International Journal of Science and Research Archive
eISSN: 2582-8185
Cross Ref DOI: 10.30574/ijsra
Journal homepage: https://ijsra.net/

(Review Article)

Navigation in autonomous mobile robots

Vanshika Gopal Karwa, Srimannarayana Bommaraju, Akshar Yogendra Malgundkar, Aryan Ahmed Shariffi, Jahida J Subhedar * and Hrushikesh B Kulkarni

School of Mechatronics Engineering, Symbiosis Skills and Professional University, Pune (MH) India.

International Journal of Science and Research Archive, 2025, 16(01), 386-393

Publication history: Received on 25 May 2025; revised on 01 July 2025; accepted on 04 July 2025

Article DOI: https://doi.org/10.30574/ijsra.2025.16.1.1897

Abstract

This paper explores the design and development of Autonomous Mobile Robots (AMRs) using the Robot Operating System (ROS) for intelligent navigation. AMRs equipped with ROS can perceive their environment, planning optimal paths, and making real-time decisions, making them suitable for a wide range of industrial and service-based applications. The integration of technologies such as Simultaneous Localization and Mapping (SLAM), sensor fusion, and dynamic obstacle avoidance enables these robots to operate reliably in both known and unknown environments. The report highlights how core ROS components like navigation stacks, cost maps, and transform libraries contribute to flexible and scalable robot behavior. It also examines the challenges involved in localization, path planning, and hardware software integration, offering insights into emerging solutions such as AI-enhanced navigation, edge computing, and ROS2 enhancements. By focusing on real-world applicability and innovation, this report shows the growing potential of ROS-powered AMRs in autonomous navigation.

Keywords: Autonomous Mobile Robots; ROS; Navigation; SLAM; Path Planning; Sensor Fusion; Localization

1. Introduction

Autonomous Mobile Robots (AMRs) are intelligent robotic systems designed to navigate and perform tasks in complex, dynamic environments without requiring human input. Unlike traditional robots that operate in controlled settings along fixed paths, AMRs leverage onboard sensors and intelligent algorithms to perceive their surroundings, make decisions, and adapt in real-time [1]. With features such as mapping, localization, path planning, and obstacle avoidance, AMRs are widely deployed in sectors like warehouse automation, agriculture, and healthcare [2].

Navigation lies at the core of AMR functionality, enabling these robots to move purposefully within their environment. For effective navigation, the system must accurately determine its position (localization), understand the environment (mapping), and calculate feasible paths to reach a goal [1]. Classical techniques like grid-based planning and heuristic searches laid the foundation, but recent advancements have introduced modern techniques such as machine learning and sensor fusion, significantly enhancing the adaptability and robustness of navigation in real-world scenarios [2].

A central enabler of AMR development is the Robot Operating System (ROS)—an open-source middleware framework that streamlines robotic software development. ROS provides essential tools and services including hardware abstraction, device drivers, inter-process communication, and package management. Its modular architecture encourages reuse and collaboration among developers, greatly accelerating innovation in the robotics field [1][3].

One of ROS's most valuable assets is its navigation stack, which integrates key navigation components—localization, mapping, path planning, and obstacle avoidance—into a cohesive system. It supports algorithms such as Simultaneous Localization and Mapping (SLAM) and the Dynamic Window Approach (DWA) for real-time decision-making [4]. The

* Corresponding author: Jahida J Subhedar

transition from ROS 1 to ROS 2 brought significant improvements, including real-time capability, enhanced security, and efficient multi-robot support through its DDS-based communication architecture [3].

In applied research and development, ROS has demonstrated effectiveness in building compact autonomous robots equipped with LiDAR sensors for real-time mapping and navigation tasks [5]. It also supports deep learning frameworks that enhance robot perception and decision-making capabilities, enabling adaptive behavior in complex environments [6]. Thus, ROS plays a pivotal role in driving forward the capabilities of AMRs, serving as a foundational software platform for developing robust, scalable, and intelligent autonomous systems.

