Results in Engineering 27 (2025) 105837

Contents lists available at ScienceDirect

Results in Engineering

journal homepage: www.sciencedirect.com/journal/results-in-engineering

Review article

Automotive navigation for mobile robots: Comprehensive review

Mayar Abdullah Taleb [a,b,*], Gyula Korsoveczki [a,b], Géza Husi [a]

[a] *Department of Electrical Engineering and Mechatronics, Faculty of Engineering, University of Debrecen, Ótemető Utca 2-4, 4028, Debrecen, Hungary*
[b] *Faculty of Informatics, University of Debrecen, Kassai út 26, 4028 Debrecen, Hungary*

ARTICLE INFO

Keywords:
Mobile robot navigation
Autonomous navigation
Path planning
Mapping
Localization
Simultaneous localization and mapping (SLAM)

ABSTRACT

Effective navigation of mobile robots in a dynamic environment poses complex challenges, including mapping, localization, and path planning. These factors are interdependent and require robust solutions for successful robot navigation. The complexity of the task increases due to the unpredictability of dynamic obstacles that resemble humans or other robots. This research proposes an in-depth investigation into these navigation challenges, emphasizing the development of hybrid algorithms that manage interdependencies more effectively. The aim is to evaluate the performance of various navigation algorithms against a set of clearly defined metrics such as success rate, path efficiency, computational cost, and adaptability to environmental changes. This paper will give special attention to algorithms incorporating machine learning and real-time data analysis to manage dynamic obstacles better; also, it will explore the adaptability and scalability of these algorithms in simulated environments of varying complexity and size, preparing them for real-world applicability. This research intends to bridge the gap between theoretical algorithms and their implementation in real-world scenarios through theoretical analysis, comparative evaluation, and practical field tests. By achieving these objectives, we anticipate offering comprehensive insights and best practices for mobile robot navigation in dynamic environments, paving the way for innovative solutions that enhance robot autonomy and efficiency.

1. Introduction

As stated by the International Organization for Standardization (ISO), the robot is a programmable machine that moves in two axes or more and has a degree of autonomy. It moves in its designated working space to accomplish its intended tasks [1]. Generally, robots can be either fixed or mobile. Fixed robots do not move about specific parts of their environment. Mobile robots can travel around their working area using different ways of locomotion [2], such as wheeled [3], legged [4], tracked slip/ skid [5], and a combination of legged and wheeled locomotion [6].

According to the International Federation of Robotics (IFR), the number of robots operating in the industrial sector is more than 4 million robots [7]. Stationary or fixed robots are usually utilized in the manufacturing sector to automate simple, repetitive tasks previously carried out by human workers [8,9]. These robots are extensively employed and have become indispensable in various industrial processes [10,11]. The types of industrial fixed robots are SCARA, Linear, Cylindrical, Delta, and Articulated robots [12,13]. Dealing with stationary robots requires knowledge of kinematics and dynamics. In general, robot kinematics focuses on the geometric and time-based aspects of motion without considering force, and it splits into:

a) Forward Kinematics (FK), where the position and orientation of the robot end effector are based on the known joint parameters. The most widely used method is Denavit-Hartenberg (DH), which involves assigning coordinate frames to each robot link using four parameters (link length, link twist, link offset, and joint angle). By deriving homogeneous transformation matrices for each joint and multiplying them in sequence to obtain the transformation from the base to the end-effector.
b) Inverse kinematics (IK) determines the joint parameters required to achieve the desired position and orientation of the end effector. Three methods exist to solve the IK problem (Analytic, Numeric, and Decoupling).

In [14], Elhadidy et al. present a comprehensive kinematics analysis of an Articulated robot with 6 Degrees Of Freedom (DOF). On the other hand, robot dynamics is the study of forces and torques and their effects on the robot's motion [15].

---

* Corresponding author.
E-mail address: mayart@eng.unideb.hu (M.A. Taleb).

https://doi.org/10.1016/j.rineng.2025.105837
Received 6 January 2025; Received in revised form 9 May 2025; Accepted 17 June 2025
Available online 18 June 2025
2590-1230/© 2025 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY-NC license (http://creativecommons.org/licenses/by-nc/4.0/).

In contrast, mobile robots are widely used because they offer flexibility and autonomy that fixed robots cannot. Their mobility allows them to navigate various environments, adapt to dynamic conditions, and perform tasks across different locations. The mobile robot’s shape and the wheels’ configuration are critical in determining its performance, stability, manoeuvrability, and control strategies [16,17]. When designing or selecting a mobile robot, different parameters are taken into consideration, such as kinematic and dynamic constraints, sensing perception, and navigation path planning. In [16], Shabalin et al. present a detailed evaluation of different wheel types used in mobile robot platforms, emphasizing how wheel selection and configuration impact manoeuvrability, stability, and suitability for various environments. While Jahn et al. in [18] emphasize that the taxonomy for mobile robots is divided based on type, applications, capabilities, and implementations. Siegwart and Nourbakhsh [17] provide a comprehensive overview of mobile robot kinematics, focusing on how robot design in the mobile robot kinematics chapter includes wheel types and configurations that affect motion, manoeuvrability, and control. Moreover, the book details mathematical modelling techniques for robot motion and emphasizes the challenges of accurate position estimation in mobile robots due to slippage and lack of a fixed base. In [19], Ivanjko et al. present two approaches for accurately modelling the dynamics of mobile robots to improve navigation, localization, and path planning. The first approach relies on the physical modelling of robot components using equations derived from energy and motion dynamics (Euler-Lagrange formulation), while the second uses experimental identification by fitting real-world velocity data to estimate dynamic behaviour.

Introducing the navigation approach in the context of controlling the motion of mobile robots contains three interconnected challenges: mapping, localization, and path planning [20]. These challenges cannot be addressed independently. Additionally, if the robot environment changes or encounters moving obstacles, such as humans or other robots, it must be able to adapt its actions accordingly. Another important consideration is what the robot will do in case of a fire or any emergency. Different problems may appear, and these problems will depend on the working environment. Therefore, developing algorithms and methods to solve these three missions became necessary.

Mobile robot navigation is a rapidly evolving field marked by significant technological advancements. Despite extensive research addressing the different aspects of navigation challenges, there remains a critical demand to consolidate these developments into comprehensive research. This paper aims to bridge that gap by synthesizing the latest research findings and innovations and providing a clear picture of the current state-of-the-art. Since Fig. 1 indicates the intersection between the topics, in which each node size represents the number of papers related to the keyword, and lines represent the links among keyword terms as shown in the figure, the intersection between the topics becomes clear. Recent breakthroughs in sensor technologies, along with enhancements in algorithmic approaches, including deep learning-based methods and integrated systems, have advanced mobile robot navigation. However, these advancements are dispersed across a wide range of publications, making it challenging for researchers to stay current and for practitioners to implement state-of-the-art solutions.

Furthermore, mobile robot navigation inherently spans multiple disciplines, including robotics, computer vision, control theory, and artificial intelligence, necessitating comprehensive research that unifies these diverse perspectives to foster interdisciplinary collaboration and drive practical innovations. The increasing deployment of mobile robots in industrial, medical, military, and exploration contexts underscores the practical relevance of this paper, which examines how contemporary methods address issues such as dynamic obstacle avoidance, sensor fusion, and emergency response challenges that are increasingly noticeable in dynamic and unpredictable environments. Additionally, by critically evaluating current methodologies and their limitations, this research identifies open research questions and potential areas for improvement, thereby providing a roadmap for future studies and assisting practitioners in understanding the trade-offs inherent in different navigation approaches. A consolidated resource that compares various navigation strategies and technologies is invaluable for academic researchers and industry practitioners, as it aids in making informed decisions regarding system design, sensor selection, and algorithm implementation, ultimately leading to more efficient and safer mobile robot deployments. By addressing these motivations, this research is a foundational resource summarizing existing knowledge and illuminating future research directions in mobile robot navigation.

Existing technologies in mobile robot navigation combine diverse approaches from sensing and perception to planning and control to enable autonomous operation. These technologies include LIDAR-based

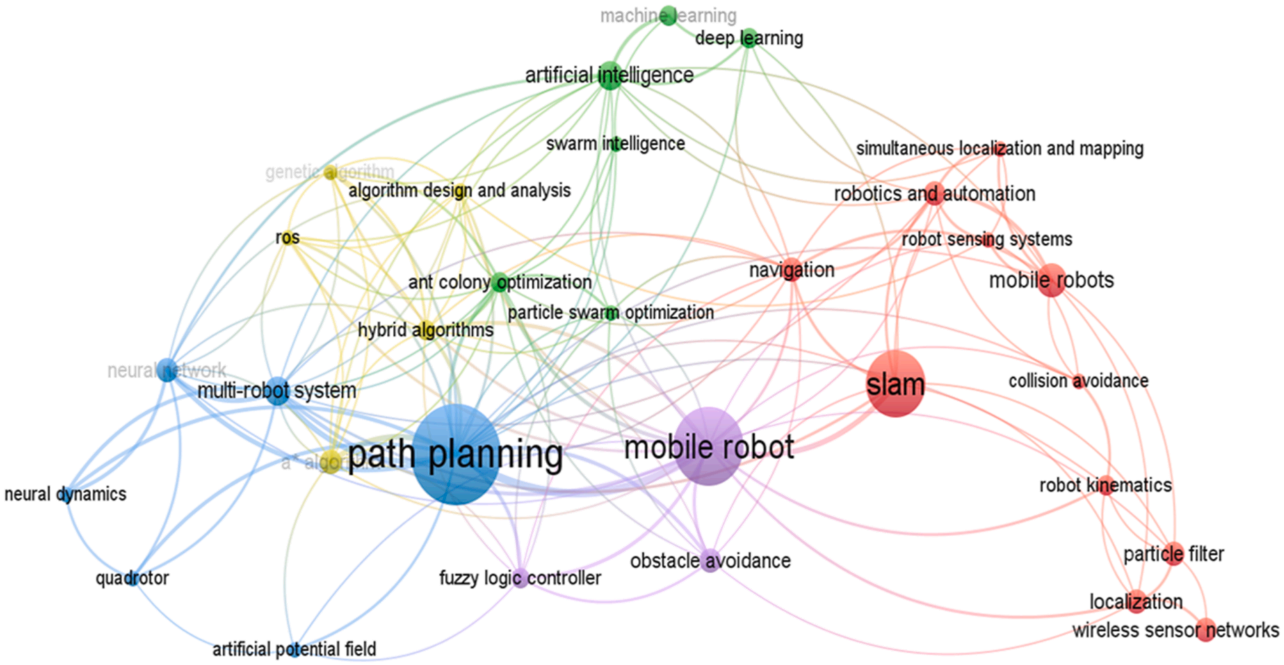

Fig. 1. Configuration of the keyword map for co-citation analysis of the research terms.

technologies, ultrasonic and infrared sensors, and vision-based and/ or radar-based technologies [21]. Among these, LIDAR-based technology is the dominant sensor technology in mobile robot navigation, noted for its precision and reliability in mapping and localization, despite the significant use of vision-based systems in other applications [22]. There are challenges in vision-based conditions, such as deformation, occlusions, and illumination. Also, viewing the detected object from different angles may look completely different [21,23].

Nowadays, mobile robots serve various purposes as they can access locations unreachable by humans, such as nuclear power plants, sewer pipelines, and even lunar exploration. Mobile robot navigation and object detection are productive topics in research and development because of their advantages. The scope of this work is to summarize the necessary tasks of allowing the robot to construct a map of its circumference, know its location on this map, and detect the objects and obstacles and distinguish them. Furthermore, it chooses its path by using a path planning algorithm. These capabilities will allow mobile robots to operate autonomously in various environments, including medical facilities, industrial fields, agricultural sites, and more [24].

This paper explores recent research on autonomous mobile robot navigation. In Section 2, the paper outlines the basic framework of robot navigation and introduces three key challenges. The integration of the essential navigation tasks together (Mapping, Localization, and path planning) is introduced in Section 3. Moving forward, Section 4 discusses the shift toward smarter navigation systems that can adapt and learn over time using advanced algorithms. Section 5 turns to simulation environments, highlighting their importance for safely developing and testing these systems. Finally, Sections 6 and 7 present a discussion of key findings and conclude with reflections on current limitations and directions for future research.

2. Navigation

Mobile robots could be navigated either through human operation [25–27] or by being fully automated to make autonomous decisions [28,29]. In the case of human control, a remote controller can be used to teleoperate the robot [30,31]. Conversely, a fully automated robot can operate independently without requiring human interactions and make informed decisions based on its environment. By integrating artificial intelligence (AI) [32,33] and machine learning (ML) [31,34] into the robot's system, it can optimize its performance further by allowing it to store and compare data [33,35–37].

Full autonomy can be either the heuristic approach or the optimal approach. Heuristic algorithms employ purpose-built functions to navigate solution spaces effectively and intelligently [38]. These algorithms are practical when exhaustive search methods are not feasible due to the large search space size [39]. The ability of heuristic algorithms to find near-optimal solutions in a reasonable amount of time has made them popular in a wide range of applications, including engineering design, scheduling, and image recognition. Despite their effectiveness, heuristic algorithms are not always guaranteed to find the best possible solution, and the quality of their results depends heavily on the quality of the heuristic functions used. Therefore, careful selection and design of these functions are crucial in achieving optimal performance from the algorithm [40,41].

Optimal approaches focus on choosing the shorter path, which is time-consuming and provides more accurate obstacle avoidance in local and global path planning [42]. It requires more knowledge of the environment. Furthermore, the plan and the resulting actions come from the maximization or minimization of an objective function, which is achieved through optimization [43,44].

In Fig. 2, a detailed flow chart illustrates the typical mobile robot navigation cycle process, starting with system initialization and sensor setup, followed by continuous data acquisition and preprocessing to remove noise. The robot localizes itself while mapping the environment and detecting obstacles. If obstacles are present, it plans an alternative or safe path; otherwise, it chooses an optimal route. The robot generates and executes the navigation commands and then checks whether it reaches the destination. If not, the cycle repeats, ensuring the robot adapts to environmental changes until it achieves its goal.

