[Iraqi Journal of Computers, Communications, Control and Systems Engineering](https://ijccce.researchcommons.org/journal)

Volume 26 | Issue 1 Article 2

4-4-2026

Enhancement Dijkstra Algorithm Approach for Path Planning Navigation

Ali Alyasin
Directorate of Najaf Education, Ministry of Education, Najaf, Iraq AND College of Electrical Engineerin University of Technology Baghdad, Iraq, eee.20.08@grad.uotechnology.edu.iq

Eyad I. Abbas
College of Electrical Engineerin University of Technology Baghdad, Iraq, eyad.i.abbas@uotechnolog.edu.iq

Jabbar K. Mohammed
College of Electrical Engineerin University of Technology Baghdad, Iraq,
jabbar.k.mohammed@uotechnology.edu.iq

Follow this and additional works at: https://ijccce.researchcommons.org/journal

How to Cite This Article

Alyasin, Ali; Abbas, Eyad I.; and Mohammed, Jabbar K. (2026) "Enhancement Dijkstra Algorithm Approach for Path Planning Navigation," *Iraqi Journal of Computers, Communications, Control and Systems Engineering*: Vol. 26: Iss. 1, Article 2.
Available at: https://ijccce.researchcommons.org/journal/vol26/iss1/2

This Article is brought to you for free and open access by Iraqi Journal of Computers, Communications, Control and Systems Engineering. It has been accepted for inclusion in Iraqi Journal of Computers, Communications, Control and Systems Engineering by an authorized editor of Iraqi Journal of Computers, Communications, Control and Systems Engineering.

IJCCCE 2026;26(1):25–34

Scan the QR to view
the full-text article on
the journal website

RESEARCH ARTICLE

Enhancement Dijkstra Algorithm Approach for Path Planning Navigation

Ali Alyasin [a,b,*], Eyad I. Abbas [b], Jabbar K. Mohammed [b]

[a] Directorate of Najaf Education, Ministry of Education, Najaf, Iraq
[b] College of Electrical Engineerin University of Technology Baghdad, Iraq

ABSTRACT

Mobile robots are developing rapidly compared for modern information technology and are used in many fields, such as medicine, industry, military services, and all public services. However, the challenges facing these technologies are significant, including positioning, environmental awareness, path planning, and motion control. Greedy algorithm, Dijkstra algorithm is used for make the path planning better process for ensure increased navigation efficiency. The algorithm uses graphs to find the shortest path between two nodes in a weighted graph by repeating the process. calculating how far things have gone. The initial process of calculating the shortest path speeds up the proposed algorithm's approach, as it travels between the starting point and each of the other nodes simultaneously, either along the same continuous path or across multiple paths until reaching the destination. Central nodes are the starting point of the algorithm, which uses data that is not affected by the paths followed. We employed Dijkstra's algorithm in an empirical study involving a service robot., where the robot successfully navigated three obstacles without colliding with them. The robot demonstrated high efficiency in identifying the shortest and quickest route, achieving an average error at a velocity of 0.26 meters per second, with an x-axis length of 0.034 meters and a y-axis length of 0.017 meters.

Keywords: Mobile robots, Path planning, Dijkstra algorithm, Rviz, LiDAR

Highlights

1. A service robot that plans a quick and secure route in dynamic environments using Dijkstra's algorithm.
2. A robot using four omni-directional wheels and an A3 type LiDAR sensor for obstacle detection.
3. A Robot Operating System (ROS) and an ESP32 microcontroller are utilized for motion control.

Received 5 September 2025; revised 27 November 2025; accepted 6 January 2026.
Available online 4 April 2026

* Corresponding author.
E-mail addresses: eee.20.08@grad.uotechnology.edu.iq (A. Alyasin), eyad.i.abbas@uotechnolog.edu.iq (E. I. Abbas), jabbar.k.mohammed@uotechnology.edu.iq (J. K. Mohammed).

https://doi.org/xx.xxxxx/2617-3352.1514

2617-3352/© 2026 IJCCCE, University of Technology-Iraq, Baghdad, Iraq. This is an open access article under the CC BY 4.0 Licence (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Robotics technology is rapidly evolving, and one of the most advanced areas is mobile robotics, which is widely used in industries such as healthcare, medicine, defense, manufacturing, and other public services [1, 2]. To avoid obstacles, robots must move with high energy and balanced efficiency in the physical environment [3, 4]. The fundamental goal of mobile robot development is to make robots that are safe, dependable, and high-performing. Even though these robots work very well, they come with several problems [5, 6]. These common challenges include environmental awareness, motion control, optimal path planning, and location tracking [7]. Among the most important and significant of these challenges, and the most important and balanced aspect, is path planning [8]. Complex and dynamic environments require safe and optimal navigation, and many highly accurate and efficient algorithms are available to plan a route of a mobile robot [9].