Navigation algorithms are fundamental to the operation of Autonomous Mobile Robots (AMRs), enabling them to efficiently plan paths, avoid obstacles, and reach specified goals. Traditional methods such as A* and Dijkstra algorithms have been widely adopted due to their effectiveness in structured environments. However, as real-world conditions are often dynamic and uncertain, more adaptive approaches like the Dynamic Window Approach (DWA), Rapidly exploring Random Trees (RRT), and reinforcement learning have gained prominence [7][8]. These modern techniques allow AMRs to adjust their navigation strategies in real time, thus enhancing safety and responsiveness in complex settings.

The Robot Operating System (ROS) provides a powerful framework to implement these algorithms and supports a variety of navigation setups tailored to different robot platforms. Its comprehensive ecosystem includes packages for sensor integration, map generation, localization, and path planning. Prominent tools such as the ROS Navigation Stack, Move Base, and Costmap 2D enable real-time operations using both global and local planning strategies [9]. This modularity allows for rapid customization, integration of new algorithms or sensors, and streamlining of deployment processes for varied robotic systems.

One of the cornerstone technologies for autonomous navigation is Simultaneous Localization and Mapping (SLAM), which enables robots to map unfamiliar environments while tracking their position. Within ROS, algorithms like GMapping, Hector SLAM, and Cartographer have proven effective for accurate map building. To enhance localization accuracy, Adaptive Monte Carlo Localization (AMCL) is often paired with SLAM, leveraging particle filter techniques [10][11]. Moreover, sensor fusion strategies—integrating data from LiDAR, cameras, inertial measurement units (IMUs), and wheel encoders—are essential for robust obstacle detection and environment understanding, particularly in cluttered or changing environments [12].

The performance of AMRs also heavily depends on their hardware platforms. Standard platforms such as TurtleBot and Clearpath Husky, along with custom-designed robots, offer flexibility for indoor and outdoor experiments. These platforms typically incorporate LiDAR, RGB-D cameras, ultrasonic sensors, and IMUs to support navigation, SLAM, and perception research [13][14]. Technological advancements in sensor miniaturization and onboard processing power have further enabled the development of compact, energy-efficient robots that are well-suited for a wide range of field applications.

AMRs have demonstrated significant utility across various domains, including warehouse logistics, precision agriculture, and emergency response. In warehouses, they autonomously transport goods and manage inventory, thus improving efficiency [15]. In agriculture, AMRs carry out activities such as crop monitoring and precision spraying, adapting effectively to diverse field conditions. Search-and-rescue operations also benefit from autonomous robots equipped with advanced navigation algorithms, enabling them to access and map dangerous or inaccessible areas to locate survivors [16][17]. These examples illustrate the versatility and transformative potential of AMRs in enhancing productivity and safety across industries.

2. Technologies and Tools Used

Autonomous Mobile Robots (AMRs) depend on a fusion of advanced technologies to function effectively in diverse and dynamic environments. These include navigation algorithms for path planning, SLAM (Simultaneous Localization and Mapping) for mapping and localization, and the Robot Operating System (ROS) as an integration backbone. LiDAR, cameras, and IMUs serve as primary sensors for environmental perception, while simulation tools like Gazebo and Webots allow developers to safely test robot behavior before real-world deployment. Furthermore, the selection of appropriate hardware platforms—from lightweight research bots to rugged field units—shapes a robot's capabilities and potential use cases.

Navigation remains central to autonomy in robots, with classical algorithms such as A*, Dijkstra's, and RRT offering structured path-planning solutions [8]. However, as robots are increasingly deployed in unstructured and dynamic environments, these methods often need reinforcement. Techniques like reinforcement learning have emerged to enable adaptive path planning based on environmental feedback [7]. Hybrid systems that combine classical search strategies with AI methodologies are also gaining popularity for enhancing flexibility and reliability in real-time navigation [18].