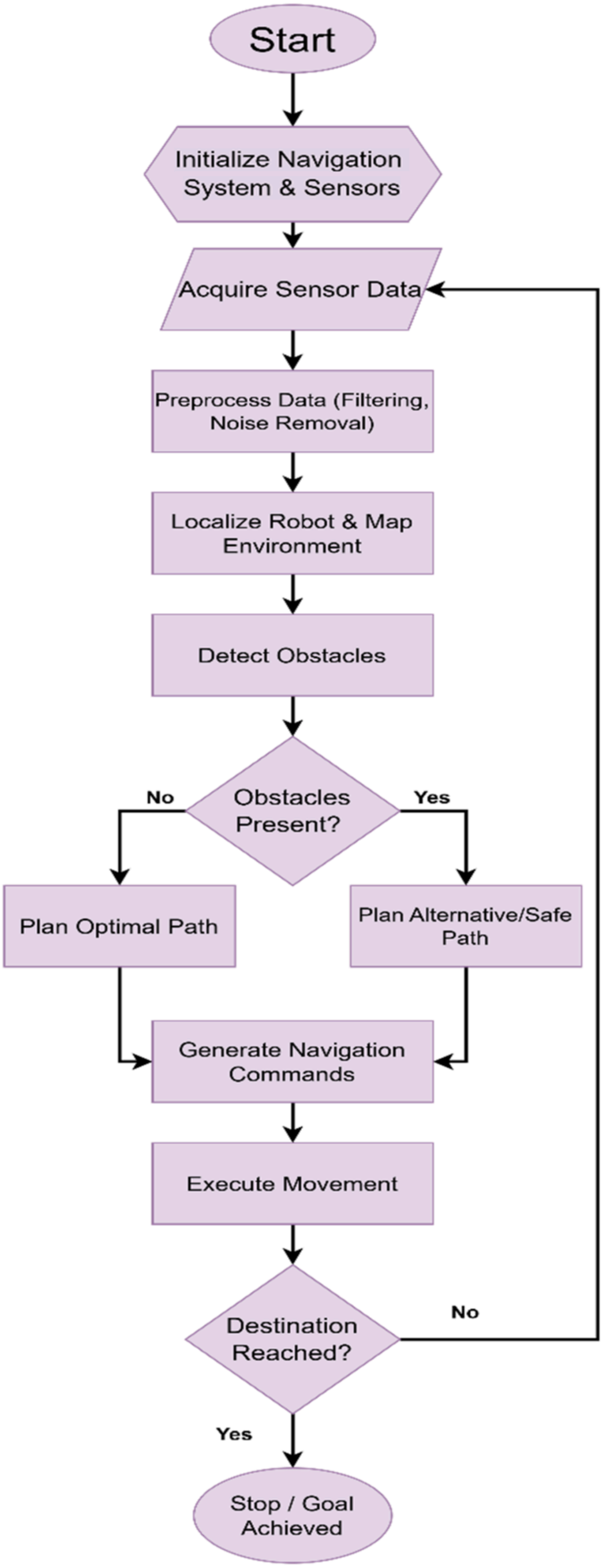

Fig. 2. Mobile robot full navigation approach (Inspired from [32]).

Learning maps using one robot generally requires three duties, as illustrated in Fig. 3: self-localization, path planning, and mapping [20,

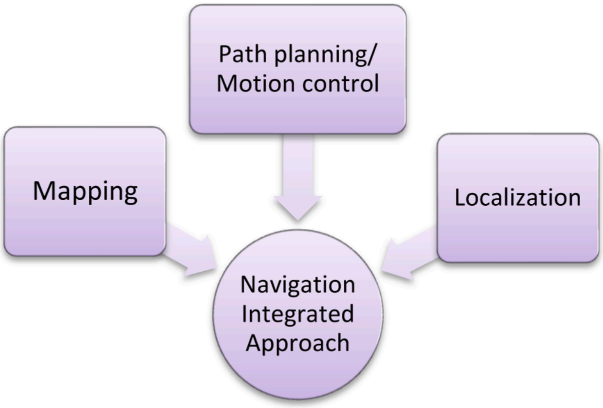

Fig. 3. Full automated navigation approach (inspired from [7]).

45,46]. The mapping refers to the combined data collected by the robot sensors with a particular representation. The inquiry, "What is the world like?" clarifies it. Mapping involves two critical elements: interpreting detector information and sketching the surrounding medium. These two aspects are pivotal in creating a comprehensive and accurate map. The precise interpretation of detector data allows the identification of relevant features and characteristics of the surrounding environment, while sketching provides a visual representation of the mapped area. These tasks require a high level of expertise and proficiency to ensure the clarity and reliability of the final output [47,48]. Thus, localization gauges the robot's position on the map. The mobile robot must know how to answer the question: "What is my position? Where am I?" One recognizes the difference between post-tracking by knowing the vehicle's initial position and the global location, in that no prior information about the starting point is available [49,50]. Path planning or motion control problem is the challenge of guiding a vehicle along a route or to the desired destination. The summarization of the issue is determined as "How do I get to a particular location?".

Regrettably, the three tasks are interdependent. The ability of a robot to comprehend its surroundings is contingent upon its ability to determine the origin of the observations made. The lack of knowledge about the origin locations of comments undermines the ability of a robot to understand its environment effectively. Thus, it is difficult to determine the current location of a vehicle without a map. For planning an optimal path toward the destination, it is imperative to possess a comprehensive understanding of the workspace and an accurate map of the area [20].

2.1. *Localization*

Robot localization indicates locating a robot relative to its environment using sensors [50]. As shown in Fig. 4, the classification of localization algorithms is according to three methods: network, anchor-based, and range-based [51–53]. This paper will focus on these categories.

On the other hand, in [54], Rashed et al. reviewed localization and identification methods for Wireless Sensor Networks (WSNs), and they divided the localization techniques into range-based, range-free, and artificial intelligence-based techniques.

In [55], The authors comprehensively reviewed mobile robot localization by categorizing techniques into three primary groups: GPS-based methods for absolute positioning, relative methods such as wheel odometry and inertial navigation that estimate position changes from onboard sensors, and map-based methods including both prior map matching and Simultaneous Localization And Mapping (SLAM) that use environmental maps to reduce accumulated errors. They discussed the strengths and limitations of each approach and highlighted the importance of sensor fusion in enhancing accuracy and robustness across various environments.

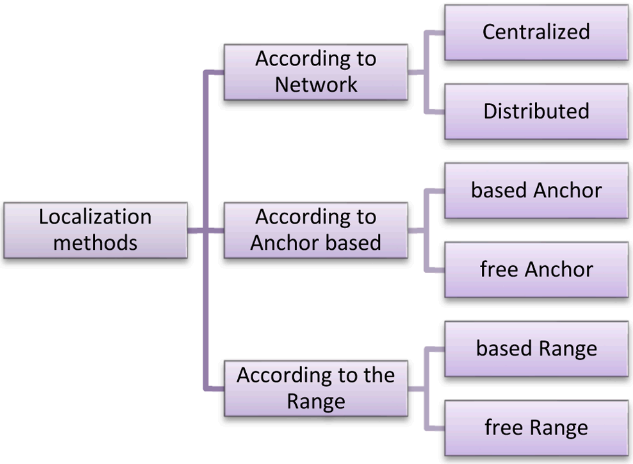

Fig. 4. Classification of localization methods.

In another comprehensive review of localization strategies for autonomous mobile robots [56], authors categorized the localization problem based on initial positional information into three classes: position tracking (where the initial position is known), global localization (with no prior knowledge of the starting position), and the kidnapped robot problem (addressing recovery when the robot is unexpectedly displaced). Then, it reviews key approaches such as probabilistic methods — including Markov localization, Kalman filter-based techniques (and their extended and unscented variants), and Monte Carlo localization — alongside autonomous map-building methods like SLAM (with detailed discussions on EKF-SLAM and UKF-SLAM) and RFID-based schemes. The paper further examines evolutionary approaches, such as those based on particle swarm optimization and genetic algorithms, to optimize localization performance, and it analyses the errors in position and orientation associated with different strategies while suggesting future research directions.

2.1.1. *According to the network architecture (distributed and centralized localisation algorithms)*

Distributed localization [57,58], where each node can locate itself, is generally more efficient in computation and easier to implement in large networks. That is because communication costs in the center are lower than the energy consumed. They are, however, more challenging to design due to the potentially complicated relationship between global and local behaviour. Also, they require multiple iterations to find a stable solution. The localization process may take longer than is necessary in specific cases. Having nodes transmit collected information directly to the centralized unit defines centralized localization [59]. It is the place where the information is processed to extract location data. However, they may not be feasible for large-scale sensor networks. They also have the disadvantage of being more complex, less reliable, and consuming more energy than computation [51,52].

2.1.2. *According to the anchor based (free anchor and based anchor methods)*

The nodes of Anchor know their coordinates via GPS or manual positioning. Free Anchor localization algorithms measure density by the average value of nearby neighbours. The minimum density of the neighbours is necessary for maximum localization coverage, or the density can determine a reasonable amount of precision. The description of anchor-based localization algorithms is possible in terms of anchor density. On average, a node has several anchors within its efficient radio transmission region or measurement area [51,53,58].

2.1.3. *According to the range (free range and based range methods)* [60]

These algorithms are range-free and include Hop counting and neighbourhood techniques. These methods only use connectivity information to locate the node. They are very cost-effective. However, the results can be inaccurate. Angle of Arrival (AoA, Received Signal Strength Indication (RSSI), and Time Difference of Arrival (TDoA) techniques are all range-based techniques. These algorithms are more precise for localization but require additional hardware for AoA or TDoA [52,61].

2.2. *Mapping*

Mapping points out the capability of a device to create its map by providing Allothetic and Idiothetic inputs. Different ways exist to combine them into an exemplification of robotic navigation utilization. In[62–64] authors split the congruent samples into two styles using topological or metric maps. The metric maps store position data of specific obstacles the device may encounter. Topological store allothetic descriptions of the positions the device can reach and data about their relative locations, as shown in Fig. 5.

The definition of the metric framework is a representation of the environment as a collection of objects having 2D coordinates in space. This representation is essential because idiothetic information allows the robot to be directly tracked in this space. The 2D space is more accurate than the topological framework because the position estimate is continuous. Metric maps show the environment's layout in an easy-to-read manner, much like an architectural sketch. Robots can reuse these maps because they provide an objective view of the environment. Because of their coordinates, metric maps are much simpler to construct than topological maps [63,65].

The topological framework depicts the environment as a collection of distinct areas. Moreover, robots can drive from one location to the next. Allothetic data at the appropriate location in the environment is required to define a place. The graph links relating to different places often contain idiothetic information collected while moving from one place to another. Topological maps have the advantage that they can abandon the metric sensors that need to convert allothetic inputs into a 2D frame. They are also closely connected to the robot's perceptual capabilities and do not require the extraction of objective representation. However, it is possible to use allothetic information only for the physically explored locations of the robot, which means that more detailed exploration is required to determine the location of objects. The definition of places is another problem; this can be difficult with unreliable sensors or a dynamic environment [63,64].

Racinskis et al [66] classify them into three primary types of maps for mobile robots. Metric maps represent the precise geometry of an environment using formats like occupancy grids, point clouds, surfels, meshes, or implicit models such as Truncated Signed Distance Function (TSDF) and Neural Radiance Fields (NRFs) to capture distances, obstacles, and spatial structure. Topological maps, on the other hand, abstract the environment into nodes and connections, emphasizing navigational relationships rather than exact measurements, which are especially useful for high-level planning. Finally, semantic maps enrich these representations by incorporating object identities and other high-level information, enabling robots to interpret and interact with their surroundings more intelligently. This multi-layered mapping framework is essential for robust autonomous navigation and decision-making.

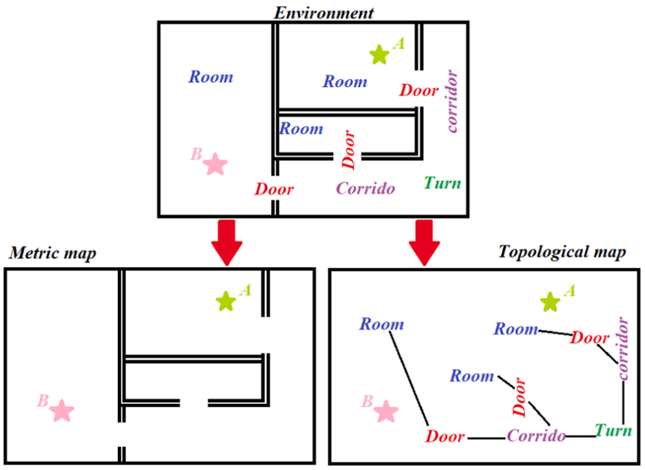

Fig. 5. Illustration of two common representations of maps.

In [67], Lang and Paulus introduced metric maps that capture exact spatial details using formats such as occupancy grids, point clouds, or surfel models to describe the environment's geometry. The topological maps abstract the environment into a graph structure where nodes represent significant landmarks or regions and edges represent the connectivity between these areas, facilitating efficient high-level planning and navigation. Hybrid maps seek to integrate the detailed spatial accuracy of metric maps with the structural simplicity of topological maps to leverage the strengths of both approaches. Based on that, semantic maps extend hybrid representations by incorporating objects and placing classifications with common-sense knowledge, thus enabling the map to convey high-level, human-interpretable information.

Soorchaei et al [68] classify the representation of maps into three primary types. Firstly, metric maps provide precise spatial information through formats such as occupancy grids, point clouds, and landmark-based models, which are essential for accurate localization and obstacle avoidance but can be computationally demanding. The second type is the topological maps, which abstract the environment into graph structures where nodes denote distinct places or landmarks and edges represent navigable connections, offering a simplified framework for high-level planning at the cost of spatial resolution. Finally, geometric maps further simplify the environment by approximating objects with basic shapes, thus reducing storage requirements and computational complexity, although they may lack detailed accuracy. Recent research has increasingly focused on hybrid map representations that combine the strengths of metric and topological and semantic approaches, integrating high-level contextual information such as object classifications and common-sense knowledge to facilitate more intuitive human-robot interaction and robust autonomous decision-making. These advances are particularly critical in High-Definition (HD) mapping for automated vehicles, where layered map structures incorporating geometric, semantic, and dynamic information enable precise localization and real-time navigation in complex environments

The localization process is not separable from map learning. Therefore, it can be challenging to include localization errors in maps. Simultaneous Localization and Mapping (SLAM) is a familiar name describing this problem. Later, it will be discussed in detail [66,69].

2.3. *Path planning algorithm (global and local navigation methods)*

Path planning is the most fundamental part of mobile robot navigation; it will find safe waypoints and determine the best path to follow from its initial position to its destination in a working area. Path planning generally involves two main issues: Obstacle prevention and route optimization [70].

There are two categories of path planning for mobile robots based on environmental information: global/ offline path planning or local/ online planning. Table 1 compares them in terms of the environment, the requirements to achieve the target, the expected time, examples of each method, and its adaptability to the dynamic environment.

Table 1 indicates different aspects of the global and local path planning methods. Global path planning methods recognize that the working area and its environment are static. They require the initial and

Table 1
Comparing the two methods of path-planning algorithms [43,71].

Comparison Aspects	Global (offline) Path Planning	Local (online) Path Planning
Environment	Recognizable and static.	Unknown or partially unknown.
Requirement	Initial and final positions	The system builds its surrounding map
Respond time	Relatively slow	Fast response
Algorithms	A* [72], Dijkstra [73], D* [74], Rapidly exploring random tree (RRT) [75]	Dynamic-Window Approach (DWA) [76]
Adaptability to Dynamic Environments	Limited adaptability, pre-built maps, and static assumptions make it less responsive to real-time changes.	Highly adaptive due to the continuously updates of its map, allowing it to respond to dynamic changes
Resource Efficiency	Often resource-intensive, as global searches over large areas require significant computation and memory	Generally, it is more efficient since localized computations and incremental updates reduce overhead.

final positions within a built map to estimate an initial path. These make them relatively slow because the robot generates a feasible path and then moves toward the target in offline mode.

On the other hand, the local path planning methods are suitable in a dynamic environment. The system builds its surrounding map based on the data collected from sensors. These make them respond faster because the robot generates an online path while moving toward the target.

Fig. 6 illustrates a visual comparison between the two methods, in which the global planner required a map to indicate the starting and ending positions to choose the optimal path toward the target. In the local planner, knowledge of the map is not essential because the robot will use the collected data from the sensors to avoid obstacles and select the optimal path.