The Dijkstra algorithm operates on the greedy principle, assessing and prioritizing the most efficient and least costly paths by evaluating the distance metric and offering various options for path optimization [10, 11]. The algorithm's main strength lies in calculating the best and shortest paths based on the exact graph using non-negative, computed weights, making it one of the best and most reliable options in many everyday applications [12, 13]. One area where the algorithm gets better at finding the best path is in mobile robots. To make sure that navigation is safe, it must be picked to make sure that robots can move around on their own, which makes it one of the finest possibilities for robot movement chains to avoid hitting things in complicated surroundings [14].

2. Research method

2.1. Overview

The robot used in this research is a service robot equipped with four omnidirectional mechanical wheels. The robot's wheels have a unique design, consisting of a central axis around which a freely rotating circular cylinder rotates, mounted at a 45° angle relative to the horizontal perimeter of the wheel [15]. To move freely while avoiding obstacles, the robot uses an A3 LiDAR sensor [16], connected to a Jetson Nano [17], a mapping data processor with high obstacle detection efficiency. This robot operating system (ROS) works with an ESP32 controller, which is the robot's main controller [18].

2.2. Robot design

This is a high-quality study, as the robot's body was designed using two layers of flexible aluminum sheets. The first layer contained electronic components such as an ESP32 module, a drive motor, a voltage regulator, DC motors, and sensors, while the second layer housed the LiDAR and Jetson Nano devices. The chassis dimensions were 260 mm wide, 290 mm long, and 240 mm high. Fig. 1 shows the prototype of the service robot.

2.3. Path planning

Nuclear Path planning technology is used to calculate the best and shortest paths for mobile robots from the moment of departure until they reach their destination [19]. Path planning is the best way to find the shortest path for a mobile robot from one source. This system is the best at finding a given objective and choosing the optimum path [20].

However, the shortest path is not always considered the best or optimal example, as the best and shortest paths can be obtained using path planning, depending on specific

Fig. 1. Service robot model.

constraints [21]. Some additional factors, such as the required stops during movement and the number of turns, must be taken into account to ensure smooth operation [22]. In this study, to obtain the best path and due to its reliability in determining optimal paths, we will choose the Dijkstra algorithm [23].

The Dijkstra's algorithm is considered one of the best algorithms and is used and considered one of the best methods for finding and determining the shortest path from the starting point until reaching the desired destination [24]. Finding the shortest pathways between two points in weighted graphs is a common application in graph theory, and all weights must have non-negative edges [25, 26]. This algorithm is iterative, as its calculations gradually calculate the path traversing the fewest nodes (between nodes) in the graph starting at the starting point through to the goal. Its workflow can be visualized in Fig. 2.

