (varying start point, destination, and obstacle configuration) to evaluate efficiency, positioning accuracy, obstacle avoidance, computational cost, and task completion of each simulated trajectory with a success rate of 100% and a collision rate of 0%. Results show that APF-based planner successfully performs navigation in evaluated scenarios. Both length and duration of trajectories are consistent across different scenarios, and positioning error confirms convergence toward target. APF method features a simple, computationally efficient formula that allows for direct integration into simulated environment, unlike graph-based (such as A* and Dijkstra), sampling-based (such as RRT and RRT*), or learning-based planners. However, APF method can be sensitive to variations in attraction and repulsion parameters and may generate trajectories corresponding to local (rather than global) minima in environments more complex than those evaluated.

Selection of these values in current implementation is not analytically derived K_G, K_O or d_{Omax} from dimensions of the environment (30×30 m) or from kinematic characteristics of forklift (turning radius of 1.5 m, total length of 2.5 m); values were iteratively adjusted until satisfactory trajectories were obtained in test set shown in Figure 8. Figure 8(a) shows how forklift approaches wooden pallet. Then, forklift interacts with wooden pallet and boxes, transporting them to their destination (Figures 8(b)-8(e)). Figure 8(f) shows forklift placing wooden pallet and products at the designated location.

Sensitivity analysis is proposed by sweeping one parameter at a time (one-factor-at-a-time), keeping others at their base values $K_G = 1$, $K_O = 1500$, $d_{Omax} = 200$, $K = 2$, $dt = 1$ show in Table 3, and recording the effect on metrics for set of initial positions evaluated. Expected behavior, based on equations, at destination, the actions are repeated in reverse order to deposit the load and return to initial position. While the simulation functions in a static, controlled environment, replicating it in a real-world setting requires accounting for lighting variations, the maximum object-discrimination distance, and environmental complexity factors that will determine the forklift's travel speed and maneuverability.

Table 2. Navigation performance of the APF-based trajectory planner

Metric	Unit	Mean	Standard deviation	Minimum	Maximum
Path length	(m)	23.82	4.46	19.85	29.52
Travel time	(s)	29.77	5.58	24.81	36.89
Final positioning error	(m)	0.18	0.06	0.09	0.27
Minimum obstacle clearance	(m)	0.88	0.18	0.62	1.12
Computation time	(s)	0.28	0.07	0.18	0.39