SLAM technologies are vital for enabling robots to localize themselves while mapping unknown environments. Early SLAM methods used filters like EKF and particle filters, but graph-based optimization now offers improved scalability and accuracy [10][11]. LiDAR-based SLAM systems such as LOAM provide dense, real-time 3D mapping [12]. In addition, sensor fusion techniques that integrate data from LiDAR, cameras, IMUs, and wheel odometry are being applied to boost mapping reliability in sensor-noisy or changing environments [19]. These innovations have expanded SLAM's utility to outdoor, urban, and high-variability scenarios.

The Robot Operating System (ROS) has revolutionized how robotics software is built, allowing for modular, scalable development. With packages like move_base for navigation, gmapping and hector_slam for mapping, and amcl for localization, ROS offers a standardized ecosystem to simplify development [9]. The ROS 2 update has added critical improvements like real-time DDS communication, better security, and multi-robot capabilities—boosting its adoption across academic and commercial domains [3]. ROS's compatibility with simulation tools and diverse hardware drivers further enhances its importance in AMR development.

Lastly, sensors, simulations, and hardware platforms all influence a robot's autonomy and robustness. LiDARs offer high-accuracy range data, RGB-D cameras provide visual-depth context [20], and IMUs help track motion. Sensor fusion integrates these modalities to overcome limitations and improve situational awareness [19]. Simulators like Webots and CoppeliaSim are instrumental in testing navigation and coordination under realistic physical conditions [21]. Hardware choices, such as TurtleBot for education or Clearpath Husky for rugged tasks [13][14], impact deployment readiness. Advances in 3D printing and microcontrollers have further democratized access, allowing rapid customization for research and field use [13].

3. Limitations, Solutions and Improvements

Autonomous Mobile Robots (AMRs) encounter significant challenges when deployed in complex, dynamic environments. Unlike static lab conditions, real-world settings often include moving obstacles, unpredictable human behavior, and frequently changing layouts. Traditional path planning algorithms, which often assume semi-static surroundings, struggle in these contexts [8]. Replanning in real-time becomes essential when navigating such spaces, demanding both computational efficiency and adaptability [4]. In congested environments, robots may become stuck or deviate inefficiently from their paths, compromising overall task performance and increasing the risk of collisions if other agents' movements cannot be reliably predicted.

Localization accuracy is another critical factor that can hinder performance. In environments where GPS is unavailable, such as indoors, position estimates degrade over time due to cumulative errors from sensor noise, wheel slippage, and poor environmental perception [10][11]. Probabilistic methods like Adaptive Monte Carlo Localization (AMCL) help mitigate these errors by modeling a distribution over possible poses, but they remain sensitive to changes in environment or map inconsistencies [11]. Reliance on static, a priori maps can further exacerbate localization drift in environments that evolve over time.

Sensor reliability is a persistent limitation for AMRs. LiDAR, RGB-D cameras, IMUs, and ultrasonic sensors are all prone to environmental disturbances and hardware limitations [19][20]. For example, LiDAR performance can be significantly reduced by fog, dust, or rain [19], while cameras struggle in poor lighting or with reflective surfaces. IMUs suffer from drift and noise over time [20]. These limitations degrade SLAM performance and decision-making, making the system less reliable in safety-critical or mission-critical scenarios. Additionally, many robots operate on constrained embedded hardware, limiting the complexity of algorithms that can be executed in real-time [6][7].

To overcome these limitations, sensor fusion has emerged as a robust solution, integrating data from multiple sources—such as LiDAR, IMU, odometry, and vision—to create a more reliable understanding of the environment [19]. Techniques like the Extended Kalman Filter (EKF), particle filters, and deep learning-based fusion architectures significantly improve mapping and localization accuracy, particularly in visually challenging environments [19][22]. Furthermore, machine learning techniques, especially reinforcement learning, have become instrumental in allowing robots to learn and adapt in real-time [7]. These methods enable end-to-end learning pipelines that bypass rigid rules, resulting in more flexible, context-aware navigation [6][23].