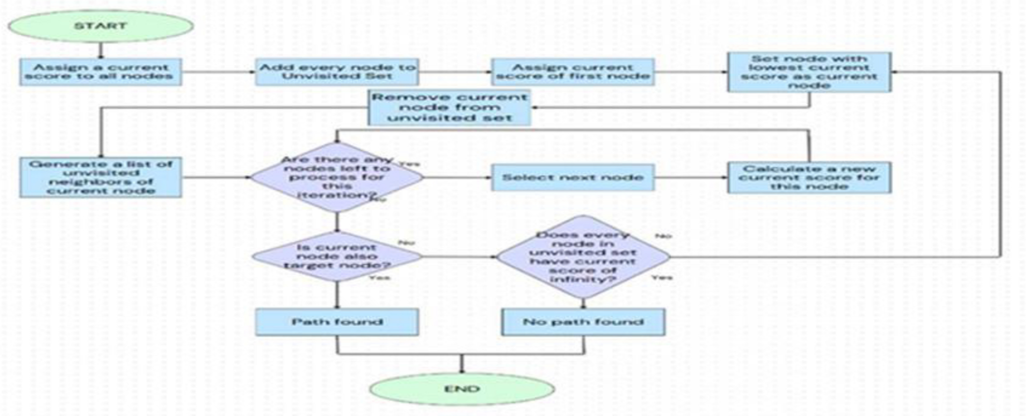
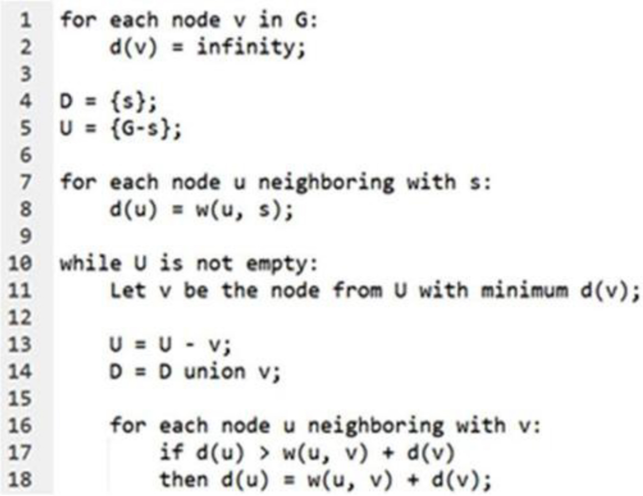


Fig. 2. Flowchart of dijkstra's algorithm.

While calculating the possible paths of the shortest length among the starting point and through all the nodes to the destination, proposed algorithm applies a balanced initial step [27]. In the best case, these paths diverge, while in the worst case, they converge on the common path. Although the best case may be less than 1, this approach may be applicable in specific cases [28]. However, one limitation of the algorithm is that it compares direct calculations from the central node in the graph based on path-independent information [29].



```
for each node v in G:
    d(v) = infinity;

D = {s};
U = {G-s};

for each node u neighboring with s:
    d(u) = w(u, s);

while U is not empty:
    Let v be the node from U with minimum d(v);

    U = U - v;
    D = D union v;

    for each node u neighboring with v:
        if d(u) > w(u, v) + d(v)
        then d(u) = w(u, v) + d(v);
```

Fig. 3. Pseudocode for dijkstra's algorithm.

In Fig. 3, We outline the pseudocode for a modified Dijkstra algorithm that was used to determine the shortest path in a network with non-negative weight between the starting node and every other node to the destination [30]. Below is a thorough description of every section of the code:

1. **Initial setup:** The distance (s) to the first node (n) in the graph (G) is D (s, n) for each node (v), and all nodes must be randomly designated as unreachable, as shown in Eq. (1).

   $$\forall v \in G : d(v) \tag{1}$$

2. **Initialization:** The starting point starts from the set (D) which contains only the source node (s) Eq. (2). The set (U) contains all the ports of other nodes except (s) Eq. (3).

   $$\mathrm{D} = \{\mathrm{S}\} \tag{2}$$

   $$\mathrm{U} = \mathrm{G} \setminus \{\mathrm{S}\} \tag{3}$$

3. **Neighbor distance initialization:** The initialization for every direct term u in the source s is d(u) = w(s,u) Eq. (4).

   $$\forall u \in adj\ \ (s) : d\ \ (u) = w(u, s) \tag{4}$$

4. **Shortest (Summary):** After extracting the closest node (v) from U, we designate it as a hosted node (D). If there is a shortest path, the distances between its neighbors need to be updated. Until every node has been processed, we repeat the procedure. Eq. (5) illustrates this.

   $$\forall u \in adj\ \ (v) : if\ d\ \ (u) > d(v) + w(v, u)\, then\, d(u) = d(v) + w(v, u) \tag{5}$$

In order to find the path of least length between the source and the edges of every node in the graph, Dijkstra's method visits each node just once and can automatically update distances using node edges.