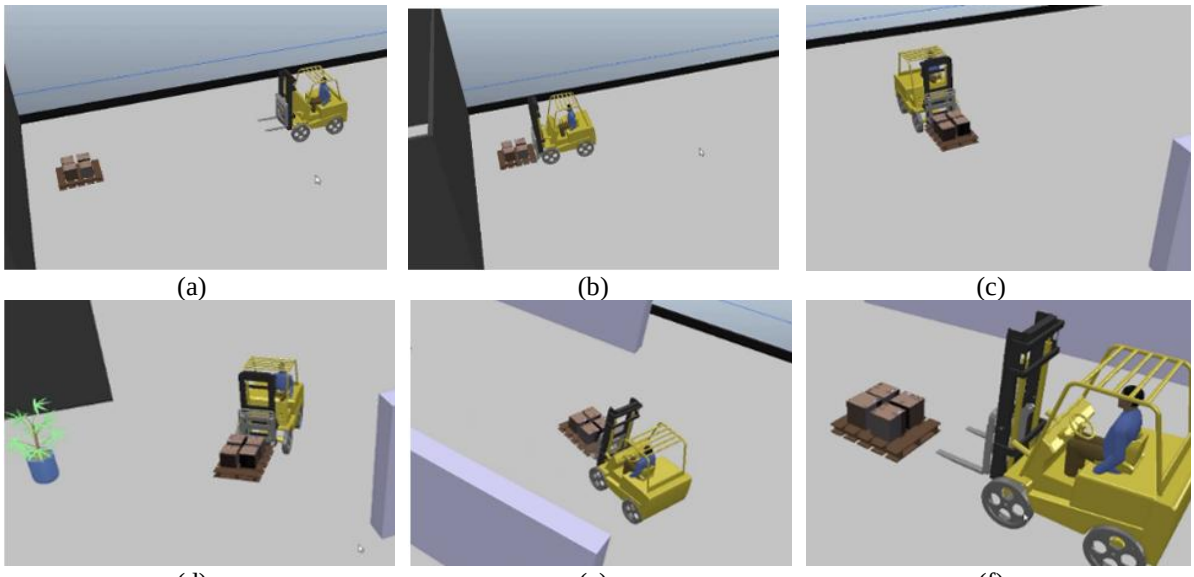


Figure 8. Image capture of movement sequence during simulation in CoppeliaSim: (a) forklift heading toward the pallet, (b) forklift picking up the pallet, (c) forklift handling the pallet, (d) forklift in motion, (e) forklift arriving at the destination, and (f) forklift placing the pallet in the designated location

Table 3. APF method configuration parameters

Parameter	Symbol	Value used	Function within the algorithm
Coefficient of attraction	K_G	1	The force scales towards the goal; its effect is linear in d_G
Repulsion coefficient	K_O	1500	Scale the evasion force; it must be considerably greater than K_G to compensate for cubic decay with d_O
Obstacle influence distance	d_{Omax}	200 px	Radius beyond which an obstacle ceases to generate repulsive force
Steering gain	K	2	Determine the convergence speed of the vehicle's orientation towards α
Threshold of reaching the finish line	—	$d_G < 5$	Trajectory stopping criterion

4. CONCLUSION

The results obtained allow us to affirm that the proposed intelligent navigation algorithms enable an autonomous forklift to reach the designated destination in the evaluated simulated scenarios while recognizing and avoiding obstacles within the virtual warehouse environment. Validation via simulation (using CoppeliaSim and MATLAB) demonstrated the correct operation of the algorithms for pallet detection, obstacle identification, and trajectory generation and control, supporting the use of the designed system as an initial proof-of-concept framework for autonomous forklift navigation. The main contribution of this work was the successful integration of pallet recognition at different angles and lighting conditions (using a YOLOv4 network), environment reconstruction from images, obstacle identification, APF-based trajectory generation, Ackermann-style kinematic control, and computer simulation. This framework can help with preliminary design and assessment of automated warehousing solutions. Results demonstrate that the algorithm generates feasible trajectories while avoiding obstacles, with a 100% success rate and a positioning error of 0.18 meters. Simulation allows for projecting the performance of solutions prior to physical implementation, thereby contributing to improved efficiency and safety within the framework of smart logistics and Industry 4.0, and narrowing the gap between the need for constant human oversight and autonomy. Future study should include physical environment validation, tests using real sensor data, and quantitative comparisons with alternative navigation systems to measure robustness, scalability, and performance in real-world scenarios. This could support the design of smart logistics systems for application in today's industry and integration of dynamic obstacle avoidance.

ACKNOWLEDGMENTS

The authors would like to thank the Universidad Militar Nueva Granada, where they are full-time associate professors.

FUNDING INFORMATION

Product derived from the research project titled “Fortalecimiento de procesos de recepción de pedidos y control de inventario de materias primas soportado en industria 4.0” INV-ING-4150, financed by the vice-rector for research of the Universidad Militar Nueva Granada, year 2025.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Juan C. Guachetá-Alba			✓		✓		✓	✓	✓					
Robinson Jiménez-Moreno	✓	✓		✓	✓	✓			✓					
Anny Astrid Espitia-Cubillos	✓			✓						✓	✓	✓	✓	✓

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

DATA AVAILABILITY

Data available upon request via email to the corresponding author.