2.3.1. *Global algorithms details*

Several review articles discuss the topic of path planning in both local and global methods. In [77], Zhuozhen and Hongzhong categorized the global path planning algorithms into three classes: Traditional algorithms based on maps like A* and Dijkstra algorithm; intelligent path planning algorithms based on bionics; and algorithms based on sampling, such as Rapidly-Exploring Random Tree (RRT). Moreover, Lixing et al [71] agreed to classify path planning methods from the perspective of algorithms into three groups: Classical algorithms like sampling-based methods and graph search algorithms. Secondly, bionic algorithms include genetic algorithms, particle swarm and ant colony optimization algorithms, and artificial intelligence algorithms include fuzzy control and neural network algorithms.

However, Zhang et al [78] sort the path search algorithm for global path planning into a heuristic approach and an artificial intelligence algorithm. Meanwhile, [66] highlights global planning and splits it into heuristic and exact methods.

A. Algorithm of Dijkstra. In 1959, a computer scientist called Edsger Dijkstra introduced the Dijkstra path planning algorithm under the title of (A Note on Two Problems in connection with Graph) [79,80]. The algorithm aims to address the complex task of determining the shortest path between a starting point and a destination within a given graph [81].

This method decodes subproblems to discover the shortest route from the starting point to the nearest vertex. It determines the nearest vertex by placing the latest vertices into a precise preference string. It also stores just a single average node; this allows for discovering the shortest and fastest trajectory. This algorithm is slow and requires lots of memory space. It will calculate all possible outcomes to determine the shortest route. The Dijkstra algorithm works best in static environments and/ or global path planning because most of the information needed to compute the shortest path is predefined [73,82].

B. A algorithm.* The A-star algorithm (A*) expands Dijkstra's method, but it has improved performance regarding time because it uses heuristics [72]. A* algorithm is part of a group of algorithms called "Graph Search Algorithms". The entire zone splits into grids. The path can be found by starting at the first point, checking adjacent squares, and then searching outwards until finding the target [83].

By using the evaluation function, it is possible to evaluate each cell of the grid; this will allow searching the shortest route by using Eq. (1) [84]:

$$F(n) = G(n) + H(n) \tag{1}$$

G(n) is the actual cost of an optimal path, and *n* is the cumulative cost to reach the current cell from the start position, known as a heuristic. *H(n)* is the Euclidian distance between *n* and the target position [85,86].

As observed in Fig. 7, a grid map indicating the cost of each straight-line motion will be 10 units. while the cost of the diagonal movement is 14. In the example in Fig. 8, by using the step cost from Fig. 7, the cost of the optimal path will be 48, where the robot will travel from the initial

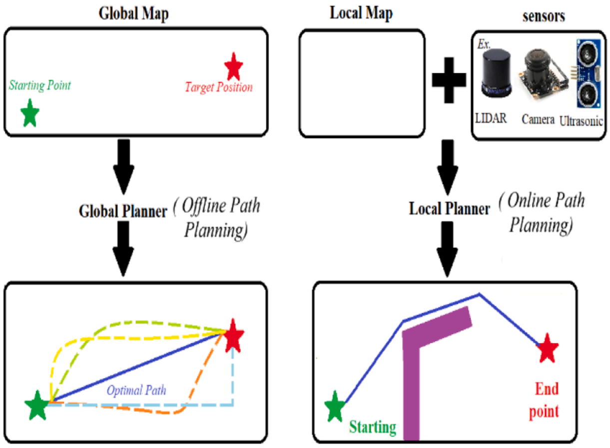

Fig. 6. Offline Path Planning VS Online Path Planning.

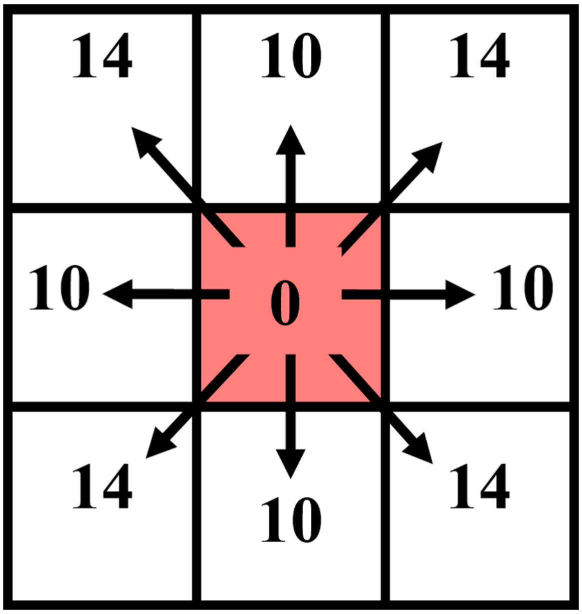

Fig. 7. Example of one step cost using A* algorithm (inspired from [72,73]).

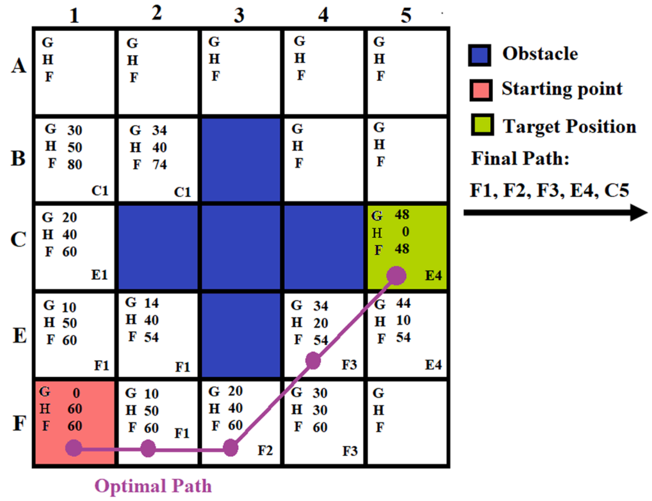

Fig. 8. A* Algorithm example (inspired from [72,73]).

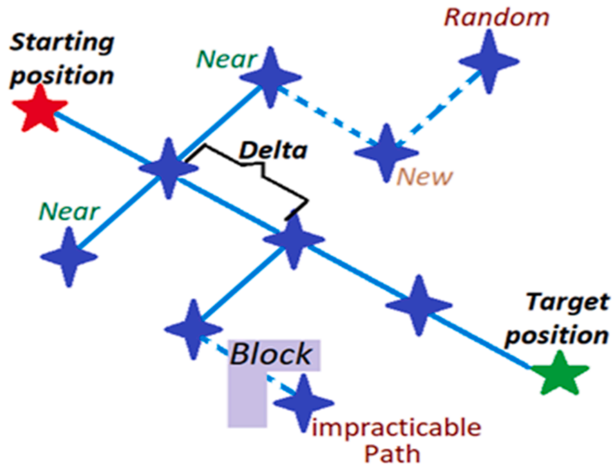

Fig. 9. RRT algorithm expansion method (inspired from [21]).

Fig. 10. RRT algorithm example.

position toward the end position by choosing the shortest path and avoiding obstacles.

C. D algorithm.* The D* algorithm (called the Dynamic A* algorithm) is similar to the A* algorithm but works in dynamic environments and is more complicated [74]. It takes longer than the average time to complete a successful run for all obstacles. D* refers to explicit policies that distinguish it from the Bug algorithm. The most significant is the prohibition against using maps in the Bug methods. D* algorithms have unique and well-designed characteristics. The motion direction for this algorithm is determined by map gridding and is based on the scoring of these maps. Iteratively, the algorithm chooses open nodes and then evaluates them. The algorithm then propagates the modifications to neighbouring nodes set on open [87,88].

D. Rapidly exploring random tree (RRT). Rapidly Random Tree exploring (RRT) is dynamically explored and does not require an explicit path to be defined. They extend in all areas and then assemble a track according to the weights allocated to each node. RRTs are algorithms that create an open-loop path for nonlinear frameworks with state constraints. The Monte-Carlo algorithm can bias search for wide Voronoi areas in a graph within a composition area [73,75,89].

This algorithm is for growing an RRT and finding a viable path. The tree starts with an initial or starting point (Starting position). Then, create a random target. In the end, a node in the tree that is closest to (Random), say (Near), is carried out using several standards (e.g., distance) [83,90].

In Fig. 9, the RRT algorithm begins at the start position (red star) and grows a tree of nodes (blue stars) through repeated random sampling of the free space. For each random sample, the algorithm locates the nearest existing node (labelled "Near") and extends a short step ("Delta") toward the sample, adding a new node if no obstacles block the path. The partial extension will be discarded if the path is blocked (as shown by the "Block"). Over multiple iterations, the tree expands in all feasible directions until one of its nodes lies sufficiently close to the goal (green star), providing a continuous path from start to goal through the connected nodes. Fig. 10 indicates an example of planning the path by selecting the optimal path towards the target.

E. Swarm intelligence. SI or Swarm intelligence is a collective behaviour that is decentralized, a self-organized system that is natural and artificial [91]. SI systems typically consist of easy agents and boids that interact locally with each other and their environment [92,93]. Motivation usually comes from nature, particularly biological systems. Although straightforward rules guide the agents, the central control structure does not exist; this dictates the agents' behaviour. However, the local and random interactions among such mechanisms result in "intelligent" global behaviour unfamiliar to individual agents [94].

F. PRM (probabilistic route map). PRM method searches for paths between randomly selected points. Collisions do not affect the workspaces containing the distributed samples' vertices; this is an undirected graph or path map construction that uses free space. The obstacles must be cleared from the randomly chosen points (vertices). A graphical search algorithm, such as A* or Dijkstra, can be used to find the best path to connect the source and destination configuration points; this is called an edge [95,96].

2.3.2. *Local algorithms details*

Local path planning, also called online path planning, uses information from sensors to assess the condition of the surroundings and data from sensors to plan the path. It aims to plan the path dynamically without collision. The traditional methods for planning local paths include spatial search, the Dynamic-Window Approach (DWA), artificial pot field methods, fuzzy logic methods, and so on [97].

2.3.2.1. APF (artificial potential field). In 1986, Khatib proposed the APF algorithm as an online planning technique [98]. This algorithm was established based on the exchange of electrostatic particles. The target point pulls the robot while the obstacle pushes it. The length between the goal and the obstacle specifies the pulling $(\overrightarrow{F}_{\overrightarrow{target}})$ or pushing

($\vec{F}_{\vec{Obstacle}}$) force rate, as shown in Eq. (2.) [99].

$$\vec{F}^{\vec{}} = \vec{F}_{\vec{target}} + \vec{F}_{\vec{Obstacle}} \tag{2}$$

Aljassani et al. [100] introduce an Enhanced Multi-Agent Formation and Obstacle Avoidance (EMAFOA) algorithm designed to address formation control and collision-free navigation for first-order, second--order, and non-holonomic multi-agent systems. Building on a leader-follower framework, the authors embed the desired formation into the system model and propose a novel artificial potential function with tunable parameters that provide high repulsive forces to prevent collisions while remaining computationally efficient. They prove the stability of their approach through Riccati-based analysis and Lyapunov theory, showing that agents can asymptotically follow the leader's trajectory and maintain the prescribed formation without succumbing to local minima or oscillations. Multiple simulation scenarios from a single leader-follower pair to twenty-agent teams demonstrate the method's agility and effectiveness in simple and complex environments. Furthermore, Monte Carlo analyses confirm the robustness and parameter sensitivity of EMAFOA, highlighting its versatility for real-world multi-agent applications.

2.3.2.2. DWA. This method is for mobile robots with synchro drives. It is based on the robot's motion dynamics [76]. Unlike other avoidance methods, the dynamic window approach is directly derived from the robot's dynamics and was created to address the limitations mandatory by its finite speeds and accelerations. It has two components: first, it generates a searching area, and then, it selects the optimal answer from that searching zone [101].

DWA produces a path nominee through fixed velocities over time. Path candidates can be either straight lines or arc paths. These path candidates do not consider obstacles and are, therefore, unsuitable for narrow spaces or dynamic environments; this means that numerous suggested trails with crashes can be produced in narrow spaces and dynamic environments [102].

2.3.3. Discussion

Researchers usually use two improvement methods for path planning: improving or combining the existing algorithms. In [103], Xing et al. proposed a new algorithm called SBREA*, which enhances the traditional A*. The new method was compared with traditional A*, an improved A*algorithm [104] called Turning weight A* (TWA*), and Rectangle Expansion A* (REA*) [105] algorithms in metrics such as search time, path length, corner reduction, and safety, especially in complex environments. The algorithm reduces search nodes by over 80 %, turning angles by over 70 %, and improves overall robot navigation efficiency and stability.

Guo et al. provide an example of merging two path-planning algorithms to increase the search efficiency and to prevent falling into local optimization [106]; they present a fusion approach between the A* algorithm, which is categorized as a global algorithm, and DWA, which is a local algorithm. Karur et al. offer an overview of mobile robot path-planning algorithms [107]; they categorize algorithms based on their suitability for static or dynamic environments and discuss foundational methods such as Dijkstra, A, D, RRT, Genetic Algorithms, Ant Colony Optimization, and Firefly Algorithm, along with their variants.

In [108], Jie et al. present a comprehensive survey of modern global path-planning algorithms for mobile robots in various environments. The authors categorize algorithms into five main groups: grid-based (Dijkstra, A*, Theta*), potential field methods (e.g., APF, harmonic fields), sampling-based (RRT, PRM), optimization-based (trajectory optimization, MILP), and hybrid approaches (bidirectional search, jump point search). The analysis of each algorithm is based on its principles, strengths, weaknesses, and application scenarios. In contrast, Lixing et al [71] present a comprehensive overview of mobile robots' global and local path-planning methods.

Dijkstra algorithm provides high accuracy with an optimal path but has a high computational cost due to exhaustive searching, low adaptability suited primarily for static environments, high memory usage, medium path smoothness, and is easy to implement, making it complete and optimal but slow on large maps. A* offers high accuracy when paired with a good heuristic, has medium computational cost, moderate memory usage, and moderate adaptability that might require replanning in static scenarios; it also features medium to high path smoothness and moderate implementation complexity, known for heuristic guidance and flexibility. The D* algorithm maintains high accuracy through heuristics and has a lower computational cost thanks to incremental updates; it is highly adaptable to dynamic environments, uses medium memory, and provides medium-to-high path smoothness while being moderately easy to implement due to its dynamic replanning capability. The Rapidly-exploring Random Tree (RRT) algorithm has medium accuracy (non-optimal), very low computational and memory usage, excellent adaptability for handling complex spaces, though it produces non-smooth paths, and is very simple to implement, making it ideal for fast exploration and scalable to high dimensions. Swarm Intelligence (SI) algorithms have high accuracy with proper tuning, moderate-to-high computational costs, low-to-moderate adaptability, medium memory usage, moderate path smoothness, and moderate implementation difficulty; they are decentralized, robust to environmental changes, and suitable for multi-robot systems. Dynamic-Window Approach (DWA) algorithms have medium accuracy (locally optimal), low computational cost suited for real-time scenarios, high adaptability for dynamic avoidance, low memory usage, medium path smoothness, and moderate implementation complexity, particularly valuable for real-time robot dynamics and velocity-space search optimization. Artificial Potential Fields (APF) deliver medium accuracy using heuristic methods, very low computational and memory requirements, medium adaptability (though prone to local-minima), and low-to-medium path smoothness, being extremely simple and fast to implement for real-time navigation despite local minima issues. Probabilistic Roadmap (PRM) provides high accuracy in known maps with a high preprocessing computational cost, low adaptability limited to static environments, high memory usage, and excellent path smoothness; it is moderately easy to implement, efficient for multiple queries, and roadmap-based, though slow in updates. The key features of the selected algorithms in this paper are determined in Table 2, focusing on the accuracy of each method and path smoothness with the range of 1 = poor and 5 = excellent, computational cost and memory usage where in the range of 1 = very high and 5 = very low, adaptability is 1 = very low, 5 = very high, and implementation 1 = difficult, 5 = very easy.