3. Result and discussion

Dijkstra's algorithm was run three times to find the shortest path between points and count the number of places visited, with the same starting point (x = 0, y = 0) in each test. In each trial, the locations of the obstacles were varied to generate distinct setups simulating different robot movement scenarios. This varied distribution of barriers allowed us to evaluate the algorithm's effectiveness in a range of complex settings. It also tested the robot's ability to adapt to changing paths and choose the optimal route. Repeating the technique in different situations yielded a large dataset on movement patterns. The algorithm's efficiency in reducing distance and the number of points traversed was assessed through data analysis. This involved using a range of sample sizes to examine how the robot behaved in different simulated settings.

In the first test, the robot moved between point A and point B, without any obstacles in an unorganized manner. The objective of the robot was to triangulate the shortest path and the shortest time of travel while successfully navigating the obstacles. The resulting optimal path was visualized using Rviz. The test results are shown in Fig. 4.

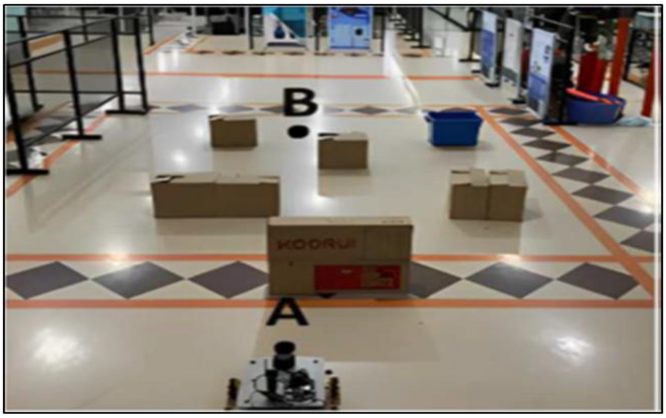


Fig. 4. First test realistic obstacle.

Test 1: The robot successfully avoided strategically placed obstacles while navigating between points A and B for a distance of 4.847 meters, completing the journey in 16.27 seconds. The robot's goal is to achieve a goal without colliding with any randomly placed obstacles, as shown in the tests in Figs. 4 and 5.

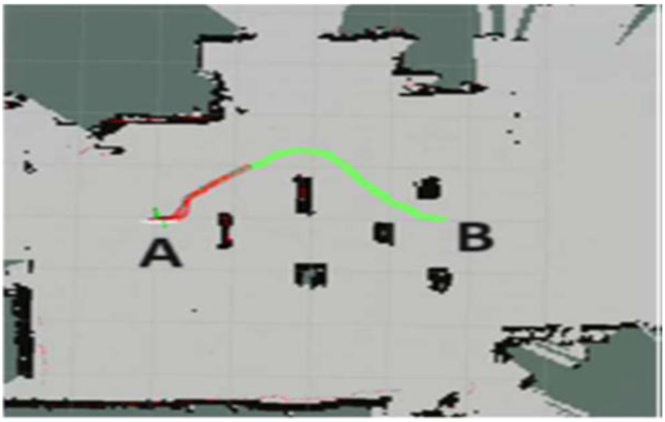


Fig. 5. Test 1 visualizing path planning in RVIZ.

Test 2: The robot follows a planned path between points A and B. This is the second scenario, where the robot moves smoothly through randomly distributed obstacles, arranged in the opposite order to the first test. The planned path can be captured using Rviz for analysis and monitoring as shown in the tests in Figs. 6 and 7.

Fig. 6. Test 2 realistic obstacle.

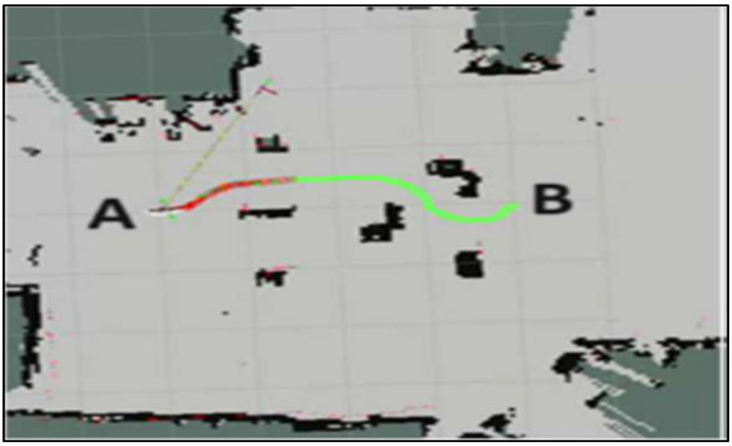


Fig. 7. Test 2 realistic obstacle.