ROS 2 offers another critical advancement by addressing many of the operational limitations found in ROS 1. Its support for real-time DDS communication middleware ensures low-latency and predictable message exchange, critical for dynamic tasks [3]. Features like built-in encryption, better multi-robot communication, and modular node management make ROS 2 suitable for scalable industrial deployments. Moreover, in human-shared environments, crowd-aware navigation strategies are increasingly essential [15][16]. By combining trajectory prediction, reinforcement learning, and social behavior modeling, robots can move fluidly among people, maintaining safety, comfort, and social norms in public or collaborative spaces like warehouses, airports, and hospitals [15][16].

4. Future Scope

As autonomous mobile robots (AMRs) advance, future developments are expected to significantly improve their intelligence, collaborative behavior, and integration into larger technological systems. Emerging innovations such as artificial intelligence (AI), semantic mapping, and frameworks like Industry 4.0 are transforming the AMR research landscape. Traditional navigation systems rely primarily on geometric data, but AI allows AMRs to achieve semantic navigation, where the robot understands the meaning and context of surrounding objects and environments [24]. Through semantic mapping, objects like desks or people can be labeled and interpreted, enabling AMRs to follow high-level commands like “fetch medicine from the kitchen” with contextual awareness [7][24].

Multi-robot systems offer promising scalability, redundancy, and operational efficiency in large-scale applications, including warehouse automation, agriculture, and disaster response [15]. These systems can collaboratively divide tasks, communicate effectively, and handle missions that are infeasible for a single robot. Techniques such as distributed SLAM, swarm behavior algorithms, and consensus-based control are under exploration to enhance coordination [16]. The ROS 2 platform, which supports real-time distributed systems, simplifies such architectures and provides improved performance for synchronized multi-robot operations [3][17].

The growing emphasis on Industry 4.0—a vision of interconnected, automated smart factories—has reinforced the relevance of ROS-powered AMRs. ROS 2 facilitates seamless communication between robots and enterprise-level systems by supporting real-time data flow, cloud integration, and interoperability [3][25]. AMRs in smart factories handle tasks like inspection, material handling, and inventory tracking, reducing human intervention while improving productivity [5]. The flexibility of ROS, along with its compatibility with industrial protocols and hardware platforms, makes it a strategic component of industrial automation ecosystems.

Despite rapid progress, several critical gaps remain that limit the widespread deployment of AMRs. One persistent challenge is ensuring robust operation in highly dynamic and human-populated environments where unpredictable movements disrupt navigation [4]. Furthermore, maintaining long-term autonomy is difficult due to sensor drift, outdated maps, and environmental changes that degrade system accuracy over time [10]. Additional concerns include the lack of explainability in AI-based decision-making and the difficulty in generalizing learned behaviors to unseen environments [26]. Standardization across diverse hardware platforms also remains underdeveloped, making benchmarking and deployment inconsistent [13].

Future research should focus on six key areas to address these limitations and unlock the full potential of AMRs. First, semantic mapping integrated with deep learning can enhance intelligent decision-making in human-centered environments [24]. Second, standardized, platform-independent ROS packages will improve software portability across diverse robotic systems [13]. Third, decentralized multi-robot planning combining swarm intelligence and structured control will boost coordination [15][16]. Fourth, adaptive sensor fusion models that learn contextually can outperform static schemes in complex scenarios [10]. Fifth, long-term autonomy can be improved with persistent memory architectures and self-updating SLAM techniques [12]. Finally, incorporating explainable AI into robot decision pipelines will build transparency and trust in sensitive or safety-critical deployments [26].