REFERENCES

[1] H. M. A. Farid, S. Dabic-Miletic, T. Jameel, V. Simic, M. Riaz, and D. Pamucar, "Promoting sustainable logistics in the electronics industry: Circular intuitionistic fuzzy framework for evaluating smart robotics technologies," *Expert Systems with Applications*, vol. 287, Aug. 2025, doi: 10.1016/j.eswa.2025.128031.

[2] I. A. Ieropoulos and D. Z. Moreno, "Microbial fuel cells powering robots and beyond; historical perspective and future directions," *Bioelectrochemistry*, vol. 166, Dec. 2025, doi: 10.1016/j.bioelechem.2025.109010.

[3] E. Reyes-Reyes *et al.*, "Robust multilevel average controller for trajectory tracking in mobile robots powered by solar panels: Integrating actuators and power electronics stage," *Results in Engineering*, vol. 26, Jun. 2025, doi: 10.1016/j.rineng.2025.105311.

[4] S. Arunkumar, M. Ragul, M. Jayaprakash, and M. Boushica, "Autonomous equipment handling vehicle for warehouse," in *2023 2nd International Conference on Advancements in Electrical, Electronics, Communication, Computing and Automation (ICAECA)*, IEEE, Jun. 2023, pp. 1–5, doi: 10.1109/ICAECA56562.2023.10200290.

[5] S. Ferraro, L. Leoni, A. Cantini, and F. De Carlo, "Analyzing forklift and drone applications in sustainable logistics: a bibliometric review," *IFAC-PapersOnLine*, vol. 58, no. 19, pp. 463–468, 2024, doi: 10.1016/j.ifacol.2024.09.255.

[6] F. Saffre, H. Hildmann, E. Heikkila, T. Malm, and D. Pakkala, "Dynamic and probabilistic safety zones for autonomous mobile robots operating near humans," *Results in Engineering*, vol. 23, Sep. 2024, doi: 10.1016/j.rineng.2024.102731.

[7] G. Yang *et al.*, "Autonomous cart docking for transportation robots in complex and dynamic construction environments," *Automation in Construction*, vol. 171, Mar. 2025, doi: 10.1016/j.autcon.2025.105968.

[8] T. Aso, "Work element estimation for forklift operation," in *2022 International Power Electronics Conference (IPEC-Himeji 2022- ECCE Asia)*, IEEE, May 2022, pp. 1372–1377, doi: 10.23919/IPEC-Himeji2022-ECCE53331.2022.9807065.

[9] B. Allani, A. Ammamou, S. Kelouwani, M. A. Ouameur, G. Benarfa, and L. Zeghmi, "Machine learning approach for charging queue waiting time prediction of electrical autonomous forklifts fleet," in *2022 IEEE Vehicle Power and Propulsion Conference (VPPC)*, IEEE, Nov. 2022, pp. 1–6, doi: 10.1109/VPPC55846.2022.10003374.

[10] A. R. Hutauruk, A. S. Rohman, and P. H. Rusmin, "Rollover stability control of autonomous forklift using optimal control approach," in *2023 IEEE 13th International Conference on System Engineering and Technology (ICSET)*, IEEE, Oct. 2023, pp. 334–339, doi: 10.1109/ICSET59111.2023.10295070.

[11] G. Garibotto, S. Masciangelo, P. Bassino, C. Coelho, A. Pavan, and M. Marson, "Industrial exploitation of computer vision in logistic automation: autonomous control of an intelligent forklift truck," in *Proceedings. 1998 IEEE International Conference on Robotics and Automation (Cat. No.98CH36146)*, IEEE, pp. 1459–1464, doi: 10.1109/ROBOT.1998.677310.

[12] M. N. Tamara, A. Darmawan, N. Tamami, C. Sugianto, S. Kuswadi, and B. Pramujati, "Electronics system design for low cost AGV type forklift," in *2018 International Conference on Applied Science and Technology (iCAST)*, IEEE, Oct. 2018, pp. 464–469, doi: 10.1109/iCAST1.2018.8751559.