Test 2: The robot successfully navigated between point A and point B, avoiding all obstacles. The total distance covered was 4.735 meters, and the arrival time was 21.82 seconds. The robot demonstrated smooth movement, avoiding and navigating near obstacles.

Test 3: The experiment demonstrated that the robot navigated between points (A) and (B) in a setting with randomly dispersed obstacles and varying locations, unlike in the prior test. RViz software was used for the path planning procedure, and Figs. 8 and 9 display the robot's initial movement results.

Test 3 Results: The robot successfully navigated between points A and B, covering a distance of 4.954 meters in 19.73 seconds. During this test, the robot successfully navigated and avoided previously identified obstacles without any collisions. Path planning allowed

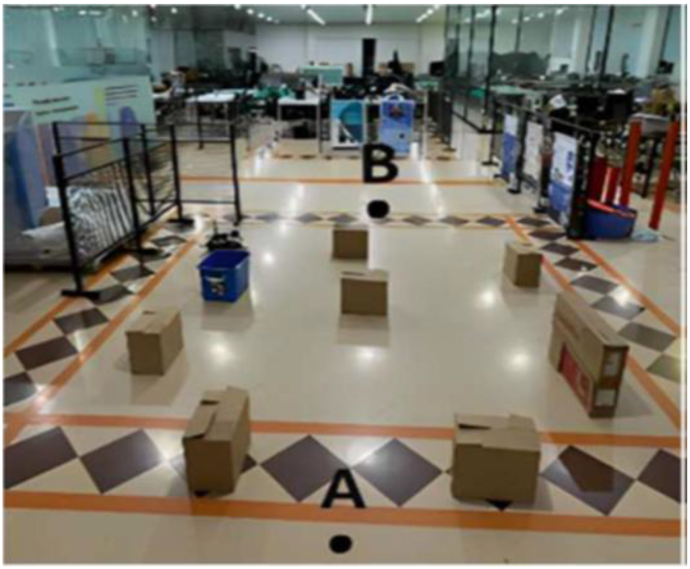


Fig. 8. Test 3 realistic obstacle.

Fig. 9. Test 3 visualizing path planning in RVIZ.

the robot to navigate and overcome obstacles with confidence and safety, ensuring that it did not touch them.

The calculated data included:

- Target location (intended destination)
- Robot location (final coordinates reached in the graph)
- Error margin (departure from the planned path)
- Total length of the route covered by the robot.
- Time to reach the target
- Actual speed of the robot (calculated based on distance and time)

Table 1. Test result.

Test number	Target position (m)		Actual position (m)		Margin of error (m)		Distance (m)	Time (s)	Speed (m/s)
	x	y	x	y	x	y			
1	4.136	0.116	3.889	0.132	0.123	0.128	4.875	16.69	0.36
2	4.123	0.124	3.897	0.124	0.127	0.105	4.856	21.68	0.32
3	4.246	0.132	4.223	0.128	0.125	0.127	4.786	19.85	0.19

Table 1 summarises these findings. The outcomes of these three tests are shown in the table, wherein the actual and target locations are recorded on x- and y-axis and the errors in location are calculated (the difference between the actual and target values). The average test also provides calculations of the distance traveled, speed, and total time. The robot traveled an average distance of 4.786 meters at a speed of approximately 0.125 meters per second. The recorded margin of error was 0.026 meters for the x axis and the 0.127 meters of the y-axis.

