IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Mathematical Modelling of Engineering Problems
Vol. 12, No. 8, August, 2025, pp. 2732-2740
Journal homepage: http://iieta.org/journals/mmep

Real-Time Control System for Mobile Robot with Intelligent Collision Avoidance

Iman S. Kareem

College of Electromechanical Engineering, Navigation and Guidance Department, University of Technology, Baghdad 10066, Iraq

Corresponding Author Email: 50071@uotechnology.edu.iq

https://doi.org/10.18280/mmep.120814

Received: 13 December 2024
Revised: 13 March 2025
Accepted: 21 March 2025
Available online: 31 August 2025

Keywords:
collision avoidance, Grey Wolf Optimization (GWO), mobile robot, path planning, Robot Operating System (ROS), Simultaneous Localization and Mapping (SLAM)

ABSTRACT

Effective collision avoidance and path planning are critical challenges in mobile robotics, especially in dynamic environments where obstacles continuously change positions. This research presents a real-time autonomous control system for a mobile robot using the Robot Operating System (ROS) and a camera-based collision avoidance system. Simultaneous Localization and Mapping (SLAM) enables the robot to navigate and construct maps of unknown environments. To optimize path planning and obstacle avoidance, a simulation model based on the Grey Wolf Optimization (GWO) algorithm is implemented in MATLAB, handling both fixed and dynamic obstacles. The system operates online, continuously updating the robot's path to ensure collision-free navigation in real time. Experimental results confirm that the robot successfully avoids obstacles, dynamically adjusts its speed and trajectory, and maps its surroundings. The simulation further demonstrates that the system adapts to varying obstacle sizes and positions, proving the effectiveness of the GWO algorithm in precise path planning and real-time collision avoidance in dynamic environments.

1. INTRODUCTION

Autonomous applications are a modern domain in technology and should be implemented in all robotic models instead of conventional engineering controllers [1, 2].

Due to the rapid advancements in robotics, mobile robots used for indoor service applications have become increasingly prevalent. This is especially relevant during the current period, as the world is facing the spread of the Coronavirus. However, this type of robot encounters several challenges, the most significant being how to achieve autonomous mobility indoors [3].

To address this issue, mobile robots must be equipped with sensors that can gather information about their surrounding environment. This information is then used to generate a map of the environment, which helps determine the robot's location within its workspace. This process is known as Simultaneous Localization and Mapping (SLAM) [4-7].

Although mobile robots are considered robust and capable of supporting both software and hardware, they are costly and not highly scalable. This is primarily due to the fact that most robot programs are not open-source, making the development of mobile robot programs from scratch both complex and time-consuming.

Robot Operating System (ROS) is an open-source software framework that offers a low-cost and highly scalable solution for mobile robotics [8-10]. It provides tools and libraries that assist software developers in creating robotic applications. Additionally, ROS offers device drivers, libraries, visualization tools, message-passing capabilities, and software packages.

ROS also enables the execution of specific parts of the code, allowing the robot to be operated and controlled remotely. Furthermore, it supports the integration of code written in different programming languages, such as Python, C++, and Lisp, into a single, unified network [11, 12].

2. RELATED WORK

There are many studies are done in the field of autonomous mobility of mobile robots.

Sayed et al. [13] focused on the model of an autonomous central multi-bot system. It is containing a golem Hexapod and a six-wheeled mobile robot. Where the Hexapod is used to scan and map the sector hospital setting with plotting a route, the six-wheeled mobile robot acts as delivery of the medical product entering supported by the map and path by SLAM.

Khalifa et al. [14] presented development mechanical design for amphibious robot with navigation system-based reading of sensor Global Positioning System (GPS) and the soft were program LabVIEW from National Instruments (NI). The results did not appear because of the device inside the corridor so is blocked from picking up the satellite signal where it needs an open space and no roof to pick up the signal from the satellite. No collusion accrued depending on the mechanical design of wheel leg – propeller protects the robot body not depending on navigation system.

Li et al. [15] addressed the path planning challenges of an intelligent vehicle in an unknown environment, this study proposed a method for map construction and path planning based on multi-sensor data fusion. First, an extended Kalman filter (EKF) integrates information from LiDAR and a depth camera, while a pose sensor provides the robot's pose and acceleration data. A SLAM system is then developed by combining LiDAR, a depth camera, and an inertial measurement unit. Next, an improved ant colony algorithm is employed for global path planning, while the dynamic window method is utilized for local planning and obstacle avoidance. Finally, experiments conducted on a robotic platform confirm the reliability of the proposed approach. Results showed that the multi-sensor fusion method enhances SLAM accuracy and robustness, producing maps that more accurately reflect the real environment. Additionally, the improved ant colony algorithm smooths path turns, while the dynamic window method enables real-time obstacle avoidance, leading to more efficient path planning and enabling automatic feedback control of the intelligent vehicle.

Najim et al. [16] presented the designing and implementation of a mobile robot travels in any directions without using the steering gear, using four meconium wheels the proposed robot works in the hospital for deliver medications to patients for minimizing human touch. The main component was depth camera and LIDAR used as sensors to discover the. The robot is run by a ROS that employs the SLAM algorithm.

According to the research by Chi et al. [17], the use of indoor laser mapping technology has advanced significantly in mobile robotics. However, traditional 2D LiDAR is limited to scanning at a fixed height, making it difficult to capture objects below that level, leading to inaccurate mapping and navigation errors. While 3D LiDAR provides a more comprehensive solution, its high cost and computational demands limit its application in indoor environments. To overcome these limitations, this study proposed a laser data compensation algorithm based on indoor depth map enhancement. The approach involved to Enhancing depth maps from a depth camera using bilateral filtering and reducing dimensionality through multi-layer projection to generate pseudo-laser data. Then Remapping laser data based on the spatial relationship between sensors and obstacles. Integrating the fused laser data into the SLAM front-end for improved multi-sensor data fusion. Simulations and real-world experiments demonstrated that this method significantly enhances environmental perception and improves mapping accuracy compared to existing fusion techniques.

Yao [18] focused on indoor mobile robots, addressing both global and local path planning to enhance the design principles and theories of navigation algorithms. The proposed approach includes: Global path planning using an improved A algorithm*, and Local path planning based on an enhanced artificial potential field (APF) algorithm. Since a single planning method may not be sufficient for complex indoor environments, a hybrid path planning algorithm is introduced, combining static global optimization with dynamic real-time local adjustments. The improved algorithms were implemented on an actual robot platform to ensure precise obstacle avoidance and efficient pathfinding. The proposed method was validated through MATLAB simulations, confirming its effectiveness. Additionally, a ROS-based indoor mobile robot experiment demonstrated that the hybrid algorithm improves real-time path planning in complex environments, proving its reliability and advantages in practical applications.