2.3.4. Related works

Different research has extensively explored various algorithms for robotic path planning. Initial studies applied fuzzy logic [109] and modified Artificial Potential Field (APF) algorithms [110], focusing on local planning through simulations and physical robots to enhance navigation and obstacle avoidance. A significant advancement included the adaptation of the A-Star (A*) algorithm for global applications such as vessel path planning [111] and improvements to APF-based methods enhanced by augmented reality and neural dynamics [99,112]. Concurrently, theoretical research emphasized the use of Deep Q-Networks (DQN) [113,114] and the Dynamic Window Approach (DWA) combined with virtual manipulators for improving navigation algorithms [102]. In 2022, experiments incorporated various sophisticated algorithms such as Bat algorithms (BA) [115], Ant Colony Optimization (ACO), and Improved Coyote Optimization Algorithms [116] to optimize path efficiency, robustness, and swarm intelligence capabilities.

Recent studies expanded the scope further, introducing Enhanced Multi-Agent Systems Formation and Obstacle Avoidance (EMAFOA) [100], motion capture-enhanced Probabilistic Roadmap Methods (PRM) [117], Non-dominated Sorting Genetic Algorithm II (NSGA-II) [118], improved sparrow search algorithms [119], and deep reinforcement

Table 2
Path planning algorithm's key features.

Algorithm	Accuracy	Computational Cost	Adaptability	Memory Usage	Path Smoothness	Implement.	Key Features
Dijkstra	5	1	1	1	3	5	Complete, optimal, and slow on large maps
A*	5	3	2	3	4	3	Heuristic-guided, faster than Dijkstra, flexible
D*	5	4	5	3	4	3	Dynamic replanning, efficient in changing environments
RRT	3	5	5	5	1	5	Fast exploration, scalable to high dimensions, randomness-based
SI	5	2	2	3	3	3	Decentralized, robust to environmental changes. Suitable for multi-robot systems and scalable.
DWA	3	5	5	5	3	3	Considers robot dynamics, local optimization, and velocity-space search
APF	3	5	3	5	2	5	Simple, fast, can get stuck in local-minima, real-time navigation
PRM	5	1	1	1	5	3	Efficient for multiple queries, slow to update, and roadmap-based planning

learning (DRL) [120], highlighting their practical effectiveness through real-time robotic platform testing and extensive simulation validation. Table 3 indicates the related works regarding the path-planning algorithms according to the type of research and the algorithm name, and method. Also, it describes whether the testing method is simulated or a real scenario. Finally, in the last column, the aim of each publication will be indicated.

While the core components of a mobile robot navigation system such as mapping, localization, and path planning—can operate independently or in sequence within predefined and static environments, their integration is crucial for achieving robust autonomy in more complex, dynamic, or unknown settings. The next section explores how these tasks are combined into cohesive frameworks that enable intelligent, context-aware navigation.

3. Combination of navigation tasks

Localization, mapping, and path planning are the primary duties of mobile robot navigation, and combining two navigation tasks will introduce a different classification, as shown in Fig. 11. The first combination is between the localization problem and mapping. Sometimes, this algorithm is called a "chicken or egg problem" because it requires a good map for localization and a precise pose estimate to build a map [20, 129]. In the case of merging mapping and path planning, this will offer the exploration task, in which the robot assumes precise pose information and focuses on guiding it through the environment efficiently to create a map. Seeking to guide the robot to locations within the map to improve the exact pose estimate is the definition of active localization, which is the task indicated due to the overlap between localization and path planning [130].

3.1. Simultaneous localization and mapping (SLAM)

SLAM is a complicated problem that must be solved [20,131]. The primary concept of SLAM is to allow the mobile robot to construct its bounding environment map and localize its location on the map at the same time (basically to build a map and to know its location on that map) [132,133].

Initially, the robot's location and map are unknown in the SLAM problem. The mobile robot uses a recognized kinematic model and drives in an unidentified environment populated by artificial or natural landmarks. Based on observations of landmarks, simultaneous estimations of robot and landmark locations are made. The SLAM problem is finding the proper representation for observation and motion models [134,135].

Several research papers and publications introduced the filters to solve the problem of uncertainty and noise for both SLAM and Visual-SLAM methods [56,129,133,136–138]. Kalman filter-based methods such as Unscented KF (UKF) (2000), Adaptive KF (AKF), and Extended KF (EKF) (2001) [139] are some of the most popular and oldest ways to solve the SLAM issue [133]. A Kalman filter allows the creation of a system model that is observable from the internal state. Then, the model and real-world system outputs are compared to find the correct state [134].

The working principle of the SLAM method is to collect data from the environment through sensors. Furthermore, from this point of view, the collected data can be visual, such as the data collected by the camera. In this case, a vSLAM is introduced, or it can be non-visual data, such as the data collected by the LIDAR sensor.

3.1.1. Related work

This review compiles different studies of advancements in Simultaneous Localization and Mapping (SLAM) techniques in both static and dynamic environments. Example in [140] Wang et al. demonstrated the integration of SLAM and Detection and Tracking of Moving Objects (DATMO) through real-world experiments in outdoor settings using the Navlab11 platform. Over the years, approaches have evolved to include active SLAM with Extended Kalman Filters (EKF) [141,142] neural networks with dense optical flow approach [143], and parallel processing using ORB-SLAM3 with dynamic point filtering [144], the latter improving accuracy and computational efficiency in mixed indoor-outdoor environments. Several studies emphasized real-time tracking, feature extraction, and sensor fusion. More recent contributions propose increasingly sophisticated models such as Graph-SLAM with Kalman filtering [148], Deep Features in Dynamic Environment (DFD-SLAM), integrating feature and target recognition in dynamic settings [149], ULG-SLAM utilizing unsupervised learning geometric-based SLAM for improved trajectory accuracy [150], and GY-SLAM combining YOLOv5 with GhostNet to boost localization performance in complex environments like plant factories [151]. Collectively, these studies highlight the progression from traditional filter-based SLAM methods to modern deep learning-enhanced and multi-sensor SLAM frameworks, capable of operating robustly in varied and dynamic conditions. In Table 4, different projects and studies were introduced as examples of SLAM and VSLAM approaches with different filtering methods in indoor and outdoor environments. The table also indicates the testing method and the tasks assigned to each project.

3.2. Active localization

Mobile robot localization involves determining a mobile robot's location using sensor data [152]. Active localization guides the robot to specific locations on the map [20]. Active localization can sometimes produce an error between the actual and estimated positions of the mobile robot because the sensor data has more information than is needed (noisy measurements). Filters such as Kalman and particle filters solve this problem. Filters will increase the accuracy, computational capability, and memory available at a sensor node [51].

Table 3
Examples of applying path planning algorithms.

Ref. No.	Year	Type of Research	Algorithm name	method type	Testing method	Publication aims
[109]	2016	Experimental research	Fuzzy logic	Local	Simulation and implementation on I-Robot with 3 ultrasonic sensors.	Articulate robotic path planning using fuzzy logic.
[110]	2017	Experimental research	modified APF	Local	Simulation (MATLAB)	Removes any obstructions in the robot environment
[111]	2019	Experimental research	A-Star (A*)	Global	Simulation and real scenarios	Improving vessel path planning
[112]	2019	Experimental research	APF	Local	Simulation (MATLAB)	Modify the APF algorithm for a quadrotor.
[99]	2021	Experimental research	AR-APF	Local	Tests performed on ROSbot 2.0 PRO using (lidar)	APF supported by augmented reality to bypass the upcoming local minimum
[121]	2021	Experimental research	CFD-based neural dynamic	Local	Simulations and experiment	Improved neural dynamics-based approach with a territorial mechanism for online path planning of MRS
[113]	2021	Experimental research	Q-learning algorithm	Global	Simulation	To find multiple feasible paths in obstacle environments using Q-learning, aiming at efficient multi-path planning
[114]	2021	Theoretical research	Deep Q-Network algorithm (DQN)	Global	Simulation results and analysis	Improving the DQN path planning algorithm.
[102]	2022	Theoretical research	DWV	Local	Simulation and hardware (Laser rangefinder)	Combination of (DWA)+ virtual manipulator (VM).
[115]	2022	Experimental research	Bat algorithm	Global	Simulation experiments in various environments	To enhance path planning robustness using a reformative bat algorithm.
[122]	2022	Theoretical research	A-Star (A*) algorithm	Global	Simulation	Improving A* algorithm for intelligent vehicle path planning.
[123]	2022	Experimental research	Improved Neural Dynamics	Local	Real experiment	Improving online path planning and tracking.
[124]	2022	Theoretical research	Ant colony optimization (ACO)	Global	Simulation results and analysis	To evaluate swarm intelligence algorithms in drone path planning
[116]	2022	Experimental research	Improved Coyote Optimization Algorithm	Global	Simulation on four maps for mobile robot path planning	Propose an improved algorithm for mobile robot path planning, focusing on global search capability, convergence speed, and stability
[100]	2023	Simulation-based study	Enhanced Multi-Agent Systems Formation and Obstacle Avoidance (EMAFOA)	Local	Multiple simulation scenarios (complex/straightforward obstacles, varying formations) and Monte Carlo analysis	Proposes and validates a general, computationally efficient multi-agent formation and obstacle avoidance algorithm capable of first/second-order and non-holonomic systems, avoiding local-minima while maintaining formation stability.
[117]	2024	Experimental research	PRM Algorithm	Global	Simulation using motion capture	Utilize motion capture technology to improve PRM path planning, enhance trajectory accuracy, and optimize navigation in dynamic environments.
[118]	2024	Experimental Research	Non-dominated Sorting Genetic Algorithm II	Global & Local	Simulation (MATLAB)	NSGA-II-based method to address local optima and enhance path smoothness using Bézier curves
[125]	2024	Experimental research	Fuzzy Logic Control (FLC)	Local	Actual experiments under varying light intensities	Develop lane-keeping assist for AGV based on computer vision and FLC to optimize path-tracking stability under varying lighting conditions.
[119]	2024	Experimental research	improved sparrow search algorithm	Global & Local	Simulation (Improved on different maps)	Enhancing SSA for mobile robot path planning, optimizing path length, running time, and path stability using advanced strategies like Lévy flight and adaptive mutations.
[126]	2024	Experimental Research	A*, RRT, PRM	Global	Real-time testing with Turtlebot2i	Evaluate the performance of heuristic algorithms like A*, RRT, and PRM in dynamic real-time scenarios for mobile robot path planning using vision-based feedback.
[120]	2024	Experimental research	Deep Reinforcement Learning	Local	Real robot platform testing	Demonstrate the application of a Deep Reinforcement Learning (DRL)-based algorithm and a two-way search Hybrid A* algorithm in mobile robot path planning
[127]	2024	Experimental Research	Improved ant colony optimization	Global & Local	Simulation (Improved on grid-based map model	Improve the ACO algorithm to optimize robot path planning in terms of convergence, avoidance of local optima, and path smoothness.
[128]	2024	Experimental Research	A*, IPSO, IGWO, and ABC	Global	simulation and real-world experiments	Identify the most suitable metaheuristic approach for efficient path planning of autonomous mobile robots in indoor environments.

3.2.1. *Kalman filter*

In 1960, R. E. KALMAN introduced a novel method for addressing issues associated with linear dynamic systems used for filtering and prediction in random noise [153]. Kalman filter is a set of mathematical equations that offers an efficient way to estimate the state of a process [154], and it does not use an arbitrary density function [56]. Sensor fusion uses the Kalman or Extended Kalman filters for linear and nonlinear systems [155]. It is also the simplest case of filtering problems because the measurement and system models are linear functions, and both noise models in the system description are Gaussian [51,156].

In the context of localization, Kalman filters have several types [157], such as:

3.2.1.1. Extended Kalman filter (EKF). In the real world, the systems are dynamic, thus they are nonlinear. For that reason, the Kalman filter will not be effective. The Extended Kalman Filter (EKF) extends the classic Kalman filter for handling nonlinear state models. It approximates the state transition and observation models that are nonlinear by using a

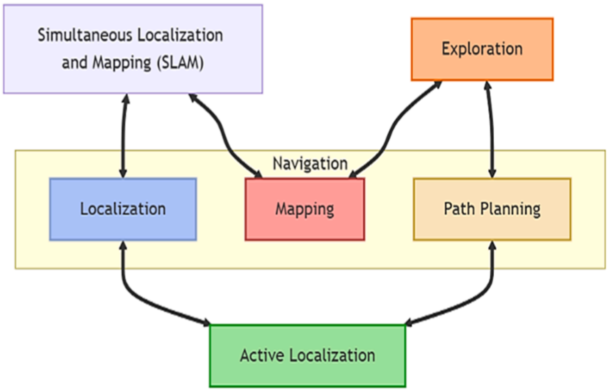

Fig. 11. Illustration of the overlapping (combination) of the mobile robot primary navigation approach (inspired from [7]).

first-order Taylor series expansion [156,158]

3.2.1.2. Unscented Kalman filter (UKF). The Unscented Kalman Filter (UKF) is a method used in signal processing to optimally estimate the internal states of a nonlinear dynamic system from a series of noisy measurements. The UKF addresses some of the limitations of the Extended Kalman Filter (EKF), particularly the problem of linearizing highly nonlinear functions, which can lead to significant approximation errors [159].

UKF's design principle balances computational complexity and estimation accuracy trade-offs. By avoiding Jacobian calculations and directly using nonlinear models, the UKF simplifies the implementation and potentially increases the fidelity of state estimation in complex systems. However, managing and propagating sigma points through these nonlinear functions remains computationally demanding, particularly as the system's complexity increases [160].

3.2.1.3. Ensemble Kalman filter (EnKF). The Ensemble Kalman Filter is a Monte Carlo-based Kalman filter (KF) implementation. It represents a critical advancement in filtering technology. Its capacity to handle vast dimensional spaces and adapt to natural systems' intrinsic uncertainties and complexities makes it invaluable. As computational resources continue to grow, the applications of EnKF are likely to expand, further solidifying its role in critical decision-making processes in environmental sciences and beyond [161].

3.2.2. Particle filters

Sequential Monte Carlo techniques are a collection of Monte Carlo algorithms that can be used to solve problems in signal processing and Bayesian statistical Inference. Particle filters are a non-parametric variation of the localization filter. They use multiple particle hypotheses to estimate the state. This flexible solution is suitable for tracking nonlinear systems with non-Gaussian measurement errors. Compared with the Kalman filter, the particle filter is often more suitable because it can be used in real-world situations that have nonlinearities or non-Gaussian errors [33,35–37,162].

3.2.3. Related work

Table 5 focuses on presenting projects with different filtering methods for active localization. The testing methods and the type of environment are considerable in this table, and the approach for each research is indicated in the last column.