5. Autonomous Navigation and Integration

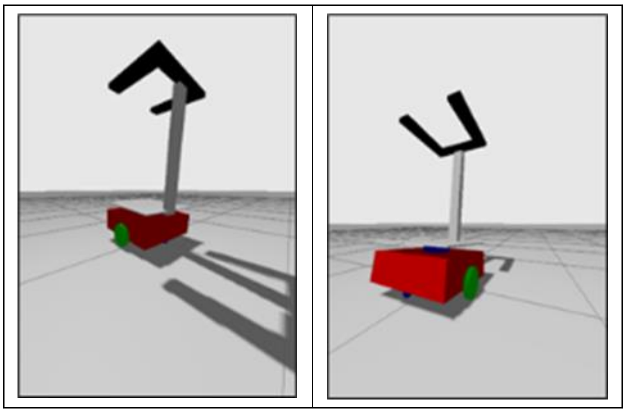

Figure 1 3-D Fusion Drawing of AMR Model

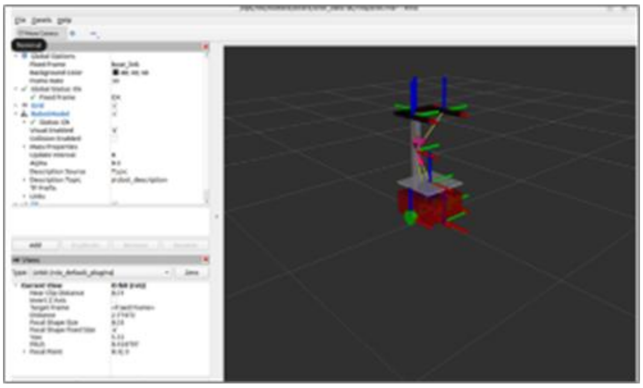

Figure 2 ROS Simulation of AMR in Gazebo

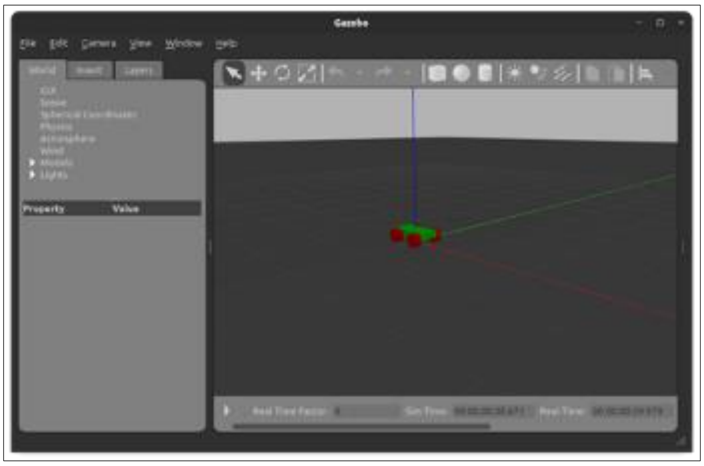

Figure 3 Simulation of AMR in Gazebo

Integrating autonomous mobile robots (AMRs) with the Robot Operating System (ROS) is a critical step following hardware design, as it enables software control of the robot's motors, sensors, and navigation logic. ROS2, the latest generation of the framework, is typically installed on a Linux-based platform such as Ubuntu 20.04 or 22.04. Developers begin by setting up a ROS workspace to organize robot-specific packages and nodes. Custom ROS packages are created to handle control nodes for actuators (e.g., differential drive motors), sensor drivers (e.g., LiDAR, IMUs), and robot models described using URDF or XACRO. ROS provides access to a variety of open-source drivers (like rplidar_ros) to simplify hardware integration.

Sensor and actuator integration with ROS involves establishing real-time communication through ROS topics, services, and actions. Motor controllers are configured using ros2_control and packages such as diff_drive_controller, while sensor data is continuously streamed for perception, mapping, and localization tasks. Each sensor (e.g., LiDAR, RGB cameras, encoders) feeds structured data into the ROS network, enabling the robot to interact with its environment. Before real-world deployment, robot behavior is validated using simulators such as Gazebo, which allows testing in virtual 3D environments that replicate physical operating conditions. The robot's geometry, joint constraints, and sensors are defined using URDF or SDF files for simulation compatibility.