Wu et al. [19] introduced a LiDAR-inertial SLAM approach that integrates DA-LIO and MC-PGO to address challenges related to inaccurate movement distortion estimation, weak pose graph constraints, and frequent feature degeneracy. In particular, it presented an innovative dual B-spline-based motion distortion correction technique that leverages a continuous-time trajectory model to refine integrated IMU pose updates, improving correction accuracy. Furthermore, a degeneracy-aware Kalman update strategy is incorporated to enhance system robustness.

Cooper-Baldocka et al. [20] presented a wake-informed 3D path planning method for Autonomous Underwater Vehicles AUVs, integrating wake effects and global currents. In this study the A* algorithm, are used due to their ability to find optimal paths in discretized environments. Two A* variants (current-informed and wake-informed) and neural network approximations are improved. The wake-informed A* planner reduces energy use by up to 11.3%, while neural networks provide faster computation but less optimal paths (4.51%–19.79% higher energy). The findings highlight the importance of wake-aware planning for efficient and safe AUV navigation.

Li et al. [21] presented an advanced control strategy for mobile robots navigating environments with obstacles. In this study Model Predictive Control (MPC) integrated with Terminal Sliding Mode Control (TSMC) to improve trajectory tracking and ensure robust obstacle avoidance. This combined approach leverages the predictive capabilities of MPC to anticipate and plan optimal paths, while TSMC contributes to system robustness against disturbances and uncertainties. Simulation results demonstrated that this integrated control scheme significantly improves the robot's ability to follow desired trajectories accurately while effectively avoiding obstacles, even in dynamic and uncertain settings.

As reported by Qiu et al. [22], a 3D path planning approach is introduced, combining game theory and particle swarm optimization (PSO) in a hybrid framework (GTPHM). Firstly, the path planning task is reformulated into an optimization trouble by redefining the cost function, and a multi-robot motion model is structured to meet dynamic constraints. Within this framework, game theory is employed for multi-robot path planning, when collision costs and multi-objective heuristic functions act as game payoffs to sustain interactive decision-making among robots. To improve performance, the PSO algorithm is incorporated, utilizing a three-dimensional vector-based particle position transformation technique as a game-theory-driven strategy update mechanism. For collision avoidance, each robot adapts its cooperative strategy in response to the actions of neighboring robots, gradually converging toward the Nash equilibrium. Experimental comparisons demonstrate that the proposed GTPHM effectively enables robots to generate safe and collision-free trajectories from the start to the target point, even in challenging 3D terrains such as mountainous and urban environments.

In the study conducted by Chen et al. [23], an improved Multi-Strategy and Improved Adaptive Reinforcement Grey Wolf Optimization (MSIAR-GWO) algorithm for mobile robot path planning is presented. Development included reinforcement learning-based parameter tuning, an adaptive position-update strategy, and artificial rabbit foraging techniques to improve convergence speed, global search ability, and solution accuracy. Compared to traditional algorithms, MSIAR-GWO achieves shorter, smoother paths with better stability and robustness, making it highly effective for autonomous navigation.

Most previous studies either presented control system models that govern a mobile robot's movement in practice or focused solely on simulations. Studies utilizing ROS and SLAM demonstrated system efficiency in practical applications, while those employing artificial intelligence algorithms successfully generated collision-free paths. However, they did not achieve real-time path planning that adapts immediately to dynamic obstacles.

In the present work, a practical design model used ROS which is utilized for real-time control of mobile robots, and it is crucial in dynamic environments with moving obstacles. ROS offers extensive library support with a large collection of pre-built packages for robotics applications, reducing development time and enhancing functionality. It enables seamless communication between sensors (camera), improving perception and decision-making.

SLAM is used to ensure that the robot can autonomously explore and localize itself by combining sensor data (camera) to enhance the accuracy of self-positioning, reduce drift errors, and generate an accurate map of its surroundings.

In addition, this paper investigates a simulation model for path planning of a proposed robot using GWO which is a nature-inspired metaheuristic algorithm that mimics the hunting behavior of grey wolves to find optimal paths while avoiding obstacles. GWO provides an efficient, adaptive, and optimized path-planning method, making it ideal for real-world robotics applications.

The main contribution of this work combining between practical design and simulation model. The design of a real-time control system for mobile robots, enabling autonomous navigation with collision avoidance. Additionally, the robot can map its surroundings. The simulation design is the novelty lies in the investigation of the GWO algorithm for path planning, allowing the robot to adapt to moving obstacles during program execution.

3. TOOLS AND METHODOLOGY

Mobile robot is ground vehicle trained to support humans in several fields, including military and civilian applications [24-27].

The robots are a combination of mechanical and electrical components, with computer science. Robot construction, in general, has an actuator may be DC, AC servo motors or hydraulic actuator to move the wheels or legs so induced the locomotion of the robot body, or that the robot can move from one point to another [28, 29].

3.1 TurtleBot3 Burger mobile robot of a practical aspect

The mobile robot in the present paper is a TurtleBot3 Burger robot as shown in Figure 1. This open-source robotics platform is used for learning and research. The most specifications and features of the TurtleBot3 Burger is small size, and small lightweight, making it suitable for indoor use.

The features are as follows:

- Fully compatible with ROS.
- **The Controller is** Raspberry Pi.
- **Autonomous Navigation** Supports SLAM.
- **Sensors are** 360° LIDAR (LDS-01) for mapping & obstacle detection, and IMU for motion tracking, and encoders for precise motor control.
- Two Dynamixel XL-430-W250 motors for differential drive with omnidirectional movement.
- Expandable with extra sensors (e.g., cameras, depth sensors), open-source hardware & software for modifications.

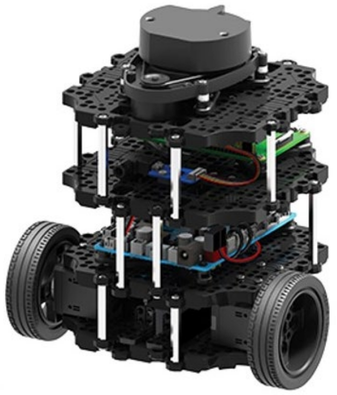


Figure 1. TurtleBot3 Burger mobile robot

Table 1 shows the main specifications of a proposed TurtleBot3 Burger robot.