3.3. Exploration

The contrast between active localization is a combination of two algorithms: path planning and mapping. Exploration methods assume precise pose information and are focused on efficiently guiding the robot through the environment to create a map [20]. Exploration can be defined as discovering a new thing, and in navigation, it is discovering remote lands.

Exploration is challenging because it can be hard to choose the

Table 4
Examples of projects and research that used SLAM/ VSLAM algorithms.

Ref. No.	Year	Type of Study	Approach	Testing method	Type of environment	Achieved Task
[140]	2003	Conference Paper	EKF for SLAM & DATMO	Real experiment using Navlab11	Outdoor environment	Integration of SLAM and DATMO in a dynamic environment.
[141]	2008	Conference Paper	active SLAM method with an EKF	Simulation system by MATLAB	/	Comparing three active SLAM algorithms
[142]	2013	conference paper	Extend Kalman Filter (EKF)	A simulated and real-world experiment	indoor environment	Solving the SLAM problem.
[143]	2022	Research Article	neural network and dense optical flow algorithm	Simulation in the OSCER environment	Outdoor environment	Localizing, mapping, and tracking dynamic objects in real-time.
[144]	2022	Academic Article	Parallel processing (ORB-SLAM3+ dynamic point filtering)	Real-world experiment using the TUM dataset	An indoor and outdoor environment	Improved accuracy and time efficiency in dynamic scenes
[145]	2022	Conference Paper	TLS and Sensing (LiDAR)	Real-world experiment	Outdoor environment	Correcting SLAM LiDAR distortions for outdoor levee monitoring
[146]	2022	Research Article	UWB ranging and odometry measurement	Real experiment and analysis	Indoor experiments with three robots	Distributed SLAM for accuracy improvement
[147]	2024	Journal Article	Extended Kalman Filter (EKF), AMCL, and Gmapping algorithm	Indoor environment	Indoor environment	Improved accuracy in indoor localization using multi-sensor fusion
[148]	2024	Journal Article	Kalman Filter and GraphSLAM	/	Various environments	Developing robust, accurate SLAM algorithms for robot localization and navigation in unknown environments.
[149]	2024	Research Article	DFD-SLAM with feature and target recognition	Simulated environments	Dynamic environments	Deep-featured VSLAM in dynamic environments, better target recognition, and mapping.
[150]	2024	Journal Article	ULG-SLAM with unsupervised learning & LSD-based features	Multiple indoor experiments using EuRoC and TUM RGB-D datasets	Indoor environment	Enhanced localizability estimation, improved trajectory accuracy, and robustness in SLAM.
[151]	2024	Research Article	GY-SLAM with YOLOv5, GhostNet	Proprietary dataset, TUM dataset	Dynamic plant factory	Significant improvements in RMSE for ATE and better detection with localization in dynamic environments.

Table 5
Projects and research used localization with different types of filters.

Ref. No.	Year	Filtering	Testing method	Type of environment	Approach
[163]	2009	Particle filter	Simulation and real experiment	office building environment	landmark sensing algorithm
[164]	2014	Particle filter	Real experiment and analysis	Indoor environment	The active global localization algorithm
[165]	2019	Integrated Bayesian filter	Simulation, real-world experiments, and analysis	Indoor environment	A Learning-based approach
[166]	2021	Monte Carlo particle filter	real experiment and analysis	Outdoor environment	Curve-localizability-SVM active localization algorithm
[167]	2022	Particle filter	Simulation and real experiment	Indoor environment	clustering algorithm
[168]	2024	Unscented Kalman filter	Simulation experiment	Indoor environment	L-PCM (Localisation and Point Cloud Registration-Based Method)
[169]	2024	Particle filter	Real experiment and analysis	Indoor environment	3D SLAM with layout map integration using particle filter
[170]	2024	Monte Carlo particle filter	Simulation experiment	Indoor environment	OFM-IPSOMCL (Offline Feature Matching and Improved Particle Swarm Optimisation for Monte Carlo Localization)
[171]	2024	Extended Kalman filter	Simulation and real experiment	Building environment	Trilateral localization technique with multi-sensor fusion
[172]	2024	Adaptive Monte Carlo Localization (AMCL)	Simulation experiment	Indoor environment	Fusion of an improved AMCL algorithm and a collision algorithm

proper viewpoints. Many aspects are important. Two things should be considered. On the one hand, there should be little uncertainty about the current map state. On the other hand, reducing the number of measures required to overcome this uncertainty and minimizing the distance travelled should be possible. There are many ways to guide the robot in the environment [20]. Fig. 12 illustrates the strategies of mobile robot exploration, including Closest Target Location (CL), information Gain exploration (IG), a combination between IG in a Local Window (IG WIN), and a combination between IG and CL (IG CL).

- CL: This is the most popular exploration strategy. It is used to drive the robot toward the nearest location where it can collect information about a cell it has not yet explored. This strategy can provide short trajectories to single-robot exploration tasks using equations. (3.) [20].

$$c^* = \arg\ \min_{c\in L(m)} dist_m(x,c) \tag{3}$$

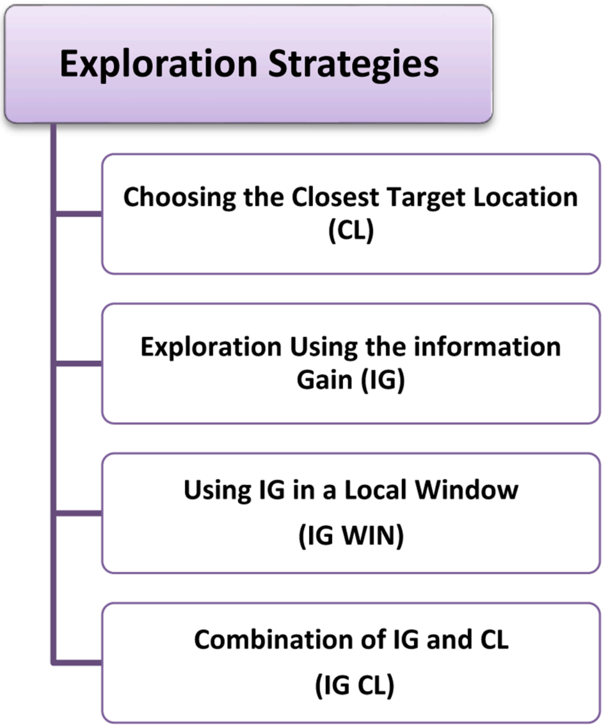

Fig. 12. Exploration strategies.

- IG: It is governed solely by information gained about the environment from a particular viewpoint. This information is used to assess the accuracy of the sensor's information. By considering each grid cell c as a potential next viewpoint and selecting the one which provides the highest expected entropy reduction Eq. (4) [20]

$$c^* = \arg\_\max_{c\in L(m)} E[I(c)] \tag{4}$$

- IG WIN: To overcome the problem of strategy IG not considering the overall length of the resulting trajectory, the strategy IG WIN was created. This technique limits the search for possible viewpoints to a local window. This window defines an area of the environment that the robot must explore before moving on to another area. The equation can determine the following viewpoint. (5.) [20]

$$c^* = \arg\_\max_{c\in L_{win}(m)} dist_m(x,c) \tag{5}$$

- IG CL: It combines the properties of strategies CL and IG. It is important to strike the right balance between expected information gain E [I (c)] from possible viewpoints L(m) c and costs dist-m (x, c).

$$c^* = \arg\_\max_{c\in L(m)} \left[\alpha \frac{E[I(c)]}{max_{c'\in L(m)} E[I(c')]} - (1-\alpha) \frac{dist_m(x,c)}{max_{c'\in L(m)} dist_m(x,c')} \right] \tag{6}$$

The integration of core navigation tasks provides a solid foundation for autonomous operation. However, static systems alone are often insufficient in unpredictable environments. To address this, recent researchers and engineers have introduced intelligent navigation strategies that allow robots to adapt their behaviour in real time by including learning algorithms, environmental awareness, and adaptive decision-making.

4. Smart navigation

The fourth industrial revolution was driven by technologies like Internet of Things (IoT), AI, Big Data (BD), cloud computing, and advanced sensor technologies [173]. These technologies significantly enhance mobile navigation algorithms by improving perception, localization, mapping, and decision-making [174]. The advanced sensors provide high-precision data, enabling AI-driven systems to navigate more accurately and autonomously in dynamic environments [174,175]. For mobile robot navigation, AI navigation systems use a combination of classical algorithms and AI/ML-powered methods to perceive, plan, and control movement in complex environments [176].

Zhang et al [177] systematically review recent progress in sensor technologies such as image, acoustic, wearable, and neuromorphic computing. The paper highlights the improvements in AI-driven sensors in several applications. On the other hand, in [178], Klančar et al. discuss the crucial role of advanced sensor technologies in enhancing localization, situational awareness, path planning, and control algorithms for mobile robots by highlighting how LIDAR, cameras, IMUs, motion capture systems, and AI-powered perception improve autonomous navigation by providing accurate environmental recognition, object detection, and tracking.

Ušinskis et al [179] propose sensor fusion techniques, integrating data from multiple sensors to enhance perception, obstacle detection, and trajectory planning. The study highlights the role of AI-driven methods in processing sensory inputs, enabling real-time decision--making and adaptive path planning for tasks such as multi-robot coordination and inspections in inaccessible areas. They underscore the role of artificial intelligence (AI) and advanced control algorithms in processing sensory data, enabling adaptive, intelligent navigation strategies for autonomous systems in complex and unpredictable scenarios.

5. Simulation environments

A simulation of a system is a way to represent the system response as a function of time and space [180]. It is defined as a programmable environment of a computer that simulates the system with flexible and intelligent interfacing between a user and the system [181]. MATLAB and Simulink are high-level interactive computing environments. They are widely used for mobile robot navigation simulations because they offer an integrated environment with robust toolboxes for modelling, control design, and visualization. MATLAB's scripting capabilities facilitate rapid prototyping and algorithm development (e.g., for path planning, sensor fusion, or control laws), while Simulink provides a block-diagram interface that makes it easier to construct and simulate dynamic systems modularly. Together, they streamline the workflow—from formulating equations of motion and control strategies to testing and tuning them in a virtual environment before deploying code to real hardware.

ROS is a meta-operating and open-source software for robots. It provides all the services expected from an operating system; this includes hardware abstraction, low-level device control, implementation of standard functionality, message passing among processes, and package management. It provides libraries and tools for building, writing, and running multiple code projects across multiple computers [182,183]. ROS is a robot operating system, but not an OS [184].

Unlike MATLAB/Simulink, which provides an all-in-one environment for modelling, simulation, and analysis, ROS serves primarily as a communication infrastructure and set of libraries that simplify message passing, device drivers, and software reuse across different robots and platforms. Developers often pair ROS with simulation tools such as Gazebo or RViz to visualize and test algorithms in virtual environments. This ecosystem makes ROS especially appealing for distributed development (where different teams or researchers can contribute independent software packages) and real-world deployment, since the same ROS-based nodes running in simulation can be deployed on physical robots with minimal changes.

6. Discussion

In [185], Zhu and Zhang review recent progress in applying deep reinforcement learning (DRL) to mobile robot navigation, emphasizing how DRL methods can overcome the limitations of classical navigation frameworks. It begins by introducing fundamental DRL concepts — value-based and policy-based algorithms — and then reviews how DRL-based navigation systems replace or integrate with traditional pipelines for local obstacle avoidance, indoor navigation, multi-robot coordination, and social navigation. Key challenges include partial observability, sparse rewards, and poor generalization across varied or real-world scenarios. The authors discuss potential solutions to improve training efficiency and transferability, such as incorporating memory (RNN/LSTM), reward shaping, auxiliary tasks, curriculum learning, and domain randomization. They conclude that although DRL policies do not always yield perfect behaviours, DRL remains promising for achieving robust and flexible navigation capabilities in increasingly complex and dynamic environments.

In [186], Loganathan and Ahmad comprehensively review path-planning techniques for Autonomous Mobile Robots (AMRs), particularly in handling dynamic or partially unknown environments. It systematically categorizes the methods into classical (e.g., cell decomposition, roadmap, RRT, APF) and heuristic (e.g., fuzzy logic, neural networks, GA, PSO, ACO, and newer bio-inspired algorithms). Each approach is evaluated in terms of its fundamental concepts, strengths, and shortcomings, highlighting key research challenges such as local minima, high computational cost, and adaptation to real-time changes. The review also discusses practical implementation issues—like kinematic constraints and multi-objective optimization—and offers guidance for selecting or combining algorithms. Concluding remarks identify research trends (e.g., hybridizing algorithms and incorporating machine learning) that can further improve the autonomy and robustness of AMRs in increasingly complex scenarios. Other reviewing papers, such as [187,188] and [189], review the topic of path-planning algorithms from different aspects.

In contrast, a considerable amount of research focuses on localization, as introduced in Subsection 2.1, and mapping, as indicated in Subsection 2.2. individually. Mobile robot navigation faces several technical, physical, and computational challenges, especially in real-world environments. These challenges impact a robot's ability to perceive, plan, and act effectively in structured and unstructured settings. In localization, determining the accurate position of the robot in a map or an unknown environment is challenging due to the sensor noises, moving objects, and featureless areas. While in path planning, finding the optimal paths in environments with moving obstacles will require real-time re-planning while considering optimality and safety.

As mentioned in Section 3.1, SLAM is a problem of building a map and determining the location of the robot, which can lead to recognizing previously visited places, which is called (Loop closure detection errors); also, it requires a high memory/computation capability in large-scale environments. Over time, the accumulation of errors can lead to the drifting of the map.

Regarding sensor fusion, data from multiple sources is processed efficiently, often facing synchronization and computational constraints. Robots also encounter difficulties in unstructured terrains, resource constraints (battery and processing power), and real-time decision-making under uncertainty. Ensuring safe human-robot interaction and designing scalable, adaptable systems remain key challenges. Advanced AI, deep learning, and improved sensor fusion techniques are crucial for overcoming these limitations and enabling more autonomous and reliable navigation.

This research aims to cover the three navigation problems and their overlapping because it is limited to finding review papers that cover the general idea of the mobile robot navigation approach.

7. Conclusion

This research, undertaken in this paper, underscores the essential components and challenges associated with the development of autonomous mobile robots. Explore the intricacies of robot navigation systems, mainly focusing on the critical tasks of mapping, localization, and path planning. These elements are fundamental to achieving robot autonomy, allowing them to operate effectively in diverse environments, from industrial settings to delicate medical facilities. As illustrated in Fig. 13, a layered navigation architecture in which sensor and actuator hardware feed raw data to a data-interaction layer, whose fused outputs enter a SLAM module to build a map and estimate robot pose; a path-planning block then combines this map, pose, and a target goal to generate an optimal trajectory, which is visualized in simulation or executed on the real platform via low-level control commands sent back down to the hardware.

Our investigation highlights that despite the advancements in robotics, significant challenges remain, particularly in dynamic and unpredictable environments where moving obstacles and emergencies such as fires can drastically affect robot behaviour. The adaptability of mobile robots in such conditions is paramount and requires continuous enhancement of object detection algorithms, which must robustly handle issues like deformations, occlusions, and variable lighting conditions.

Furthermore, this paper has demonstrated the fruitful applications of deep camera technologies and advanced path-planning algorithms to ensure that robots navigate efficiently and recognize and interact with their surroundings reliably. Integrating these technologies facilitates a higher degree of robot autonomy, which is crucial for tasks that are either hazardous or inaccessible to humans.