[13] Z. Tong and S. Guan, "Developing high-precision battery electric forklift driving cycle with variable cargo weight," *Transportation Research Part D: Transport and Environment*, vol. 136, Nov. 2024, doi: 10.1016/j.trd.2024.104443.

[14] A. Bhat, N. Kai, T. Suzuki, T. Shiroshima, and H. Yoshida, "An advanced autonomous forklift based on a networked control system," *IFAC-PapersOnLine*, vol. 56, no. 2, pp. 11444–11449, 2023, doi: 10.1016/j.ifacol.2023.10.432.

[15] H. Abuzied, N. Nazih, and A. Sahbel, "Design and simulation of eco-friendly smartphone controlled forklift," *Heliyon*, vol. 10, no. 9, May 2024, doi: 10.1016/j.heliyon.2024.e30682.

[16] G. Li, X. Wang, J. Yang, and B. Wang, "A new method to design fuzzy controller for unmanned autonomous forklift," in *2015 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, IEEE, Dec. 2015, pp. 946–951, doi: 10.1109/ROBIO.2015.7418893.

[17] M. Becerra-Fernandez, O. R. Romero, J. Trujillo-Diaz, and M. M. Herrera, "Assignment-simulation model for forklifts in a distribution center with aisle constraints," *Simulation Modelling Practice and Theory*, vol. 133, May 2024, doi: 10.1016/j.simpat.2024.102902.

[18] M. S. Islam, S. J. N. Zahabi, S. Kim, N. Lau, M. A. Nussbaum, and S. Lim, "Changes in forklift driving performance and postures among novices resulting from training using a high-fidelity virtual reality simulator: An exploratory study," *International Journal of Industrial Ergonomics*, vol. 104, Nov. 2024, doi: 10.1016/j.ergon.2024.103648.

[19] M. Mohammadpour *et al.*, "Energy-efficient motion planning of an autonomous forklift using deep neural networks and kinetic model," *Expert Systems with Applications*, vol. 237, Mar. 2024, doi: 10.1016/j.eswa.2023.121623.

[20] G. Langxiong, X. Li, T. Yan, L. Song, J. Xiao, and Y. Shu, "Intelligent ship path planning based on improved artificial potential field in narrow inland waterways," *Ocean Engineering*, vol. 317, Feb. 2025, doi: 10.1016/j.oceaneng.2024.119928.

[21] C. Cui, S. Wu, and X. Chen, "QuatAPF: Safety-aware and energy-efficient dynamic path planning and tracking control for deep-sea mining vehicles via a quatre-artificial potential field method," *Ocean Engineering*, vol. 321, Mar. 2025, doi: 10.1016/j.oceaneng.2024.120236.

[22] C.-J. Lin, B.-H. Chen, and J.-Y. Jhang, "IoT-based obstacle avoidance and navigation for UGVs in wooded environments using adaptive fuzzy artificial potential field," *Internet of Things*, vol. 30, Mar. 2025, doi: 10.1016/j.iot.2025.101524.

[23] T. A. Tamba, Q. T. T. Bui, and Keum-Shik Hong, "Trajectory generation of an unmanned forklift for autonomous operation in material handling system," in *2008 SICE Annual Conference*, IEEE, Aug. 2008, pp. 1800–1805, doi: 10.1109/SICE.2008.4654955.

[24] A. Keser and P. O. Ekim, "Otonom forklift ile palet tespiti ve hassas yanaşma pallet detection and docking with autonomous forklift," in *2024 Innovations in Intelligent Systems and Applications Conference (ASYU)*, IEEE, Oct. 2024, pp. 1–5, doi: 10.1109/ASYU62119.2024.10756973.

[25] I. P. Latini, W. E. Barioni, M. Teixeira, F. Neves, and L. V. R. de Arruda, "A CNN pallet detection model for an autonomous forklift," in *2023 Latin American Robotics Symposium (LARS), 2023 Brazilian Symposium on Robotics (SBR), and 2023 Workshop on Robotics in Education (WRE)*, IEEE, Oct. 2023, pp. 490–495, doi: 10.1109/LARS/SBR/WRE59448.2023.10333062.