Table 1. Specifications of the TurtleBot3 Burger

Items	Specification
Maximum translational velocity	0.22 m/s
Maximum rotational velocity	2.84 rad/s (162.72 deg/s)
Maximum payload	15 kg
Size (L × W × H)	13.8 cm × 17.8 cm × 19.2 cm
Weight (+ SBC + Battery + Sensors)	1 kg
Single Board Computers (SBC)	Raspberry Pi
Actuator	XM430-W250
Laser Distance Sensor (LDS)	360 Laser Distance Sensor LDS-01 or LDS-02
Battery	Lithium polymer 11.1V 1800 mAh / 19.98Wh 5C

The control system is crucial, with the Raspberry Pi 4 Model B (Pi4B) handling decision-making and high-processing tasks (Figure 2). The MT9M001 AR0330 depth camera (Figure 3) is selected based on the operating environment. Using SLAM, the robot processes depth sensor data to generate a 2D map and perform real-time collision avoidance [6, 30].

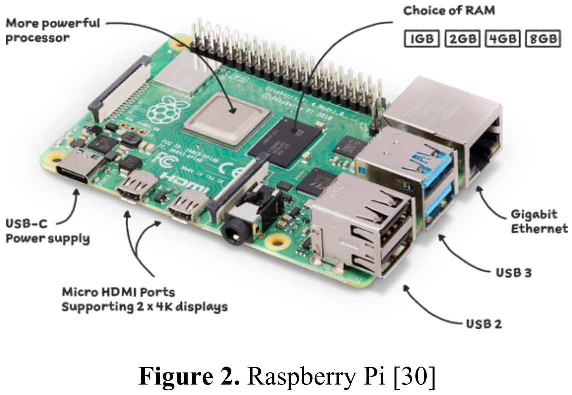


Figure 2. Raspberry Pi [30]

Figure 3. Sensor MT9M001 AR0330 [30]

3.2 SLAM

SLAM enables autonomous navigation in unknown environments using sensors like lasers, depth cameras, and ultrasound. It is widely used in self-driving vehicles, security, warehouse management, and disaster relief [31].

SLAM integrates mapping and localization (Figure 4) to build real-time maps and track robot position in both indoor and outdoor settings. As a stochastic method, it processes environmental data probabilistically, allowing robots to generate optimal paths and navigate back to their starting point in unfamiliar areas [32-34].

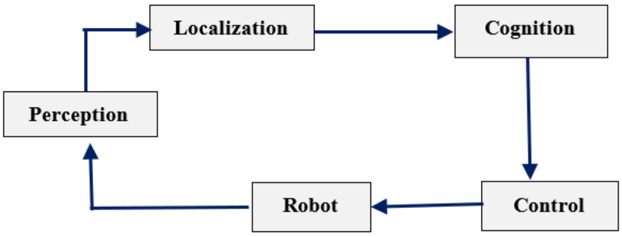


Figure 4. Block diagram of autonomous robot navigation [32]

3.3 ROS program

In the past decade, robotics technology has advanced significantly, with open-source platforms like ROS playing a key role in expanding research and application opportunities. Initially developed for large-scale service robots in Stanford's STAIR project and Willow Garage's Personal Robots Program, ROS has since become widely adopted beyond these domains [6].

ROS has become a leading software framework for robot modeling, simulation, and development, offering a vast environment for transformation and innovation. It supports the creation of new applications and robots, with a variety of tools and infrastructure.

ROS operates through nodes, executable programs that communicate via Topics to exchange information, as illustrated in Figure 5 [35, 36].

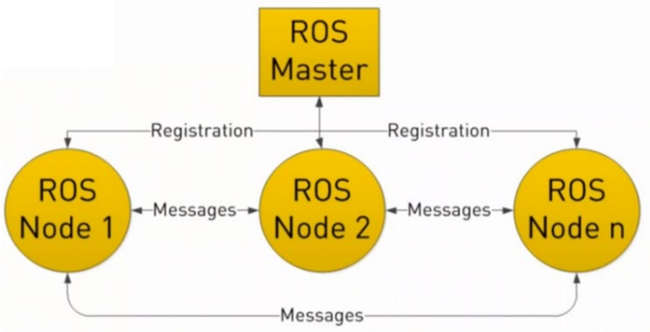


Figure 5. ROS components [35]

3.4 Robot work methodology

All algorithms were implemented in the ROS framework for real-time testing and visualization. The robot is equipped with two sensors: a depth camera and a 360° LiDAR scanner, allowing a performance comparison of the SLAM algorithm when using each sensor.

The process begins with the robot navigating an unknown indoor environment with obstacles, which it must detect while simultaneously mapping the area. Rviz is used to simulate mobile robot movement, control the robot remotely, and visualize SLAM implementation. For practical testing, the TurtleBot3 Burger platform was selected and tested within the Rviz library on ROS (Linux).

Figure 6 shows the basic operation where the depth camera provides depth images. These are converted into laser scans, which SLAM processes to estimate position. SLAM publishes transformations (pose updates) via /tf, linking frames like: Map → Base link (robot position in the global map) and Base link → Camera (camera position relative to the robot). The Rviz reads /tf to display real-time robot movement and SLAM generated maps accurately.

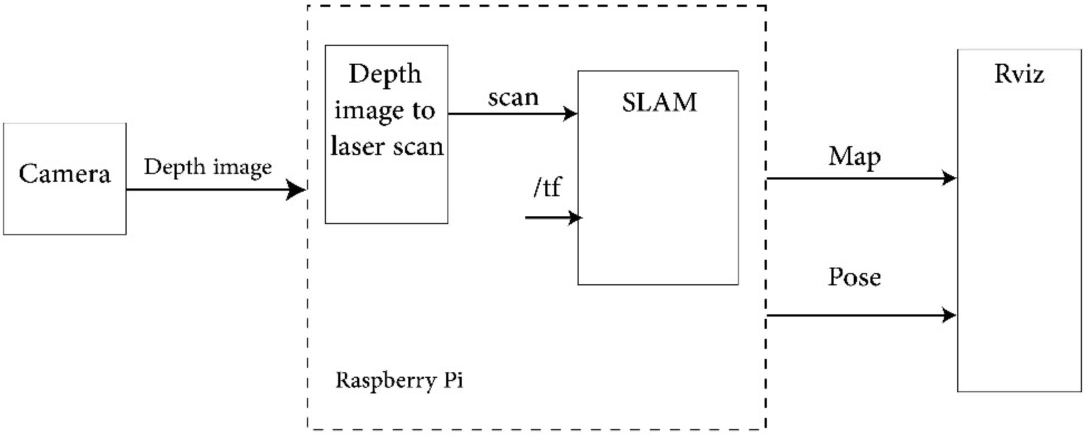


Figure 6. Block diagram for basic operation of robot

4. SIMULATION MODEL FOR PATH PLANNING

To implement path planning for proposed mobile robot used the GWO algorithm with collision avoidance in MATLAB. Figure 7 shows the flowchart for general steps, where, search space limits [x_min x_max; y_min y_max] = [0 100; 0 100], and to show the starting point and end point of the path clearly used the starting position =[5,5], goal position =[95,95], in the current simulation, fixed and dynamic cuboidal obstacles with different dimensions like that are used in the practical testes.in the present algorithm was used seven fixed and four dynamic obstacles distributed by choosing center, length, width, and height for each of them, for any iteration the dynamic obstacles have arbitrary motion.