Looking ahead, the field of mobile robotics holds tremendous potential for further research and development. The continuous evolution of autonomous navigation technologies and the integration of more sophisticated sensory and computational capabilities will undoubtedly expand the scope and effectiveness of robots across various sectors. This ongoing progression will enhance operational efficiencies and contribute significantly to safety and accessibility in human environments.

Therefore, the collective efforts in robotics research, as surveyed will allow the robot to enhance the capabilities of autonomous robots but and paving the way for innovative applications that will continue to revolutionize multiple aspects of society.

This paper represents the first stage of studying and developing an optimal navigation approach for social assistive mobile robots used in eldercare facilities. There is still a pressing need for a navigation approach that operates in real-time, responds to changes in the physical environment with minimal delay, and enables robots to exchange data, coordinate actions, learn from each other, and optimize their behaviour over time.

CRediT authorship contribution statement

Mayar Abdullah Taleb: Writing – review & editing, Writing – original draft, Visualization, Validation, Methodology, Investigation, Conceptualization. **Gyula Korsoveczki:** Writing – review & editing, Validation, Supervision, Resources, Investigation, Formal analysis, Conceptualization. **Géza Husi:** Writing – review & editing, Supervision, Project administration, Funding acquisition, Conceptualization.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

Mayar Abdullah Taleb reports article publishing charges was provided by University of Debrecen Faculty of Engineering. Husi geza reports a relationship with University of Debrecen Faculty of Engineering that includes: consulting or advisory and employment. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

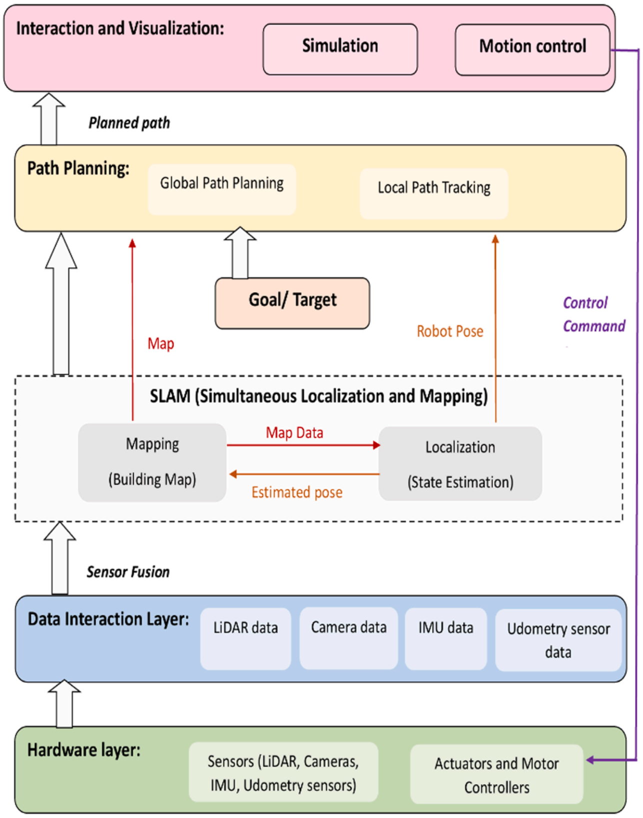

Fig. 13. layered architecture diagram of mobile robot navigation.

Data availability

No data was used for the research described in the article.

References

[1] ISO 8373, Robots and robotic devices — Vocabulary n.d. https://www.iso.org/obp/ui/#iso:std:iso:8373:ed-2:v1:en, 2012.

[2] D. Mihai, Standards and classifications of industrial robots • smashing robotics, Smashing Robot (2012). https://www.smashingrobotics.com/standards-and-classifications-of-industrial-robots/.

[3] Wheeled Mobile Robot - an overview | ScienceDirect topics n.d. https://www.sciencedirect.com/topics/engineering/wheeled-mobile-robot.

[4] Tenreiro Machado J., Silva M.. An overview of legged robots. 2006.

[5] Tracks military technology, Tracked robots uses, advantages & disadvantages | science online. https://www.online-sciences.com/robotics/tracks-military-technology-tracked-robots-uses-advantages-disadvantages/, 2019.

[6] M. Bjelonic, P. Sankar, D. Bellicoso, H. Vallery, M. Hutter, Rolling in the deep – Hybrid locomotion for wheeled-legged robots using online trajectory optimization, IEEe Robot. Autom. Lett. (2020), https://doi.org/10.1109/LRA.2020.2979661.

[7] Robotics IIF of. Record of 4 Million Robots in Factories Worldwide. IFR Int Fed Robot n.d. https://ifr.org/ifr-press-releases/news/record-of-4-million-robots-working-in-factories-worldwide.

[8] M. Ben-Ari, F. Mondada, Robots and their applications, in: M. Ben-Ari, F. Mondada (Eds.), Elem. Robot., editors, Springer International Publishing, Cham, 2018, pp. 1–20, https://doi.org/10.1007/978-3-319-62533-1_1.

[9] E. Petraki, D. Herath, Teaching and learning robotics: A pedagogical perspective, in: D. Herath, D. St-Onge (Eds.), Found. Robot. Multidiscip. Approach Python ROS, editors, Springer Nature, Singapore, 2022, pp. 43–62, https://doi.org/10.1007/978-981-19-1983-1_2.

[10] M. Mihelj, T. Bajd, A. Ude, J. Lenarčič, A. Stanovnik, M. Munih, J. Rejc, S. Šlajpah, Introduction, in: M. Mihelj, T. Bajd, A. Ude, J. Lenarčič, A. Stanovnik, M. Munih, J. Rejc, S. Šlajpah (Eds.), Robotics, editors, Springer International Publishing, Cham, 2019, pp. 1–9, https://doi.org/10.1007/978-3-319-72911-4_1.

[11] U.S.P. Srinivas Aditya, R. Singh, P.K. Singh, A. Kalla, A survey on blockchain in robotics: issues, opportunities, challenges and future directions, J. Netw. Comput. Appl. 196 (2021) 103245, https://doi.org/10.1016/j.jnca.2021.103245.

[12] V. Gusan, M.A. Țîțu, C. Oprean, Industrial robots versus collaborative robots - the place and role in nonconventional technologies, ACTA Tech. Napoc. - Ser. Appl. Math. Mech. Eng. (2022) 65.

[13] G. Singh, V.K. Banga, Robots and its types for industrial applications, Mater. Today Proc. 60 (2022) 1779–1786, https://doi.org/10.1016/j.matpr.2021.12.426.

[14] M.S. Elhadidy, W.S. Abdalla, A.A. Abdelrahman, S. Elnaggar, M. Elhosseini, M. S. Elhadidy, W.S. Abdalla, A.A. Abdelrahman, S. Elnaggar, M. Elhosseini, Assessing the accuracy and efficiency of kinematic analysis tools for six-DOF industrial manipulators: the KUKA robot case study, AIMS Math. 9 (2024) 13944–13979, https://doi.org/10.3934/math.2024678.

[15] Z. Jiawei, Robot Kinematics: Motion, Kinematics Dyn. (2022), https://doi.org/10.48550/arXiv.2211.15093.

[16] Shabalina K., Sagitov A., Magid E. Comparative analysis of mobile robot wheels design. 2018. https://doi.org/10.1109/DeSE.2018.00041.

[17] Introduction to Autonomous Mobile Robots. MIT Press n.d. https://mitpress.mit.edu/9780262015356/introduction-to-autonomous-mobile-robots/.

[18] U. Jahn, D. Heß, M. Stampa, A. Sutorma, C. Röhrig, P. Schulz, C. Wolff, A taxonomy for mobile robots: types, applications, capabilities, implementations, requirements, and challenges, Robotics 9 (2020) 109, https://doi.org/10.3390/robotics9040109.

[19] Ivanjko E., Petrini T., Petrovic I. Modelling of mobile robot dynamics 2010.

[20] Stachniss C. Robotic mapping and exploration. vol. 55. 2009. https://doi.org/10.1007/978-3-642-01097-2.

[21] Y. Liu, S. Wang, Y. Xie, T. Xiong, M. Wu, A review of sensing technologies for indoor autonomous mobile robots, Sensors 24 (2024) 1222, https://doi.org/10.3390/s24041222.

[22] M. Mallik, A.K. Panja, C. Chowdhury, Paving the way with machine learning for seamless indoor–outdoor positioning: a survey, Inf. Fusion. 94 (2023) 126–151, https://doi.org/10.1016/j.inffus.2023.01.023.

[23] T. Shan, E.B. Lego-Loam, Lightweight and ground-optimized lidar odometry and mapping on variable terrain, in: 2018 IEEERSJ Int. Conf. Intell. Robots Syst. IROS, 2018, pp. 4758–4765, https://doi.org/10.1109/IROS.2018.8594299.

[24] A review of mobile robots: concepts, methods, theoretical framework, and applications - Francisco Rubio, Francisco Valero, Carlos Llopis-Albert, n.d, https://journals.sagepub.com/doi/10.1177/1729881419839596, 2019.

[25] H. Shin, S.H. Jung, Y.R. Choi, C. Kim, Development of a shared remote control robot for aerial work in nuclear power plants, Nucl. Eng. Technol. 50 (2018) 613–618, https://doi.org/10.1016/j.net.2018.03.006.

[26] A. Yadav, A. Prakash, A. Kumar, S. Roy, Design of remote-controlled land mine detection troops safety robot, Mater. Today Proc. 56 (2022) 274–277, https://doi.org/10.1016/j.matpr.2022.01.128.

[27] D. Chikurtev, I. Rangelov, K. Yovchev, N. Chivarov, Communication system for remote control of service robots, IFAC-Pap 52 (2019) 186–191, https://doi.org/10.1016/j.ifacol.2019.12.470.

[28] I. Nizar, M. Mestari, Mobile robot autonomous navigation: a path planning approach, IFAC-Pap 55 (2022) 610–615, https://doi.org/10.1016/j.ifacol.2022.07.379.

[29] B. Xu, D. Liu, M. Xue, M. Miao, C. Hu, A. Song, Continuous shared control of a mobile robot with brain–computer interface and autonomous navigation for daily assistance, Comput. Struct. Biotechnol. J. 22 (2023) 3–16, https://doi.org/10.1016/j.csbj.2023.07.033.

[30] G. Niemeyer, C. Preusche, G. Hirzinger, Telerobotics, in: B. Siciliano, O. Khatib (Eds.), Springer Handb. Robot, editors, Springer, Berlin, Heidelberg, 2008, pp. 741–757, https://doi.org/10.1007/978-3-540-30301-5_32.

[31] E. Tsagkournis, D. Panagopoulos, G. Petousakis, G. Nikolaou, R. Stolkin, M. Chiou, A supervised machine learning approach to operator intent recognition for teleoperated mobile robot navigation, IFAC-Pap. 56 (2023) 8333–8338, https://doi.org/10.1016/j.ifacol.2023.10.1023.

[32] Artificial intelligence (AI) | definition, examples, types, applications, companies, & facts | britannica n.d. https://www.britannica.com/technology/artificial-intelligence.

[33] S. Fahle, C. Prinz, B. Kuhlenkötter, Systematic review on machine learning (ML) methods for manufacturing processes – Identifying artificial intelligence (AI) methods for field application, Procedia CIRP 93 (2020) 413–418, https://doi.org/10.1016/j.procir.2020.04.109.

[34] E. Alpaydin, Introduction to Machine Learning, Fourth edition, MIT Press, 2020.

[35] F. Gul, W. Rahiman, S.S. Nazli Alhady, A comprehensive study for robot navigation techniques, Cogent. Eng. 6 (2019) 1632046, https://doi.org/10.1080/23311916.2019.1632046.

[36] M. Soori, B. Arezoo, R. Dastres, Artificial intelligence, machine learning and deep learning in advanced robotics, a review, Cogn. Robot. 3 (2023) 54–70, https://doi.org/10.1016/j.cogr.2023.04.001.

[37] M. Woschank, E. Rauch, H. Zsifkovits, A review of further directions for artificial intelligence, machine learning, and deep learning in smart logistics, Sustainability 12 (2020) 3760, https://doi.org/10.3390/su12093760.

[38] S. Desale, A. Rasool, S. Andhale, P. Rane, Heuristic and meta-Heuristic algorithms and their relevance to the real world: a survey, Int. J. Comput. Eng. Res. Trends 351 (2015) 2349–7084.

[39] O.W. Khalid, N.A.M.,.M. Isa, HA. Sakim, Emperor penguin optimizer: A comprehensive review based on state-of-the-art meta-heuristic algorithms, Alex. Eng. J. 63 (2023) 487–526, https://doi.org/10.1016/j.aej.2022.08.013.

[40] P. DrA, Mobile robot navigation and obstacle avoidance techniques: a review, Int. Robot. Autom. J. 2 (2017) 1–12, https://doi.org/10.15406/iratj.2017.02.00023.

[41] S. Abgenah, AA. Jamal, Heuristic algorithms for robot path planning: a review, IJARCCE 10 (2021), https://doi.org/10.17148/IJARCCE.2021.10112.

[42] Q. Song, S. Li, J. Yang, Q. Bai, J. Hu, X. Zhang, A. Zhang, Intelligent optimization algorithm-based path planning for a mobile robot, Comput. Intell. Neurosci. 2021 (2021) e8025730, https://doi.org/10.1155/2021/8025730.

[43] J.R. Sánchez-Ibáñez, C.J. Pérez-del-Pulgar, A. García-Cerezo, Path planning for autonomous mobile robots: A review, Sensors 21 (2021) 7898, https://doi.org/10.3390/s21237898.

[44] Korkmaz M., Durdu A. Comparison of optimal path planning algorithms. 2018. https://doi.org/10.1109/TCSET.2018.8336197.

[45] ER. Martinez, Robotic Navigation Based On Servo Task And Sensory Memory. phdthesis, Université de Picardie Jules Verne, 2020.

[46] Francis A., Perez-D'Arpino C., Li C., Xia F., Alahi A., Alami R., Bera A., Biswas A., Biswas J., Chandra R., Chiang H.-T., Everett M., Ha S., Hart J., How J., Karnan H., Lee T.-W., Manso L., Mirsky R., Martin-Martin R. Principles and guidelines for evaluating social robot navigation algorithms. 2023.

[47] M.K. Habib, Real time mapping and dynamic navigation for mobile robots, Int. J. Adv. Robot. Syst. 4 (2007) 35, https://doi.org/10.5772/5681.

[48] F.-M. Juan-Antonio, Simultaneous Localization and Mapping for Mobile Robots: Introduction and Methods: Introduction and Methods, IGI Global, 2012.

[49] Mobile robot localization - challenges and various methods n.d. https://roboticsbiz.com/mobile-robot-localization-challenges-and-various-methods/.

[50] S. Huang, G. Dissanayake, Robot Localization: An Introduction, Wiley Encycl. Electr. Electron. Eng., John Wiley & Sons, Ltd, 2016, pp. 1–10, https://doi.org/10.1002/047134608X.W8318.

[51] M. Guoqiang, F. Baris, Localization Algorithms and Strategies for Wireless Sensor Networks: Monitoring and Surveillance Techniques for Target Tracking: Monitoring and Surveillance Techniques for Target Tracking, IGI Global, 2009.