Autonomous navigation is implemented using the ROS2 Navigation Stack (Nav2), which provides a modular architecture for mapping, localization, and path planning. The mapping process begins with Simultaneous Localization and Mapping (SLAM), using packages like slam_toolbox or cartographer. These tools process LiDAR data to construct a dynamic map and track the robot's position as it moves. slam_toolbox excels in real-time environments, while cartographer supports advanced 2D and 3D mapping. Localization is achieved through Adaptive Monte Carlo Localization (AMCL), a probabilistic method that continually estimates the robot's pose by comparing incoming sensor data to the known map. AMCL is essential for correcting positional drift and maintaining situational awareness.

With mapping and localization established, path planning algorithms guide the robot to its goals while avoiding obstacles. Algorithms like A* (A-star) and DWA (Dynamic Window Approach) are used within Nav2 for global and local path optimization. A* determines the shortest, safest route using map cost functions, while DWA dynamically adjusts the path in response to moving objects and sudden changes in the environment. The global planner defines the high-level route, while the local planner continuously updates it based on real-time sensor data, ensuring robust navigation in dynamic and unpredictable settings.

Testing and optimization complete the integration cycle. Simulation tools such as Gazebo allow the robot's behavior to be validated through obstacle courses and navigation tasks. Visualization tools like Rviz and monitoring utilities like ros2 topic echo and rqt help track the robot's state, sensor inputs, and node performance. Fine-tuning involves parameter adjustments for sensor calibration, motor control, and localization algorithms. Edge-case testing—such as sensor interference or map inconsistencies—helps refine system reliability. With successful debugging and validation, the AMR can be confidently deployed in real environments, equipped with a complete and robust ROS-based control and navigation system.

6. Conclusion

Autonomous Mobile Robots (AMRs) have emerged as a transformative force across sectors such as logistics, manufacturing, agriculture, and service robotics. Enabled by advanced navigation algorithms, SLAM techniques, sensor fusion, and middleware platforms like the Robot Operating System (ROS), these robots have evolved from basic mobile agents to intelligent systems capable of operating in dynamic, unstructured environments. This seminar report has explored the foundational concepts, implementation methodologies, hardware platforms, and software tools that underpin modern AMR development. It also highlighted real-world applications that demonstrate the growing utility and adaptability of AMRs across various industries. However, despite notable progress, AMRs continue to face challenges in areas like localization accuracy, real-time navigation in cluttered environments, and system reliability in human-populated spaces. Chapters 4 and 5 specifically address these issues—such as sensor noise and computational limitations—and propose emerging solutions including machine learning-based navigation, enhanced ROS2 frameworks, and human-aware path planning.

Looking ahead, the field of AMRs offers vast opportunities for research and innovation. Chapter 6 outlines future directions that include AI-powered semantic navigation, coordinated multi-robot collaboration, and seamless integration of ROS in Industry 4.0 ecosystems. These advances are not only expected to enhance technical performance but also deepen the contextual awareness and societal relevance of AMRs in industrial, urban, and natural settings. As open-source platforms and interdisciplinary research continue to accelerate development, AMRs are poised to become more autonomous, adaptive, and socially integrated. This seminar thus serves as both a snapshot of current advancements and a roadmap for future exploration, emphasizing the critical role AMRs will play in augmenting human capabilities and solving real-world challenges in increasingly complex environments

Compliance with ethical standards

Disclosure of conflict of interest

No conflict of interest to be disclosed.

References

[1] Tang, Y., Zakaria, M. A., and Younas, M. (2025). Path planning trends for autonomous mobile robot navigation: A review. Sensors, 25(4), 1206.

[2] Loganathan, A., and Ahmad, N. S. (2023). A systematic review on recent advances in autonomous mobile robot navigation. Engineering Science and Technology, an International Journal, 40, 101343.

[3] Macenski, S., Foote, T., Gerkey, B., Lalancette, C., and Woodall, W. (2022). Robot Operating System 2: Design, architecture, and uses in the wild. arXiv preprint arXiv:2211.07752.

[4] Kobayashi, M., Zushi, H., Nakamura, T., and Motoi, N. (2023). Local path planning: Dynamic window approach with Q-learning considering congestion environments for mobile robot. IEEE Access, 11, 96733–96742.