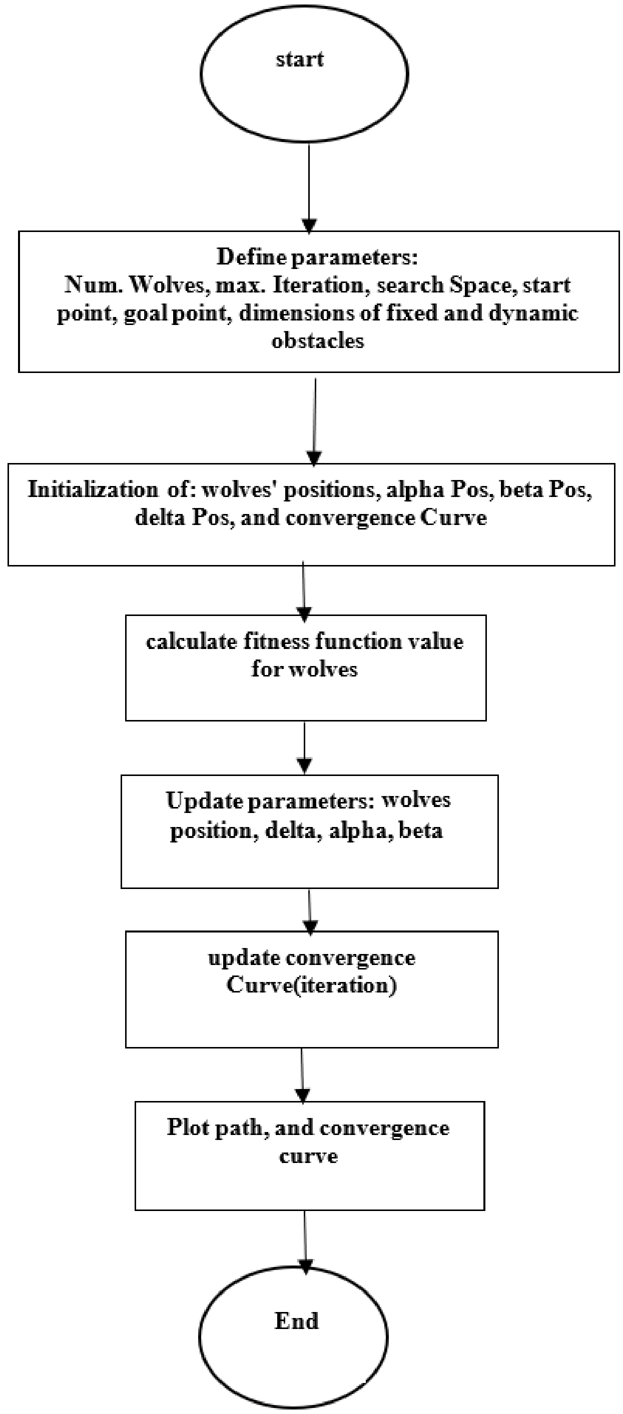

start
Define parameters:
Num. Wolves, max. Iteration, search Space, start point, goal point, dimensions of fixed and dynamic obstacles
Initialization of: wolves' positions, alpha Pos, beta Pos, delta Pos, and convergence Curve
calculate fitness function value for wolves
Update parameters: wolves position, delta, alpha, beta
update convergence Curve(iteration)
Plot path, and convergence curve
End

Figure 7. Flow chart for GWO

5. TEST SETUP FOR PRACTICAL RESULTS

To evaluate the effectiveness of the proposed mobile robot and its collision avoidance control system, several experimental tests were conducted. The operation of the control system was verified in real-world conditions, demonstrating its ability to navigate smoothly while avoiding both fixed and dynamic obstacles.

The setups and procedures are as follows:

- The robot was equipped with a camera, LiDAR, and real-time processing unit to detect obstacles and map the environment.
- The navigation system utilized SLAM for mapping, path planning algorithms for route optimization, and real-time control strategies for collision avoidance.
- The tests were conducted in both structured (corridors) and unstructured (real-world) environments to evaluate the adaptability of the system.

5.1 Corridor navigation test

The first test assessed the robot's movement in a narrow corridor with a length of 9.4 meters and a width of 0.7 meters, as shown in Figure 8. The robot's navigation was based on SLAM system, utilizing data from a camera. The results of this test are summarized as follows:

- Time taken to move forward: Approximately 42.73 seconds.
- Time taken to rotate 180 degrees: Approximately 1.11 seconds.
- Time taken to return after rotation: Approximately 43.83 seconds.
- Total time for a round trip: Approximately 86.56 seconds.

During this test, the robot demonstrated smooth and stable movement, accurately following the SLAM-generated path without significant deviations.

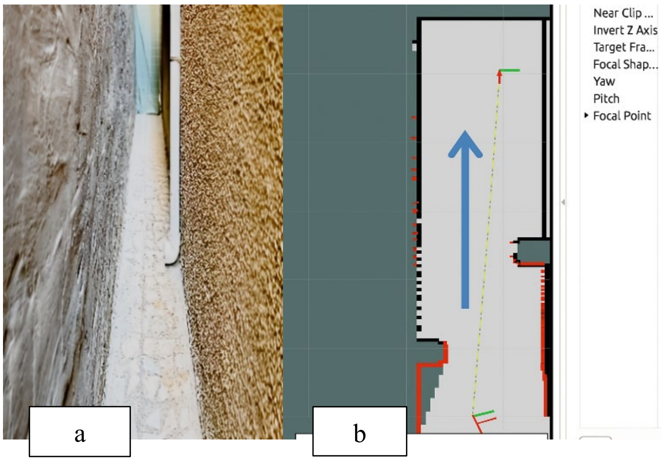


Figure 8. The environment of corridor (a) The actual environment of corridor, (b) The environment of corridor in SLAM

5.2 Obstacle navigation in a real environment

In the second test, the robot was placed in a real-world environment with multiple distributed obstacles, as illustrated in Figure 9. The environment map was generated using SLAM, and the robot's navigation was visualized in RViz, a software tool in ROS, as shown in Figure 10. The red path indicates the robot's movement while mapping. Black areas represent obstacles detected by the robot's LiDAR or depth camera.