[52] (PDF) Classification of localization algorithms for wireless sensor network: A survey n.d. https://www.researchgate.net/publication/235944445_Classification_of_Localization_Algorithms_for_Wireless_Sensor_Network_A_Survey.

[53] R. Dagher, R. Quilez, Chapter 9: localization in wireless sensor networks. Wirel. Sens. Robot Netw., World Scientific, 2014, pp. 203–247, https://doi.org/10.1142/9789814551342_0009.

[54] R. Ali, R. Liu, Y. He, A. Nayyar, B. Qureshi, Systematic review of dynamic multi-object identification and localization: techniques and technologies, IEEe Access. 9 (2021) 122924–122950, https://doi.org/10.1109/ACCESS.2021.3108775.

[55] S. Campbell, N. O'Mahony, A. Carvalho, L. Krpalkova, D. Riordan, J. Walsh, Where am I? Localization Techniques for Mobile Robots A Review, in: 2020 6th Int. Conf. Mechatron. Robot. Eng, ICMRE, 2020, pp. 43–47, https://doi.org/10.1109/ICMRE49073.2020.9065135.

[56] P.K. Panigrahi, SK. Bisoy, Localization strategies for autonomous mobile robots: A review, J. King Saud. Univ - Comput. Inf. Sci. 34 (2022) 6019–6039, https://doi.org/10.1016/j.jksuci.2021.02.015.

[57] S.P. Singh, S. Sharma, Critical analysis of distributed localization algorithms for wireless sensor networks, Int. J. Wirel. Microw. Technol. 6 (2016) 72–83, https://doi.org/10.5815/ijwmt.2016.04.07.

[58] Q. Luo, C. Liu, X. Yan, Y. Shao, K. Yang, C. Wang, Z. Zhou, A distributed localization method for wireless sensor networks based on anchor node optimal selection and particle filter, Sensors 22 (2022) 1003, https://doi.org/10.3390/s22031003.

[59] Bachrach J., Taylor C. 1 Localization in sensor networks, 2004.

[60] E. Saad, M. Elhosseini, A. Haikal, Recent achievements in sensor localization algorithms, Alex. Eng. J. 57 (2018), https://doi.org/10.1016/j.aej.2018.11.008.

[61] N.A. Alrajeh, M. Bashir, B. Shams, Localization techniques in wireless sensor networks, Int. J. Distrib. Sens. Netw. 9 (2013) 304628, https://doi.org/10.1155/2013/304628.

[62] S. Thrun, Learning metric-topological maps for indoor mobile robot navigation, Artif. Intell. 99 (1998) 21–71, https://doi.org/10.1016/S0004-3702(97)00078-7.

[63] Filliat D., Meyer J-A. Map-based navigation in mobile robots - I. A review of localisation strategies n.d.:56.

[64] Thrun S., Gutmann S., Fox D., Burgard W., Kuipers B. Integrating topological and metroc maps for mobile robot navigation: a statistical approach. 1998.

[65] A. Zare Zardiny, F. Hakimpour, Route matching in sketch and metric maps, J. Geogr. Syst. 23 (2021) 381–405, https://doi.org/10.1007/s10109-020-00343-1.

[66] P. Racinskis, J. Arents, M. Greitans, Constructing maps for autonomous robotics: an introductory conceptual overview, Electronics 12 (2023) 2925, https://doi.org/10.3390/electronics12132925.

[67] Lang D., Paulus D. Semantic Maps for robotics n.d.

[68] B. Ebrahimi Soorchaei, M. Razzaghpour, R. Valiente, A. Raftari, Y.P. Fallah, High-definition map representation techniques for automated vehicles, Electronics 11 (2022) 3374, https://doi.org/10.3390/electronics11203374.

[69] Thrun S.. Robotic mapping: A survey n.d.

[70] C.-J. Lin, C.-C. Lai, K.-H. Hsia, C-S. Chen, Apply model-free adaptive control approach for mobile robot path following, J. Robot. Netw. Artif. Life 7 (2020) 190–193, https://doi.org/10.2991/jrnal.k.200909.010.

[71] L. Liu, X. Wang, X. Yang, H. Liu, J. Li, P. Wang, Path planning techniques for mobile robots: review and prospect, Expert. Syst. Appl. 227 (2023) 120254, https://doi.org/10.1016/j.eswa.2023.120254.

[72] Robot path planning and cooperation. N.d.

[73] Editorial, Path planning algorithms for robotic systems. RoboticsBiz. https://roboticsbiz.com/path-planning-algorithms-for-robotic-systems/, 2022.

[74] Stentz A. The D*algorithm for real-time planning of optimal traverses n.d.:34.

[75] S. LaValle, Rapidly-Exploring Random Trees : a New Tool For Path Planning, Annu Res Rep, 1998.

[76] D. Fox, W. Burgard, S. Thrun, The dynamic window approach to collision avoidance, IEEe Robot. Autom. Mag. 4 (1997) 23–33, https://doi.org/10.1109/100.580977.

[77] Z. Tang, H. Ma, An overview of path planning algorithms, IOP Conf. Ser. Earth Env. Sci. 804 (2021) 022024, https://doi.org/10.1088/1755-1315/804/2/022024.

[78] H. Zhang, W. Lin, A. Chen, Path planning for the mobile robot: A review, Symmetry 10 (2018) 450, https://doi.org/10.3390/sym10100450.

[79] DrrKsrRA E.W.. A note on two problemsin connexionwith graphs n.d.

[80] Debets R., Dijkstra E.W. Edsger Dijkstra and the shortest-path algorithm David Gries 2018.

[81] A. Javaid, Understanding Dijkstra algorithm, SSRN Electron. J. (2013), https://doi.org/10.2139/ssrn.2340905.

[82] Venkat R.J. Path finding - Dijkstra's algorithm n.d.

[83] Sasi Kumar G., Shravan B., Gole H., Barve P., Ravikumar L.. Path Planning algorithms: a comparative study. 2011.

[84] P.E. Hart, N.J. Nilsson, B. Raphael, A formal basis for the heuristic determination of minimum cost paths, IEEE Trans. Syst. Sci. Cybern. 4 (1968) 100–107, https://doi.org/10.1109/TSSC.1968.300136.

[85] M.R. Wayahdi, S.H.N. Ginting, SD. Greedy, A-star, and Dijkstra's algorithms in finding shortest path, Int. J. Adv. Data Inf. Syst. 2 (2021) 45–52, https://doi.org/10.25008/ijadis.v2i1.1206.

[86] L. Chu, Y. Wang, S. Li, Z. Guo, W. Du, J. Li, Z. Jiang, Intelligent vehicle path planning based on optimized A* algorithm, Sensors 24 (2024) 3149, https://doi.org/10.3390/s24103149.

[87] Recent advances in robot path planning algorithms: a review of theory and experiment – Nova Science Publishers n.d. https://novapublishers.com/shop/recent-advances-in-robot-path-planning-algorithms-a-review-of-theory-and-experiment/.

[88] S. Kadry, G. Alferov, V. Fedorov, D-star algorithm modification, Int. J. Online Biomed. Eng. IJOE 16 (2020) 108, https://doi.org/10.3991/ijoe.v16i08.14243.

[89] L. Zhang, Z. Lin, J. Wang, B. He, Rapidly-exploring random Trees multi-robot map exploration under optimization framework, Robot. Aut. Syst. 131 (2020) 103565, https://doi.org/10.1016/j.robot.2020.103565.

[90] X. Jiang, Z. Wang, C. Dong, A path planning algorithm based on improved RRT sampling region, Comput. Mater. Contin. 80 (2024) 4303–4323, https://doi.org/10.32604/cmc.2024.054640.

[91] Chakraborty A., Kar A. Swarm intelligence: a review of algorithms, 2017, p. 475–94. https://doi.org/10.1007/978-3-319-50920-4_19.

[92] J. Hu, A.E. Turgut, T. Krajník, B. Lennox, F. Arvin, Occlusion-based coordination protocol design for autonomous robotic shepherding tasks, IEEe Trans. Cogn. Dev. Syst. 14 (2022) 126–135, https://doi.org/10.1109/TCDS.2020.3018549.

[93] O. Khare, S. Ahmed, Y. Singh, An overview of swarm intelligence-based algorithms, in: D. Singh, V. Garg, K. Deep (Eds.), Des. Appl. Nat. Inspired Optim. Contrib. Women Lead. Field, editors, Springer International Publishing, Cham, 2022, pp. 1–18, https://doi.org/10.1007/978-3-031-17929-7_1.

[94] J. Hu, P. Bhowmick, I. Jang, F. Arvin, A. Lanzon, A decentralized cluster formation containment framework for multirobot systems, IEEe Trans. Robot. 37 (2021) 1936–1955, https://doi.org/10.1109/TRO.2021.3071615.

[95] Vision-Based Mobile Robot Control and Path Planning Algorithms in Obstacle Environments Using Type-2 Fuzzy Logic | SpringerLink n.d. https://link.springer.com/book/10.1007/978-3-030-69247-6.

[96] L. Qiao, X. Luo, Q. Luo, An optimized probabilistic roadmap algorithm for path planning of mobile robots in complex environments with narrow channels, Sensors 22 (2022) 8983, https://doi.org/10.3390/s22228983.

[97] Y. Lu, X. Zhucun, G.-S. Xia, L. Zhang, A survey on vision-based UAV navigation, Geo.-Spat. Inf. Sci. (2018) 1–12, https://doi.org/10.1080/10095020.2017.1420509.

[98] O. Khatib, Real-time obstacle avoidance for manipulators and mobile robots, in: I. J. Cox, G.T. Wilfong (Eds.), Auton. Robot Veh, editors, Springer, New York, NY, 1990, pp. 396–404, https://doi.org/10.1007/978-1-4613-8997-2_29.

[99] R. Szczepanski, A. Bereit, T. Tarczewski, Efficient local path planning algorithm using artificial potential field supported by augmented reality, Energies 14 (2021) 6642, https://doi.org/10.3390/en14206642.

[100] A.M.H. Aljassani, S.N. Ghani, A.M.H. Al-Hajjar, Enhanced multi-agent systems formation and obstacle avoidance (EMAFOA) control algorithm, Results Eng. 18 (2023) 101151, https://doi.org/10.1016/j.rineng.2023.101151.

[101] Adrian Boeing: blog: dynamic window algorithm motion planning n.d. http://adrianboeing.blogspot.com/2012/05/dynamic-window-algorithm-motion.html.

[102] M. Kobayashi, N. Motoi, Local path planning: dynamic window approach with virtual manipulators considering dynamic obstacles, IEEe Access. 10 (2022) 17018–17029, https://doi.org/10.1109/ACCESS.2022.3150036.

[103] X. Xu, J. Zeng, Y. Zhao, X. Lü, Research on global path planning algorithm for mobile robots based on improved A*, Expert Syst. Appl. 243 (2024) 122922, https://doi.org/10.1016/j.eswa.2023.122922.

[104] Y. Ou, Y. Fan, X. Zhang, Y. Lin, W. Yang, Improved A* path planning method based on the grid map, Sensors 22 (2022) 6198, https://doi.org/10.3390/s22166198.

[105] A. Zhang, C. Li, W. Bi, Rectangle expansion A* pathfinding for grid maps, Chin. J. Aeronaut. 29 (2016) 1385–1396, https://doi.org/10.1016/j.cja.2016.04.023.

[106] T. Guo, Y. Sun, Y. Liu, L. Liu, J. Lu, An automated guided vehicle path planning algorithm based on improved A* and dynamic window approach fusion, Appl. Sci. 13 (2023) 10326, https://doi.org/10.3390/app131810326.

[107] K. Karur, N. Sharma, C. Dharmatti, J.E. Siegel, A survey of path planning algorithms for mobile robots, Vehicles 3 (2021) 448–468, https://doi.org/10.3390/vehicles3030027.

[108] Yu J., Nguyen T.T., Dao T.-K., Nguyen T-T. Global path planning algorithm for mobile robots: A review n.d.

[109] Davis D., Supriya P. Implementation of fuzzy-based robotic path planning 2016; 380:375–83. https://doi.org/10.1007/978-81-322-2523-2_36.

[110] Bounini F., Gingras D., Pollart H., Gruyer D. Modified artificial potential field method for online path planning applications. 2017. https://doi.org/10.1109/IVS.2017.7995717.

[111] C. Liu, Q. Mao, X. Chu, S. Xie, An improved A-star algorithm considering water current, traffic separation and berthing for vessel path planning, Appl. Sci. 9 (2019) 1057, https://doi.org/10.3390/app9061057.

[112] A. Ma'arif, O. Wahyunggoro, A. Imam, Artificial potential field algorithm implementation for quadrotor path planning, Int. J. Adv. Comput. Sci. Appl. 10 (2019), https://doi.org/10.14569/IJACSA.2019.0100876.

[113] Y. Hu, L. Yang, Y. Lou, Path planning with Q-learning, J. Phys. Conf. Ser. 1948 (2021) 012038, https://doi.org/10.1088/1742-6596/1948/1/012038.

[114] J. Li, Y. Chen, X. Zhao, J. Huang, An improved DQN path planning algorithm, J. Supercomput. 78 (2022) 616–639, https://doi.org/10.1007/s11227-021-03878-2.

[115] G. Xin, L. Shi, G. Long, W. Pan, Y. Li, J. Xu, Mobile robot path planning with reformative bat algorithm, PLoS One 17 (2022) e0276577, https://doi.org/10.1371/journal.pone.0276577.

[116] J. Huang, H. Wu, Mobile robot path planning based on improved coyote optimization algorithm, Math. Probl. Eng. 2022 (2022) 1–14, https://doi.org/10.1155/2022/4465542.

[117] S. Jameel Al-Kamil, R. Szabolcsi, Optimizing path planning in mobile robot systems using motion capture technology, Results Eng. 22 (2024) 102043, https://doi.org/10.1016/j.rineng.2024.102043.

[118] S. Liu, Q. Tian, C. Tang, Mobile robot path planning algorithm based on NSGA-II, Appl. Sci. 14 (2024) 4305, https://doi.org/10.3390/app14104305.

[119] Y. Xu, B. Sang, Y. Zhang, Application of improved sparrow search algorithm to path planning of mobile robots, Biomimetics 9 (2024) 351, https://doi.org/10.3390/biomimetics9060351.

[120] H. Liu, Y. Shen, S. Yu, Z. Gao, T. Wu, Deep reinforcement learning for mobile robot path planning, J. Theory Pr. Eng. Sci. 4 (2024) 37–44, https://doi.org/10.53469/jtpes.2024.04(04).07.

[121] X. Yi, A. Zhu, S.X. Yang, D. Shi, An improved neural dynamics based approach with territorial mechanism to online path planning of multi-robot systems, Int. J. Mach. Learn. Cybern. 12 (2021) 3561–3572, https://doi.org/10.1007/s13042-021-01405-6.

[122] R. Wang, Z. Lu, Y. Jin, C. Liang, Application of A* algorithm in intelligent vehicle path planning, Math. Models Eng. 8 (2022), https://doi.org/10.21595/mme.2022.22828.

[123] X. Yi, A. Zhu, S. Yang, MPPTM: A bio-inspired approach for online path planning and high-accuracy tracking of UAVs, Front. Neurorobotics 15 (2022), https://doi.org/10.3389/fnbot.2021.798428.