[5] Kumar, A., and Sharma, R. (2023). ROMR: A ROS-based open-source mobile robot. Robotics and Computer-Integrated Manufacturing, 78, 102110.

[6] Nguyen, A., and Tran, Q. (2020). Autonomous navigation with mobile robots using deep learning and the Robot Operating System. arXiv preprint arXiv:2012.02417.

[7] Chen, X., Wang, L., and Hu, Y. (2022). Reinforcement learning-based navigation for autonomous mobile robots: A review. Robotics and Autonomous Systems, 148, 103918.

[8] Silva, J., and Pereira, J. (2021). A comprehensive review of path planning algorithms in robotics. Journal of Intelligent and Robotic Systems, 101(3), 47.

[9] Quigley, M., Conley, K., Gerkey, B. P., Faust, J., Foote, T., Leibs, J., ... and Ng, A. Y. (2009). ROS: An open-source Robot Operating System. In ICRA workshop on open source software (Vol. 3, No. 3.2, p. 5).

[10] Durrant-Whyte, H., and Bailey, T. (2006). Simultaneous localization and mapping: Part I. IEEE Robotics and Automation Magazine, 13(2), 99–110.

[11] Fox, D., Burgard, W., and Thrun, S. (1999). Markov localization for mobile robots in dynamic environments. Journal of Artificial Intelligence Research, 11, 391–427.

[12] Zhang, J., and Singh, S. (2014). LOAM: Lidar odometry and mapping in real-time. In Robotics: Science and Systems Conference (RSS).

[13] Gopalan, A., and Prasanth, S. (2020). Review of open-source hardware platforms for autonomous mobile robots. International Journal of Robotics Research, 39(6), 740–759.

[14] Hentschel, J., and Thiel, T. (2021). ROS-based navigation on the Clearpath Husky: A case study. Robotics and Computer-Integrated Manufacturing, 67, 101994.

[15] Wurman, P. R., D'Andrea, R., and Mountz, M. (2008). Coordinating hundreds of cooperative, autonomous vehicles in warehouses. AI Magazine, 29(1), 9–20.

[16] Murphy, R. R. (2014). Disaster robotics. MIT Press.

[17] Valente, J., Sousa, J., and Gomes, A. (2020). Agricultural robots for precision crop monitoring: A survey. Computers and Electronics in Agriculture, 175, 105572.

[18] Huang, Y., Zhang, C., and Yang, L. (2023). Hybrid navigation algorithms for autonomous mobile robots: Combining classical and AI approaches. Robotics and Autonomous Systems, 155, 104202.

[19] Lee, S., Kim, D., and Park, J. (2021). Multi-sensor fusion for robust SLAM in dynamic environments. IEEE Transactions on Robotics, 37(3), 735–749.

[20] Borenstein, J., Everett, H. R., Feng, L., and Wehe, D. (1997). Mobile robot positioning: Sensors and techniques. Journal of Robotic Systems, 14(4), 231–249.

[21] Koenig, N., and Howard, A. (2004). Design and use paradigms for Gazebo, an open-source multi-robot simulator. In IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (pp. 2149–2154).

[22] Wan, Y., and Wang, Z. (2022). Deep sensor fusion for autonomous mobile robot localization and mapping. IEEE Transactions on Industrial Informatics, 18(7), 4741–4752.

[23] Luo, C., and Liu, H. (2021). Deep learning based visual navigation for autonomous mobile robots: A review. IEEE Access, 9, 112234–112250.

[24] Khan, M. S., Javed, M. Y., and Ullah, M. (2021). Semantic navigation in autonomous mobile robots: A survey. Journal of Field Robotics, 38(4), 510–531.

[25] Mejias, L., and Watterson, M. (2020). Leveraging ROS in Industry 4.0: Real-time integration with cyber-physical systems. Robotics and Computer-Integrated Manufacturing, 66, 101999.

[26] Cacace, J., Finzi, A., and Lippiello, V. (2022). Challenges and perspectives in explainable AI for autonomous mobile robots. AI Perspectives, 4(1), 1–15.