The robot successfully detected and avoided both fixed and dynamic obstacles, demonstrating the effectiveness of the control system. The movement results confirmed that the proposed system enables smooth and adaptive navigation, even in complex environments.

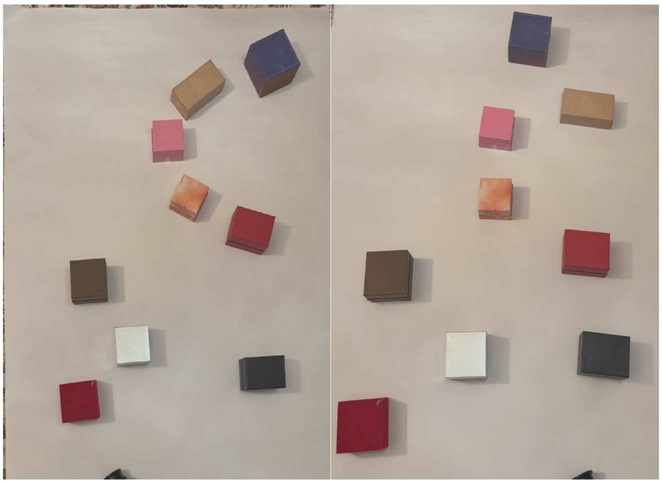
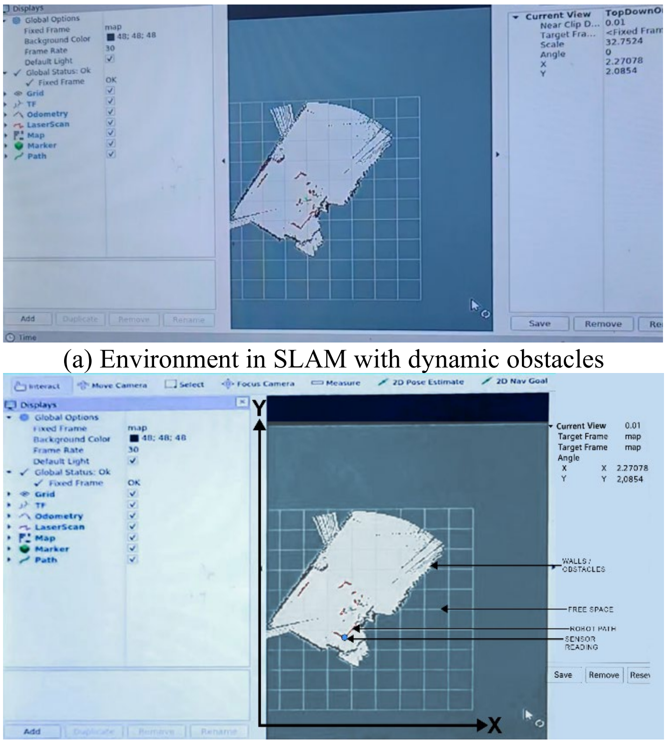


Figure 9. The actual environment (a) Actual environment with dynamic obstacles, (b) Actual environment with static obstacles



Fixed Frame map
Background Color 48; 48; 48
Frame Rate 30
Default Light ✓
Current View TopDownO
Near Clip D... 0.01
Target Fra... <Fixed Fram
Scale 32.7524
Angle 0
X 2.27078
Y 2.0854

(a) Environment in SLAM with dynamic obstacles

(b) Environment in SLAM with static obstacles

Figure 10. Testing the environment of the robot

6. SIMULATION RESULTS AND DISCUSSION

For optimal path planning used GWO algorithm for many iterations to obtain the optimal path drawing with fixed and dynamic obstacles. the simulation model which is adaptive for any change in position of obstacles. The evaluation involved as follows:

a) Initializing the Wolf Pack:
   - A population of 30 grey wolves was randomly positioned in the search space.
   - The wolves' positions were iteratively updated over 100 iterations (maxIter), refining their movement toward the optimal path.

b) Updating Positions Using Alpha, Beta, and Delta Wolves:
   - The alpha wolf represents the best solution (shortest, collision-free path).
   - The beta and delta wolves supply additional guidance to enhance the search.
   - Each wolf updates its position based on attraction to these leaders, with an adaptive step size (a = 2 - iter * (2 / maxIter)) that decreases over time.

c) Collision Handling:
   - A collision penalty (1e6) was added to the fitness function if a path collided with any obstacle.
   - Both fixed and dynamic obstacles were checked for intersection with the computed path to ensure feasible motion planning.

d) Obstacle Dynamics:
   - Movable obstacles were randomly repositioned at each iteration within the search space, simulating real-world dynamic obstacles.

The following metrics were used to evaluate the GWO algorithm's performance:

- Path Length: The fitness function included the total Euclidean distance traveled from the start to the goal. A shorter path indicates better efficiency.
- Computation Time: The execution time per iteration (not explicitly recorded but can be measured using MATLAB. Faster convergence implies better real-time performance.