[124] R.A. Saeed, M. Omri, S. Abdel-Khalek, E.S. Ali, M.F. Alotaibi, Optimal path planning for drones based on swarm intelligence algorithm, Neural. Comput. Appl. 34 (2022) 10133–10155, https://doi.org/10.1007/s00521-022-06998-9.

[125] M. Munadi, B. Radityo, M. Ariyanto, Y. Taniai, Automated guided vehicle (AGV) lane-keeping assist based on computer vision, and fuzzy logic control under varying light intensity, Results Eng. 21 (2024) 101678, https://doi.org/10.1016/j.rineng.2023.101678.

[126] A. Singh, Y. Sridhar, V. Kalaichelvi, R. Karthikeyan, Performance evaluation of vision based path planning for dynamic real-time scenarios of mobile robot, Multimed. Tools. Appl. (2024), https://doi.org/10.1007/s11042-024-19267-9.

[127] T. Zhou, W. Wei, Mobile robot path planning based on an improved ACO algorithm and path optimization, Multimed. Tools. Appl. (2024), https://doi.org/10.1007/s11042-024-19370-x.

[128] N. Promkaew, S. Thammawiset, P. Srisan, P. Sanitchon, T. Tummawai, S. Sukpancharoen, Development of metaheuristic algorithms for efficient path planning of autonomous mobile robots in indoor environments, Results Eng. 22 (2024) 102280, https://doi.org/10.1016/j.rineng.2024.102280.

[129] H. Taheri, Z.C. Xia, SLAM; definition and evolution, Eng. Appl. Artif. Intell. 97 (2021) 104032, https://doi.org/10.1016/j.engappai.2020.104032.

[130] I. Lluvia, E. Lazkano, A. Ansuategi, Active mapping and robot exploration: a survey, Sensors 21 (2021) 2445, https://doi.org/10.3390/s21072445.

[131] R. Mahony, T. Hamel, J. Trumpf, An homogeneous space geometry for simultaneous localisation and mapping, Annu. Rev. Control 51 (2021) 254–267, https://doi.org/10.1016/j.arcontrol.2021.04.012.

[132] An introduction to simultaneous localization and mapping (SLAM) for robots - technical articles n.d. https://control.com/technical-articles/an-introduction-to-simultaneous-localization-and-mapping-slam-for-robots/.

[133] Aulinas J., Petillot Y., Salvi J., Llado X. The SLAM problem: a survey. vol. 184. 2008. https://doi.org/10.3233/978-1-58603-925-7-363.

[134] X. Wang, X. Wang, D.M. Wilkes, Machine Learning-based Natural Scene Recognition for Mobile Robot Localization in An Unknown Environment, Springer, 2019.

[135] H. Durrant-Whyte, T. Bailey, Simultaneous localization and mapping: part I, IEEe Robot. Autom. Mag. 13 (2006) 99–110, https://doi.org/10.1109/MRA.2006.1638022.

[136] H. Strasdat, J.M.M. Montiel, A.J. Davison, Visual SLAM: why filter? Image Vis. Comput. 30 (2012) 65–77, https://doi.org/10.1016/j.imavis.2012.02.009.

[137] A. Macario Barros, Y. Moline, F. Carrel, M. Michel, G. Corre, A comprehensive survey of visual SLAM algorithms, Robotics 11 (2022), https://doi.org/10.3390/robotics11010024.

[138] T. Bailey, H. Durrant-Whyte, Simultaneous localization and mapping (SLAM): part II, IEEe Robot. Autom. Mag. 13 (2006) 108–117, https://doi.org/10.1109/MRA.2006.1678144.

[139] S.J. Julier, J.K. Uhlmann, A counter example to the theory of simultaneous localization and map building, Proc. 2001 ICRA IEEE Int. Conf. Robot. Autom. Cat No01CH37164 4 (2001) 4238–4243, https://doi.org/10.1109/ROBOT.2001.933280, vol. 4.

[140] C.-C. Wang, C. Thorpe, Simultaneous localization and mapping with detection and tracking of moving objects, in: Proc. 2002 IEEE Int. Conf. Robot. Autom. Cat No02CH37292 3, 2002, pp. 2918–2924, https://doi.org/10.1109/ROBOT.2002.1013675, vol.3.

[141] B. Xi, R. Guo, F. Sun, Y. Huang, Simulation research for active simultaneous localization and mapping based on extended kalman filter, in: 2008 IEEE Int. Conf. Autom. Logist., 2008, pp. 2443–2448, https://doi.org/10.1109/ICAL.2008.4636578.

[142] D'Alfonso L., Griffo A., Muraca P., Pugliese P.. A SLAM algorithm for indoor mobile robot localization using an extended Kalman filter and a segment based environment mapping. 2013. https://doi.org/10.1109/ICAR.2013.6766461.

[143] Wadud R., Sun W. DyOb-SLAM : dynamic object tracking SLAM system. 2022. https://doi.org/10.48550/arXiv.2211.01941.

[144] L. Sun, J. Wei, S. Su, P. Wu, SOLO-SLAM: A parallel semantic SLAM algorithm for dynamic scenes, Sensors 22 (2022) 6977, https://doi.org/10.3390/s22186977.

[145] D. Bolkas, M. O'Banion, C. Belleman, Combination of tls and slam lidar for levee monitoring, ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci. (2022) 641–647, https://doi.org/10.5194/isprs-annals-V-3-2022-641-2022. V-3–2022.

[146] Liu R., Deng Z., Zhiqiang C., Bin Othman M.S., Lau P.L.B., Chen K., Bhowmik K., Yuen C., Tan U.-X.. Distributed ranging SLAM for multiple robots with ultra-WideBand and odometry measurements. 2022. https://doi.org/10.48550/arXiv.2207.03700.

[147] X. Teng, Z. Shen, L. Huang, H. Li, W. Li, Multi-sensor fusion based wheeled robot research on indoor positioning method, Results Eng. 22 (2024) 102268, https://doi.org/10.1016/j.rineng.2024.102268.

[148] J. Qiao, J. Guo, Y. Li, Simultaneous localization and mapping (SLAM)-based robot localization and navigation algorithm, Appl. Water. Sci. 14 (2024) 151, https://doi.org/10.1007/s13201-024-02183-6.

[149] W. Qian, J. Peng, H. Zhang, DFD-SLAM: visual SLAM with deep features in dynamic environment, Appl. Sci. 14 (2024) 4949, https://doi.org/10.3390/app14114949.

[150] Y. Huang, F. Xie, J. Zhao, Z. Gao, J. Chen, F. Zhao, X. Liu, ULG-SLAM: A novel unsupervised learning and geometric feature-based visual SLAM algorithm for robot localizability estimation, Remote Sens. 16 (2024) 1968, https://doi.org/10.3390/rs16111968.

[151] X. Xie, Y. Qin, Z. Zhang, Z. Yan, H. Jin, M. Xu, C. Zhang, GY-SLAM: A dense semantic SLAM system for plant factory transport robots, Sensors 24 (2024) 1374, https://doi.org/10.3390/s24051374.

[152] Seifzadeh S., Wu D., Wang Y. Cost-effective active localization technique for mobile robots. 2010. https://doi.org/10.1109/ROBIO.2009.5420704.

[153] Kalman 1960.pdf n.d.

[154] Welch G., Bishop G. An introduction to the Kalman filter 2006.

[155] An introduction to key algorithms used in SLAM - technical articles n.d. https://control.com/technical-articles/an-introduction-to-key-algorithms-used-in-slam/.

[156] Kim Y., Bang H. Introduction to Kalman filter and its applications, 2018. https://doi.org/10.5772/intechopen.80600.

[157] S. Ting Goh, S.A. (Reza) Zekavat, O. Abdelkhalik, An introduction to Kalman filtering implementation for localization and tracking applications. Handb. Position Locat., John Wiley & Sons, Ltd., 2018, pp. 143–195, https://doi.org/10.1002/9781119434610.ch5.

[158] Im G.. Notes on Kalman Filter (KF, EKF, ESKF, IEKF, IESKF). 2024.

[159] E.A. Wan, R. Van Der Merwe, The unscented Kalman filter for nonlinear estimation, in: Proc. IEEE 2000 Adapt. Syst. Signal Process. Commun. Control Symp. Cat No00EX373, Lake Louise, Alta., Canada, IEEE, 2000, pp. 153–158, https://doi.org/10.1109/ASSPCC.2000.882463.

[160] K. György, A. Kelemen, L. Dávid, Unscented Kalman filters and particle filter methods for nonlinear State estimation, Procedia Technol. 12 (2014) 65–74, https://doi.org/10.1016/j.protcy.2013.12.457.

[161] M. Roth, G. Hendeby, C. Fritsche, F. Gustafsson, The ensemble Kalman filter: a signal processing perspective, EURASIP. J. Adv. Signal. Process. (2017) 56, https://doi.org/10.1186/s13634-017-0492-x, 2017.

[162] (PDF) Sequential Monte Carlo methods n.d. https://www.researchgate.net/publication/242754136_Sequential_Monte_Carlo_Methods.

[163] J. Correa, A. Soto, Active visual perception for mobile robot localization, J. Intell. Robot. Syst. 58 (2010) 339–354, https://doi.org/10.1007/s10846-009-9348-4.

[164] Y. Wang, W. Chen, J. Wang, H. Wang, Active global localization based on localizability for mobile robots, Robotica 33 (2015) 1609–1627, https://doi.org/10.1017/S0263574714000940.

[165] Krishna S., Seo K., Bhatt D., Mai V., Murthy K., Paull L. Deep active localization. 2019.

[166] L. Gong, X. Yu, J. Wang, Curve-localizability-SVM active localization research for mobile robots in outdoor environments, Appl. Sci. 11 (2021) 4362, https://doi.org/10.3390/app11104362.

[167] F. Andrade, M. Llofriu, M. Marzoa Tanco, G. Trinidad Barnech, G. Tejera, Active localization strategy for hypotheses pruning in challenging environments, J. Intell. Robot. Syst. 106 (2022) 47, https://doi.org/10.1007/s10846-022-01748-4.

[168] D. Ning, S. Huang, L-PCM: Localization and point cloud registration-based method for pose calibration of mobile robots, Information 15 (2024) 269, https://doi.org/10.3390/info15050269.

[169] M. Sasaki, Y. Tsuda, K. Matsushita, Development of autonomous mobile robot with 3DLidar self-localization function using layout map, Electronics 13 (2024) 1082, https://doi.org/10.3390/electronics13061082.

[170] Y. Xia, Y. Huang, H. Qin, Y. Shi, Monte Carlo localization based on off-line feature matching and improved particle swarm optimization for mobile robots, Intell. Robot.. (2024), https://doi.org/10.1007/s11370-024-00524-7.

[171] Buildings | free full-text | an improved trilateral localization technique fusing extended kalman filter for mobile construction robot n.d. https://www.mdpi.com/2075-5309/14/4/1026.

[172] H. Zhu, Q. Luo, Indoor localization of mobile robots based on the fusion of an improved AMCL algorithm and a collision algorithm, IEEe Access. 12 (2024) 67199–67208, https://doi.org/10.1109/ACCESS.2024.3399192.

[173] M. Pellicelli, Chapter two - A long road to maximizing efficiency, in: M. Pellicelli (Ed.), Digit. Transform. Supply Chain Manag., editor, Elsevier, 2023, pp. 37–61, https://doi.org/10.1016/B978-0-323-85532-7.00009-8.

[174] M. Ogunsina, C. Pelumi Efunniyi, O. Soji Osundare, S. Olaoluwa Folorunsho, L. A. Akwawa, Advanced sensor fusion and localization techniques for autonomous systems: a review and new approaches, Int. J. Frontline Res. Eng. Technol. 2 (2024) 051–060, https://doi.org/10.56355/ijfret.2024.2.1.0022.

[175] L. Chen, C. Xia, Z. Zhao, H. Fu, Y. Chen, AI-driven sensing technology: review, Sensors 24 (2024) 2958, https://doi.org/10.3390/s24102958.

[176] R.K. C, J. Raja, S.S. Marotrao, N. Neelima, R. Sujatha, K. Chidambarathanu, AI algorithms for autonomous robotics navigation, in: 2024 Second Int. Conf. Comput. Charact. Tech. Eng. Sci. IC3TES, 2024, pp. 1–4, https://doi.org/10.1109/IC3TES62412.2024.10877509.

[177] Z. Zhang, L. Wang, C. Lee, Recent advances in artificial intelligence sensors, Adv. Sens. Res. 2 (2023) 2200072, https://doi.org/10.1002/adsr.202200072.

[178] G. Klančar, M. Seder, S. Blažič, Advanced sensors technologies applied in mobile robot, Sensors 23 (2023) 2958, https://doi.org/10.3390/s23062958.

[179] V. Ušinskis, M. Nowicki, A. Dzedzickis, V. Bučinskas, Sensor-fusion based navigation for autonomous mobile robot, Sensors 25 (2025) 1248, https://doi.org/10.3390/s25041248.

[180] S. Friedenthal, A. Moore, R. Steiner, Chapter 18 - integrating SysML into a systems development environment, in: S. Friedenthal, A. Moore, R. Steiner (Eds.), Pract. Guide SysML Third Ed, Morgan Kaufmann, Boston, 2015, pp. 507–541, https://doi.org/10.1016/B978-0-12-800202-5.00018-7.

[181] L. Dekker, Simulation environments, in: A. Sydow, S.G. Tzafestas, R. Vichnevetsky (Eds.), Syst. Anal. Simul. II, editors, Springer US, New York, NY, 1988, pp. 344–350, https://doi.org/10.1007/978-1-4613-8936-1_74.

[182] ROS/introduction - ROS wiki n.d. http://wiki.ros.org/ROS/Introduction.

[183] ROS: home n.d. https://www.ros.org/.

[184] St-Onge D., Herath D. The robot operating system (ROS1 &2): programming paradigms and deployment, 2022, p. 105–26. https://doi.org/10.1007/978-981-19-1983-1_5.

[185] K. Zhu, T. Zhang, Deep reinforcement learning based mobile robot navigation: a review, Tsinghua Sci. Technol. 26 (2021) 674–691, https://doi.org/10.26599/TST.2021.9010012.

[186] A. Loganathan, N.S. Ahmad, A systematic review on recent advances in autonomous mobile robot navigation, Eng. Sci. Technol. Int. J. 40 (2023) 101343, https://doi.org/10.1016/j.jestch.2023.101343.

[187] A.N. Atiyah, N. Adzhar, N.I. Jaini, An overview: on path planning optimization criteria and mobile robot navigation, J. Phys. Conf. Ser. 1988 (2021) 012036, https://doi.org/10.1088/1742-6596/1988/1/012036.

[188] N. Adzhar, Y. Yusof, M.A. Ahmad, A review on autonomous mobile robot path planning algorithms, Adv. Sci. Technol. Eng. Syst. J. 5 (2020) 236–240, https://doi.org/10.25046/aj050330.

[189] A. Sanyal, N.Z. Mohd, J.C. Mohanta, F.A. Md, Path planning approaches for mobile robot navigation in various environments: a review, in: N. Kumar, S. Tibor, R. Sindhwani, J. Lee, P. Srivastava (Eds.), Adv. Interdiscip. Eng., Springer, Singapore, 2021, pp. 555–572, https://doi.org/10.1007/978-981-15-9956-9_55.