4. Conclusions

In this paper, we present a study using the modified Algorithms and path_planning of Dijkstra method for a service robot. This study includes three scenarios with different obstacle tests to evaluate the without collisions in each scenario. The robot demonstrated remarkable precision while moving at a nearly constant speed of 0.25 meters per second. During navigation, a slight positional variation was noted, with an average error of roughly 0.018 meters on the x-axis. The average inaccuracy on the y-axis was 0.032 m, suggesting strong orientation stability. These outcomes demonstrate that the system can operate accurately and smoothly in various operational settings. They also indicate how well the algorithm maintains the intended direction while minimising spatial errors. This demonstrates how effectively the control system enhances navigation accuracy and prevents deviations. All things considered, the studies showed a balanced performance that combined positional precision and speed. According to the calculations and experimental findings, it is obvious that the altered Dijkstra algorithm is much more reasonable enables the robot to be effective in determining the shortest, best, and most efficient paths while avoiding obstacles. This conclusion is supported by the three scenarios tested separately and with different obstacle graphs, all of which confirmed the algorithm's reliability in moving and navigating without collisions.

Acknowledgments

The authors would like to thank the anonymous reviewers for their valuable comments.

Conflict of interest statement

The authors declare no conflict of interest.

Data availability

The data that support the findings of this study are available on request from the corresponding author.

References

[1.] A. Newman, G. Yang, B. Wang, D. Arnold, and J. Saniie, “Embedded Mobile ROS Platform for SLAM Application with RGB-D Cameras,” in *2020 IEEE International Conference on Electro Information Technology (EIT)*, Chicago, IL, USA, pp. 449–453, 2020.

[2.] C. Molder, D. Toma, and A. Țigău, “Navigation Algorithms with LIDAR for Mobile Robots,” *Journal of Military Technology*, vol. 2, pp. 5–10, 2019.

[3.] R. Amirullah, A. Rusdina, and D. Darlis, “Implementasi Sistem Path Planning Dan Routing Untuk Mobile Robot Berbasis Visible Light Communication,” *e-Proceeding of Engineering*, vol. 8, pp. 4283–4291, 2021.

[4.] T.-W. Zhang, G.-H. Xu, X.-S. Zhan, and T. Han, “A New Hybrid Algorithm for Path Planning of Mobile Robot,” *The Journal of Supercomputing*, vol. 78, pp. 4158–4181, 2022.

[5.] H. Pratiwi, “Application of The Dijkstra Algorithm To Determine The Shortest Route From City Center Surabaya To Historical Places,” *JTEKSIS*, vol. 4, 2022.

[6.] R. S. Wijaya, A. Mahendra, S. Prayoga, and A. Wibisana, “Path Planning Application using Dijkstra Algorithm on Service Robot,” in *Proceedings of the 7th International Conference on Applied Engineering (ICAE2024)*, Atlantis Press, pp. 262–271, 2024.

[7.] M. J. Mohamed and M. K. Hamza, “Design PID Neural Network Controller for Trajectory Tracking of Differential Drive Mobile Robot Based on PSO,” *Engineering and Technology Journal*, vol. 37, pp. 574–583, 2019.

[8.] R. Al-Hakim, H. Satria, Y. Arief, A. Pangestu, A. Jaenul, R. Hertin, and D. Nugraha, “Aplikasi Algoritma Dijkstra dalam Penyelesaian Berbagai Masalah,” *Jurnal Manajemen Sistem Informasi Dan Teknologi*, vol. 11, pp. 42–47, 2021.

[9.] J. Díaz-Téllez, R. S. García-Ramírez, J. Pérez-Pérez, J. Estevez-Carreón, and M. A. Carreón-Rosales, “ROS-based Controller for a Two-Wheeled Self-Balancing Robot,” *Journal of Robotics and Control (JRC)*, vol. 4, pp. 491–499, 2023.

[10.] A. Ubaidillah and H. Sukri, “Application of Odometry and Dijkstra Algorithm as Navigation and Shortest Path Determination System of Warehouse Mobile Robot,” *Journal of Robotics and Control (JRC)*, vol. 4, pp. 413–423, 2023.

[11.] P. Raja and S. Pugazhenthi, “Optimal path planning of mobile robots: A review,” *International Journal of Physical Sciences*, vol. 7, 2012.

[12.] L. K. Majeed, “Formation Control of Mobile Robots Using Sliding Mode Control Based on Conditional Servocompensator,” *Engineering and Technology Journal*, vol. 36, pp. 1302–1310, 2018.

[13.] X. Li, “Path planning of intelligent mobile robot based on Dijkstra algorithm,” *Journal of Physics: Conference Series*, vol. 2083, p. 042034, 2021.