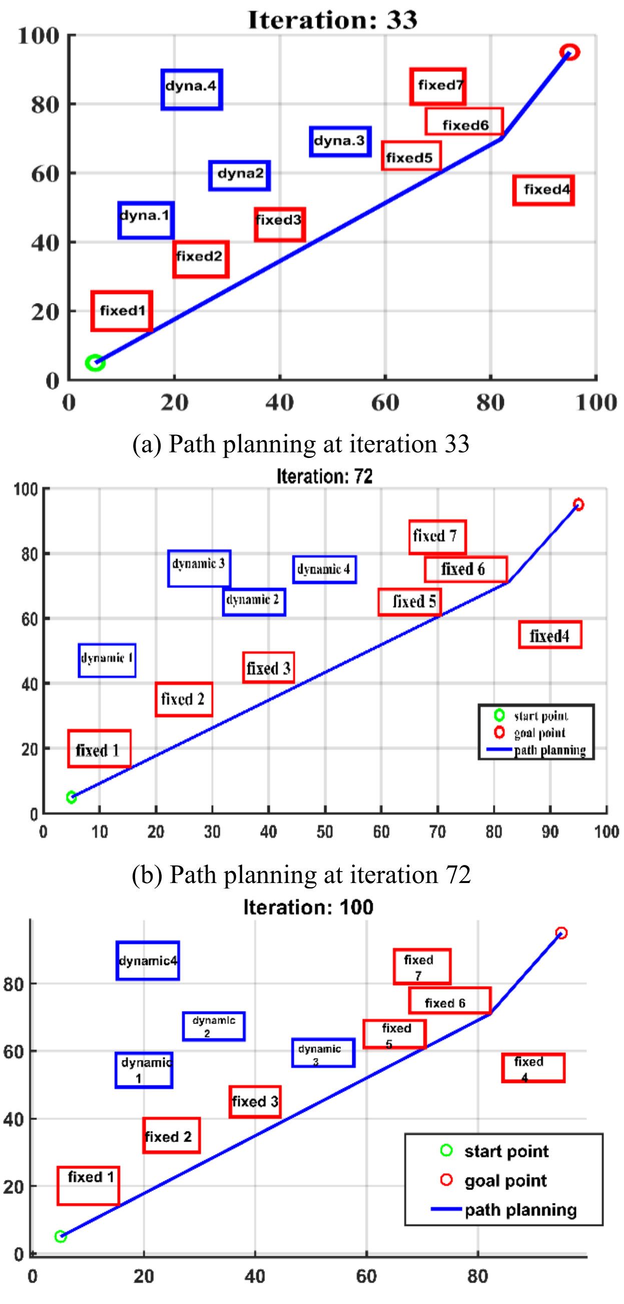
The execution time for each experiment is approximately 17 seconds.

- Convergence Curve: The convergence curve recorded the fitness value of the best solution at each iteration. A steeper decrease in the convergence curve indicates rapid optimization.
- Collision Avoidance Rate: The number of collisions detected vs. collision-free paths provides insight into GWO's ability to navigate safely.
- Dynamic Obstacle Handling: The algorithm was evaluated based on its ability to adapt to changing environments (due to dynamic obstacles) and still reach the goal efficiently.

Several experimental tests were implemented as follows:

The First Test: The fixed obstacles represented by red cubes and the dynamic obstacles represented by blue cubs. Figure 11 shows the path planning of mobile robot collision avoidance, for a same running where Figure 11(a) shows when it reaches iteration 33. Figure 11(b) shows another position for dynamic obstacles for the same running at alteration 77, and Figure 11(c) shows the path planning and last updating of obstacles position at full iteration 100. In this test the shortest path has been achieved, and the mobile robot skip obstacles without collision despite the movement of obstacles along the time of implementation of algorithm.

Figure 12 shows the convergence curve. As iterations progress, the fitness value drops rapidly in the first few iterations, indicating fast initial convergence. After around 20 iterations, the changes become minimal, and the fitness stabilizes close to 128, meaning that the algorithm has found an optimal path planning.



(a) Path planning at iteration 33

(b) Path planning at iteration 72

(c) Path planning at full iteration 100

Figure 11. First testing for simulation path planning of mobile robot collision avoidance

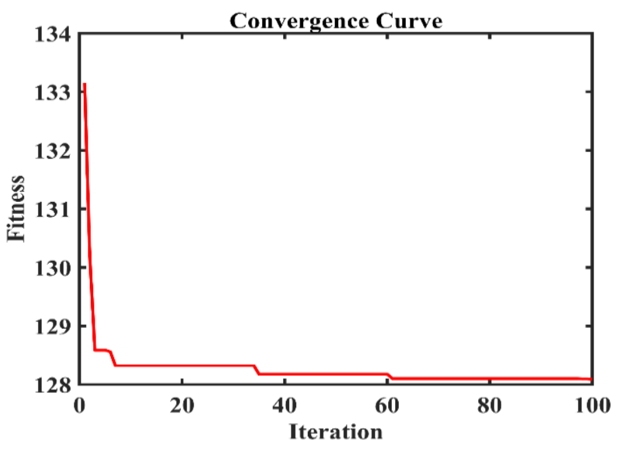
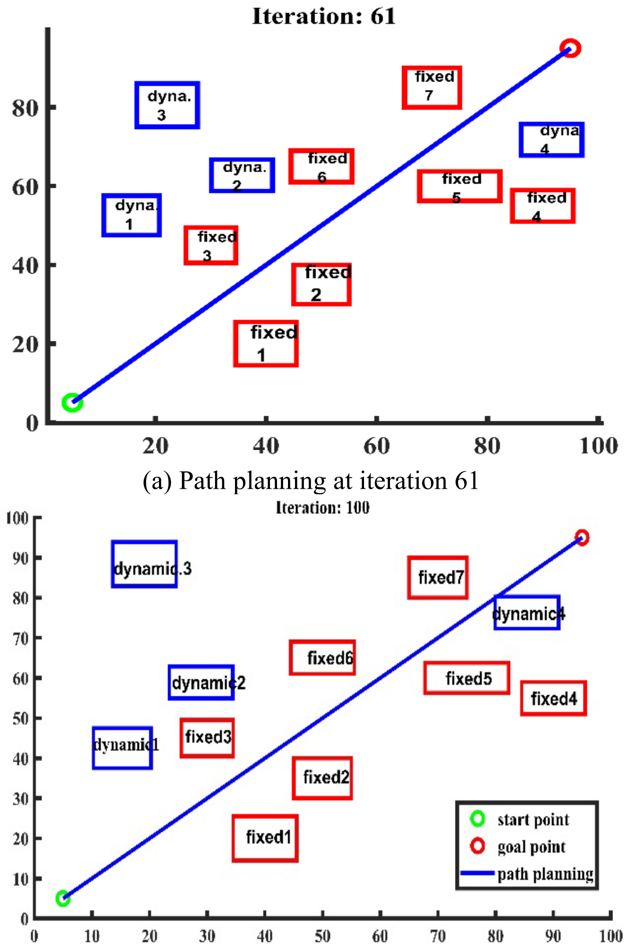


Figure 12. Convergence curve for first test

The Second Test: In this steep the same dimensions of fixed and dynamic obstacles are used, but location for some of them is changed Figure 13 shows the path planning of a proposed robot for a one running with new positions for obstacles. Figure 14 shows the convergence curve for this stage. Despite the changing locations of the obstacles, the algorithm was able to plot the path planning accurately as shown in Figure 13. So, it is possible to adapt this algorithm to draw the path of any vehicle as soon as the sizes and locations of the obstacles change.