[26] A. Bhat, N. Kai, T. Suzuki, T. Shiroshima, and H. Yoshida, "Fake obstacle generation-based pallet alignment for forklifts," in *2024 16th International Conference on Computer and Automation Engineering (ICCAE)*, IEEE, Mar. 2024, pp. 676–680, doi: 10.1109/ICCAE59995.2024.10569669.

[27] S. M, S. Mahapatra, and S. Jhansi, "Innovative object detection with YOLOv4 and adaptive bounding boxes," in *2025 4th International Conference on Sentiment Analysis and Deep Learning (ICSADL)*, IEEE, Feb. 2025, pp. 885–891, doi: 10.1109/ICSADL65848.2025.10933169.
[28] H. Li, W. Liu, C. Yang, W. Wang, T. Qie, and C. Xiang, "An optimization-based path planning approach for autonomous vehicles using the DynEFWA-artificial potential field," *IEEE Transactions on Intelligent Vehicles*, vol. 7, no. 2, pp. 263–272, Jun. 2022, doi: 10.1109/TIV.2021.3123341.
[29] M. Veneri and M. Massaro, "The effect of Ackermann steering on the performance of race cars," *Vehicle System Dynamics*, vol. 59, no. 6, pp. 907–927, Jun. 2021, doi: 10.1080/00423114.2020.1730917.
[30] The MathWorks, "Mobile robot kinematics," The MathWorks Inc. Accessed: Sep. 17, 2026. [Online]. Available: https://www.mathworks.com/help/robotics/ug/mobile-robot-kinematics-equations.html.
[31] M. Hussain, "YOLO-v5 variant selection algorithm coupled with representative augmentations for modelling production-based variance in automated lightweight pallet racking inspection," *Big Data and Cognitive Computing*, vol. 7, no. 2, p. 120, Jun. 2023, doi: 10.3390/bdcc7020120.
[32] J. J. L. Del Olmo, E. P. Ballesteros, P. E. López-De-Teruel, and A. Ruiz, "From few-shot to zero-shot pallet load recognition: a deployed embedding-based vision system for industrial logistics," *Proceedings - 2026 IEEE/CVF Winter Conference on Applications of Computer Vision, WACV 2026*, pp. 2901–2911, 2026, doi: 10.1109/WACV61042.2026.00284.

BIOGRAPHIES OF AUTHORS



Juan C. Guacheta-Alba received his Bachelor of Science degree in mechatronics engineering from Universidad Militar Nueva Granada, Colombia, in 2020 and his Master of Science degree in mechatronics engineering from the same university in 2022. He is currently pursuing a Doctor of Philosophy degree in mechanical engineering at the Federal University of Rio de Janeiro, Brazil. His research interests include robotics, multi-agent systems, and optimization. He can be contacted at email: juan.camilo@mecanica.coppe.ufrj.br.



Robinson Jiménez-Moreno received his degree in electronic engineering from Universidad Distrital Francisco José de Caldas, Colombia, in 2002. He received his Master of Science degree in engineering from the National University of Colombia in 2012 and his Doctor of Philosophy degree in engineering from Universidad Distrital Francisco José de Caldas in 2018. He is currently an associate professor at Universidad Militar Nueva Granada, Colombia. His research focuses on convolutional neural networks for object recognition and image processing in robotic applications, particularly human–machine interaction. He can be contacted at email: robinson.jimenez@unimilitar.edu.co.



Anny Astrid Espitia-Cubillos performed her undergraduate studies in Industrial Engineering in the Universidad Militar Nueva Granada in 2002 and M.Sc. in Industrial Engineering from the Universidad de Los Andes in 2006. She is an associate professor in the Industrial Engineering Program at Universidad Militar Nueva Granada, Bogotá, Colombia. She can be contacted at email: anny.espitia@unimilitar.edu.co.