[14.] E. Oneil and I. Sugiarto, “Omni-Directional Mobile Robot Control using Raspberry Pi and Jetson Nano,” *International Journal of Industrial Research and Applied Engineering*, vol. 4, pp. 57–62, 2019.

[15.] M. Noto and H. Sato, “A method for the shortest path search by extended Dijkstra algorithm,” in *2000 IEEE International Conference on Systems, Man and Cybernetics*, Nashville, TN, USA, 2000, pp. 2316–2320.

[16.] M. B. Alatise and G. P. Hancke, “A Review on Challenges of Autonomous Mobile Robot and Sensor Fusion Methods,” *IEEE Access*, vol. 8, pp. 39830–39846, 2020.

[17.] A. Alyasin, E. I. Abbas, and S. D. Hasan, “An Efficient Optimal Path Finding for Mobile Robot Based on Dijkstra Method,” in *2019 4th Scientific International Conference Najaf (SICN)*, Al-Najaf, Iraq, 2019, pp. 11–14.

[18.] J. Amin, J. Bokovic, and R. Mehra, “A Fast and Efficient Approach to Path Planning for Unmanned Vehicles,” in *AIAA Guidance, Navigation, and Control Conference and Exhibit*, American Institute of Aeronautics and Astronautics, 2016.

[19.] R. Hercik, R. Byrtus, R. Jaros, and J. Koziorek, “Implementation of Autonomous Mobile Robot in SmartFactory,” *Applied Sciences*, vol. 12, p. 8912, 2022.

[20.] M. Luo, X. Hou, and J. Yang, “Surface Optimal Path Planning Using an Extended Dijkstra Algorithm,” *IEEE Access*, vol. 8, pp. 147827–147838, 2020.

[21.] L. C. Garaffa, M. Basso, A. A. Konzen, and E. P. de Freitas, “Reinforcement Learning for Mobile Robotics Exploration: A Survey,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, pp. 3796–3810, 2023.

[22.] R. Raj and A. Kos, “A Comprehensive Study of Mobile Robot: History, Developments, Applications, and Future Research Perspectives,” *Applied Sciences*, vol. 12, p. 6951, 2022.

[23.] J. Yang, Z. Qu, J. Wang, and K. Conrad, “Comparison of Optimal Solutions to Real-Time Path Planning for a Mobile Vehicle,” *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 40, pp. 721–731, 2010.

[24.] H. Nassima, “Design and Implementation of an Autonomous Mobile Robot Based on ESP-32,” M.Sc. thesis, University 8 Mai 1945- Guelma, 2023.

[25.] H. Wang, X. Qi, S. Lou, J. Jing, H. He, and W. Liu, “An Efficient and Robust Improved A* Algorithm for Path Planning,” *Symmetry*, vol. 13, p. 2213, 2021.

[26.] A. F. Hussein and H. A. R. Akkar, “Intelligent Controller Design based on wind-solar system,” *Engineering and Technology Journal*, vol. 39, pp. 326–337, 2021.

[27.] H. Liu, N. Stoll, S. Junginger, and K. Thurow, “A Floyd-Dijkstra hybrid application for mobile robot path planning in life science automation,” in *Proceedings of the 2012 IEEE International Conference on Automation Science and Engineering (CASE)*,Seoul, Korea, 2012, pp. 279–284. doi: 10.1109/CoASE.2012.6386325.

[28.] M. M. Abdulwahid, M. S. Al-Hakeem, M. F. Mosleh, and R. A. Abd-alhmeed, “Investigation and optimization method for wireless AP deployment based indoor network,” *IOP Conference Series: Materials Science and Engineering*, vol. 745, p. 012031, 2020. doi: 10.1088/1757-899X/745/1/012031.

[29.] J. Yim, S. Ganesan, and B. H. Kang, “Location-Based Mobile Marketing Innovations,” *Mobile Information Systems*, vol. 2017, p. 1303919, 2017. doi: 10.1155/2017/1303919.

[30.] M. Kuang, P. Ouyang, W. Yue, and H. Kang, “Nonlinear PD Terminal Sliding Mode Control for PUMA Robotic Manipulator,” in *Proceedings of the 2019 14th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, Xi’an, China, 2019, pp. 250–255. doi: 10.1109/ICIEA.2019.8833984.