(a) Path planning at iteration 61

(b) Path planning at iteration 100

Figure 13. Path planning of a proposed robot at one running with new positions for obstacles

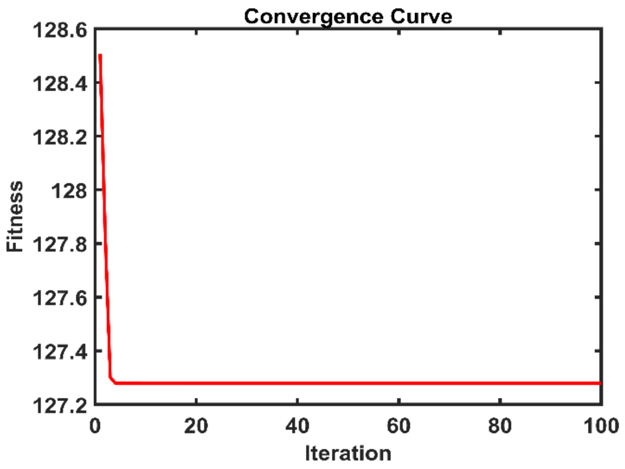


Figure 14. Convergence curve at second test

7. CONCLUSIONS

The data collected from the camera is used to generate a map of the surrounding environment for SLAM, enabling the mobile robot to efficiently detect obstacles. The Rviz platform provides visualization of sensor data, including the camera feed and point clouds. By analyzing the raw sensor values, Rviz offers an autonomous representation of the input data structures.

As a summary, the conclusions of the current research are:

- SLAM enables the robot to autonomously explore, localize itself, and generate an accurate map of its surroundings.
- ROS serves as a robust platform for developing intelligent, real-time, and modular robotic systems.

The results indicate that the system can successfully construct a map compatible with indoor and outdoor environments.

- GWO provides an efficient and adaptive path-planning method, making it ideal for real-world robotics applications.
- The GWO algorithm was explored as an adaptive controller for path planning and obstacle avoidance. Simulation results confirm that the proposed algorithm efficiently computes an optimal path, ensuring collision avoidance even in the presence of continuously moving dynamic obstacles. Unlike traditional path-planning methods, GWO dynamically adjusts the robot's trajectory when obstacles move or change positions. Additionally, it exhibits faster convergence and lower computational complexity compared to conventional approaches.

The Main Contributions & Challenges:

The primary contribution of this work lies in enabling the robot to navigate an unknown environment by mapping its surroundings and autonomously completing navigation tasks using the generated map. Furthermore, the control system, based on sensor feedback, ensures effective collision avoidance for all obstacles.

The proposed intelligent GWO algorithm, implemented in MATLAB, successfully determines the shortest path with minimal execution time, even while adapting to real-time obstacle movement. Unlike previous studies, this work achieves real-time path optimization and visualization, making it a significant advancement in robotic navigation.

8. FUTURE WORK

- Use machine learning models for enhanced obstacle recognition, semantic mapping, and predictive path planning.
- Extend GWO by integrating other metaheuristic algorithms (e.g., PSO, ACO) to balance trade-offs like speed, energy efficiency, and safety.
- Utilize cloud computing or edge AI for real-time data processing and map sharing, enabling seamless multi-robot collaboration.
- Transition to ROS2 for improved real-time performance, security, and support for multi-threaded execution in high-demand applications.

REFERENCES

[1] Kugi, A. (2024). AI-enhanced control in robotics: From classical loops to autonomous systems. In IFAC 2025 MSROB – Keynote Preview. https://ifac2025-msrob.com/keynote-talks.

[2] Ha, Q.P., Yen, L., Balaguer, C. (2019). Robotic autonomous system for earth moving in military applications. Automation in Construction, 107. https://doi.org/10.1016/j.autcon.2019.102934

[3] Rivai, M., Hutabarat, D., Nafis, Z.M.J. (2020). 2D mapping using omni-directional mobile robot equipped with LiDAR. Telkomnika, 18(3): 1467-1474. https://doi.org/10.12928/telkomnika.v18i3.14872

[4] Sadruddin, H., Mahmoud, A., Atia, M.M. (2020). Enhancing body-mounted LiDAR SLAM using an IMU-based pedestrian dead reckoning (PDR) model. In IEEE 63rd International Midwest Symposium on Circuits and Systems (MWSCAS), Springfield, MA, USA, pp. 901-904. https://doi.org/10.1109/MWSCAS48704.2020.9184561

[5] Hasan, H.M., Mohammed, T.H. (2017). Implementation of mobile robot's navigation using SLAM based on cloud computing. Engineering and Technology Journal, 35(6). https://doi.org/10.30684/etj.35.6A.11

[6] Abdulsaheb, J.A., Kadhim, D.J. (2023). Real-time SLAM mobile robot and navigation based on cloud-based implementation. Journal of Robotics. https://doi.org/10.1155/2023/9967236

[7] Roy, R., Tu, Y.P., Sheu, L.J., Chieng, W.H., Tang, L.C., Ismail, H. (2023). Path planning and motion control of indoor mobile robot under exploration-based SLAM (e-SLAM). Sensors, 23(7): 3606. https://doi.org/10.3390/s23073606

[8] Romanov, K. (2018). Development of an OMNI based traction system for a teleoperated mobile robot utilizing ROS platform. LUT University. https://urn.fi/URN:NBN:fi-fe2018052524659.

[9] Prist, M., Cavanini, L., Longhi, S., Monteriù, A., Ortenzi, D., Freddi, A. (2014). A low cost mobile platform for educational robotic applications. In IEEE/ASME 10th International Conference on Mechatronic and Embedded Systems and Applications (MESA), Senigallia, Italy, pp. 1-6. https://doi.org/10.1109/MESA.2014.6935571

[10] Nwankwo, L., Fritze, C., Bartsch, K., Rueckert, E. (2023). ROMR: A ROS-based open-source mobile robot. HardwareX, 14: e00426. https://doi.org/10.1016/j.ohx.2023.e00426

[11] Li, Y., Shi, C.X. (2018). Localization and navigation for indoor mobile robot based on ROS. In 2018 Chinese Automation Congress (CAC), Xi'an, China, pp. 1135-1139. https://doi.org/10.1109/CAC.2018.8623225

[12] Bruzzone, L., Quaglia, G. (2012). Review article: Locomotion systems for ground mobile robots in unstructured environments. Mechanical Sciences, 3: 49-62. https://doi.org/10.5194/ms-3-49-2012

[13] Sayed, A.S., Ammar, H.H., Shalaby, R. (2020). Centralized multi-agent mobile robots SLAM and navigation for COVID-19 field hospitals. In 2nd Novel Intelligent and Leading Emerging Sciences Conference (NILES), Giza, Egypt, pp. 444-449. https://doi.org/10.1109/NILES50944.2020.9257919

[14] Khalifa, M.Z., Kareem, I.S., Jaffer, R. (2020). Development of mechanical design of the amphibious robot. Journal of Mechanical Engineering Research and Developments, 43(2): 61-73.

[15] Li, A.J., Cao, J.P., Li, S.M., Huang, Z., Wang, J.B., Liu,

G. (2022). Map construction and path planning method for a mobile robot based on multi-sensor information fusion. Applied Sciences, 12(6): 2913. https://doi.org/10.3390/app12062913

[16] Najim, H.A., Kareem, I.S., Abdul-Lateef, W.E. (2023). Design and implementation of an omnidirectional mobile robot for medicine delivery in hospitals during the COVID-19 epidemic. AIP Conference Proceedings, 2830(1). https://doi.org/10.1063/5.0156862

[17] Chi, X.N., Meng, Q.Y., Wu, Q.X., Tian, Y.Y., Liu, H., Zeng, P.L., Zhang, B.T., Zhong, C.L. (2023). A laser data compensation algorithm based on indoor depth map enhancement. Electronics, 12(12): 2716. https://doi.org/10.3390/electronics12122716

[18] Yao, J.M. (2023). Path planning algorithm of indoor mobile robot based on ROS system. In IEEE International Conference on Image Processing and Computer Applications (ICIPCA), Changchun, China, pp. 523-529. https://doi.org/10.1109/ICIPCA59209.2023.10257966

[19] Wu, W.T., Chen, C., Yang, B.S., Zou, X.H., Liang, F.X., Xu, Y.H., He, X.F. (2025). DALI-SLAM: Degeneracy-aware LiDAR-inertial SLAM with novel distortion correction and accurate multi-constraint pose graph optimization. ISPRS Journal of Photogrammetry and Remote Sensing, 221: 92-108. https://doi.org/10.1016/j.isprsjprs.2025.01.036

[20] Cooper-Baldocka, Z., Turnockb, S., Sammuta, K. (2025). Wake-informed 3D path planning for autonomous underwater vehicles using A* and neural network approximations. arXiv Preprint, arXiv:2502.01918v2. https://doi.org/10.48550/arXiv.2502.01918

[21] Li, B., Ji, Z.C., Zhao, Z.J., Yang, C.G. (2025). Model predictive optimization and terminal sliding mode motion control for mobile robot with obstacle avoidance. IEEE Transactions on Industrial Electronics, 72(9): 9293-9303. https://doi.org/10.1109/TIE.2025.3536628

[22] Qiu, H., Yu, W.T., Zhang, G., Xia, X., Yao, K. (2025). Multi-robot collaborative 3D path planning based on game theory and particle swarm optimization hybrid method. Journal of Supercomputing, 81: 487. https://doi.org/10.1007/s11227-025-06960-1

[23] Chen, D.F., Liu, J.L., Li, T.Y., He, J., Chen, Y., Zhu, W.B. (2025). Research on mobile robot path planning based on MSIAR-GWO algorithm. Sensors, 25(3): 892. https://doi.org/10.3390/s25030892

[24] Najim, H.A., Kareem, I.S., Abdul-Lateef, W.E. (2023). Omnidirectional mobile robot with navigation using SLAM. Engineering and Technology Journal, 41(1): 196-202. https://doi.org/10.30684/etj.v41i1.2252

[25] Ni, J., Hu, J.B., Xiang, C.I. (2020). A review for design and dynamics control of unmanned ground vehicle. Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering. 235(4): 1084-1100. https://doi.org/10.1177/0954407020912097

[26] Reina, G., Das, T.K., Quaglia, G., Vidoni, R. (2021). Special section: Mobile robots and unmanned ground vehicles. Journal of Mechanisms and Robotics, 13(5): 050301. https://doi.org/10.1115/1.4051702

[27] Latif1, A., Shankar, K., Nguyen, P.T. (2020). Legged fire fighter robot movement using PID. Journal of Robotics and Control, 1(1): 15-19. https://doi.org/10.18196/jrc.1104

[28] Yunardi, R.T., Arifianto, D., Bachtiar, F., Prananingrum, J.I. (2021). Holonomic implementation of three wheels omnidirectional mobile robot using DC motors. Journal of Robotics and Control, 2(2): 65-71. https://doi.org/10.18196/jrc.2254

[29] Rubio, F., Valero, F., Llopis-Albert, C. (2019). A review of mobile robots: Concepts, methods, theoretical framework, and applications. International Journal of Advanced Robotic Systems, 16(2). https://doi.org/10.1177/1729881419839596

[30] Riera, J., Cachiguango, S., Pedraza, M., Andaluz, G.M., Leica, P. (2024). Sliding mode control for trajectory tracking of a TurtleBot3 mobile robot in obstacle environments. Engineering Proceedings, 77(1): 7. https://doi.org/10.3390/engproc2024077007

[31] Zheng, K.S., Wu, F., Chen, X.P. (2021). Laser-based people detection and obstacle avoidance for a hospital transport robot. Sensors, 21(3): 961. https://doi.org/10.3390/s21030961

[32] Zheng, S.R., Wang, J.L., Rizos, C., Ding, W.D., El-Mowafy, A. (2023). Simultaneous Localization and Mapping (SLAM) for autonomous driving: Concept and analysis. Remote Sensing, 15(4): 1156. https://doi.org/10.3390/rs15041156

[33] Abdelhafid, E.F., Abdelkader, Y.M., Ahmed, M., Rachid, D., Abdelilah, E.I. (2022). Visual and light detection and ranging-based simultaneous localization and mapping for self-driving cars. International Journal of Electrical and Computer Engineering, 12(6): 6284-6292. https://doi.org/10.11591/ijece.v12i6.pp6284-6292

[34] Naraharisetti, P.R., Saliba, M.A., Fabri, S.G. (2023). Mapping, localization and navigation for an assistive mobile robot in a robot-inclusive space. In Proceedings of the 20th International Conference on Informatics in Control, Automation and Robotics, pp. 172-179. https://doi.org/10.5220/0012172900003543

[35] Yang, X.Z. (2021). Slam and navigation of indoor robot based on ROS and lidar. Journal of Physics: Conference Series, 1748. https://doi.org/10.1088/1742-6596/1748/2/022038

[36] Thale, S.P., Prabhu, M.M., Thakur, P.V., Kadam, P. (2020). ROS based SLAM implementation for autonomous navigation using Turtlebot. ITM Web of Conferences, 32: 01011. https://doi.org/10.1051/itmconf/20203201011