Indian Journal of Computer Science and Technology
Volume 3, Issue2 (May-August 2024), PP: 202-206.
www.indjcst.com
ISSN No: 2583-5300

Exploring Mobile Robot Navigation Protocols: A Detailed Summary

Abhinav Bhardwaj[1], Meenakshi Arora[2]
[1]*P.G. Student, Department of CSE, Sat Kabir Institute of Technology and Management, Haryana, India.*
[2]*Assistant Professor, Department of CSE, Sat Kabir Institute of Technology and Management, Haryana, India.*

To Cite this Article: *Abhinav Bhardwaj[1], Meenakshi Arora[2], "Exploring Mobile Robot Navigation Protocols: A Detailed Summary ", Indian Journal of Computer Science and Technology, Volume 03, Issue 02 (May-August 2024), PP: 202-206.*

Abstract: *Mobile robot navigation is a critical aspect of robotics, involving the development and implementation of protocols that enable robots to move autonomously and efficiently within various environments. Mobile robots must learn how to travel independently in an unfamiliar indoor environment while avoiding both static and dynamic hazards. The traditional navigational system lacks the capacity for self-learning. Nevertheless, the conditions found outside are significantly different from those observed in the lab. Because natural habitats are by their very nature unorganized, safe travel is practically impossible due to potholes, concealed rocks, irregular pavement, and ramps. To overcome these challenges and guarantee the successful deployment of a mobile robot in an outdoor environment, precise location and attitude data are needed. This summary provides an overview of the key navigation protocols and techniques used in mobile robotics, highlighting advancements, challenges, and future directions.*

Keywords: *Mobile robot navigation, Machine Learning, Obstacle Avoidance, Path Planning*

I.INTRODUCTION

Robots are used in many aspects of our daily life, including cleaning, medical support, rescue, military support, working in hazardous environments, and autonomous driving. The majority of the aforementioned applications call for mobile robots to maneuver through an unfamiliar environment without running into both stationary and moving impediments. The process via which a mobile robot moves to carry out a certain mission in its surroundings is called navigation. When a robot navigates its surroundings on its own, without assistance from an outside controller (such as a human), this is known as autonomous navigation. One of the main areas of study for mobile robotics is autonomous navigation [1]. Significant progress has been made in autonomous mobile robot navigation, thanks to advances in artificial intelligence (AI) and computer vision [2][3].

The conventional navigation method (i.e., map-building-based navigation) consists of localization, map building, and path planning (e.g., simultaneous localization and mapping (SLAM)) [4]. Using a map has many advantages since the entire planning and control system becomes computationally tractable by projecting high-dimensional observation, such as a camera image, into a three-dimensional pose on the map. In addition, it is possible to conveniently guarantee the optimality of the global path, however, it also has some drawbacks. First of all, making a precise environmental map takes a lot of effort and time, and it frequently calls for specialized knowledge. Second, over time, upkeep and updating the map may prove to be far more expensive, especially in the face of sudden developments. Third, the robot's theoretical model is the only factor that determines how effective the control is. This model is usually linearized or oversimplified, which eventually reduces the resilience of the navigation system. As a substitute to map-building-based navigation, mapless navigation is more frequently thought of as a way to get beyond the need for a map from the navigation system because it typically simulates a direct mapping between sensory inputs and robot actions. Regretfully, it is very challenging to plan the global journey for the best path without a map. Consequently, mapless navigation is used more often for tasks like collision avoidance that have no stated destinations or that have a destination that is known and within the robot's local coordinate frame. Because behavior-based navigation [5] lacks a high-level processing process based on past knowledge about the environment, mapless navigation is comparable to behavior-based navigation in this respect.

II.RESEARCH BACKGROUND

A suggested approach in [6] addresses the SLAM problem, a crucial issue for mobile robot autonomous navigation in an uncharted area. A minimal system implemented in basic mobile robots for indoor office-like settings is [7], a lightweight and real-time efficient SLAM algorithm. It makes the assumption that all of the lines in the environment structure are parallel to one another in an effort to simplify things. In order to enable autonomous interior navigation of a wheeled mobile robot in an environment characteristic of a greenhouse—one that is prohibited GPS reception—a novel approach combining the Hector SLAM and the artificial potential field (APF) controller was presented in [8].

The robot employs an APF controller for autonomous navigation, an open-source Hector SLAM for posture estimation, and single light detection and ranging (LiDAR) for localization. In [9], a conceptually rich graph model for indoor robotic navigation was developed. The navigational tasks use the semantic information directly to produce motor commands. As a result,

the robot can avoid performing explicit calculations to determine its exact location or the environment's shape. But no actual robot is used in the implementation of the method. In [10], a method for training the convolutional neural network (CNN) model only for autonomous mobile robot navigation was presented in an end-to-end manner utilizing an RGB-D camera. In [11], a navigation technique was presented to learn the end-to-end control policy by directly generating the velocity and angle rates from the present view and target using the CNN model. However, the previous methods focused on labeled observation-based behavior learning rather than environment-based learning, and they too required a lot of labeling and performed poorly when it came to generalization.

III.SIMULTANEOUS LOCALIZATION AND MAPPING (SLAM)

SLAM is a fundamental technique that enables a robot to create a map of an unknown environment while simultaneously keeping track of its location within that map. Various algorithms, such as Extended Kalman Filter (EKF) SLAM, Graph-Based SLAM, and Particle Filter SLAM, are employed to achieve this.

ORB-SLAM:

ORB-SLAM uses a monocular camera and features ORB (Oriented FAST and Rotated BRIEF) for feature extraction and tracking. It includes loop closure and relocalization capabilities, which are crucial for correcting drift and maintaining accurate positioning over time.

Hector SLAM:

Hector SLAM relies on lidar sensors for mapping and does not require odometry, which makes it suitable for environments where odometry data is unreliable. It is particularly effective in structured, indoor environments like greenhouses.

Graph SLAM:

This method employs a probabilistic approach, representing the SLAM problem as a graph of constraints. It is well-suited for large-scale mapping tasks, such as urban environments**.**

Continuous Control with RL and SLAM:

This approach integrates reinforcement learning with SLAM to enable continuous control and real-time navigation. It is designed to adapt to dynamic changes in the environment, making it suitable for complex and evolving scenarios.

SLAM Method	Authors	Key Features	Applications	Strengths	Limitations
ORB-SLAM	Mur-Artal, R.; Montiel, J.M.M.; Tardós, J.D.[12]	Monocular, feature-based, loop closure, relocalization	Indoor and outdoor environments, small to medium scale	High accuracy, versatile, real-time performance	Requires good feature-rich environment, struggles in texture-less areas
Hector SLAM	Harik, E.H.C.; Korsaeth, A.[8]	Lidar-based, map fusion, no odometry required	Indoor environments, greenhouses	High precision in structured environments, does not require odometry	Limited to 2D mapping, depends heavily on high-quality lidar data
Graph SLAM	Thrun, S.; Montemerlo, M.[13]	Probabilistic, graph-based optimization, large-scale mapping	Urban environments, large-scale outdoor mapping	Handles large-scale environments, robust against sensor noise	Computationally intensive, requires good initial guess for convergence
Continuous Control with RL and SLAM	Mustafa, K.A.A.; Botteghi, N.; Sirmacek, B.; Poel, M.; Stramigioli, S.[1]	Reinforcement learning integration, continuous control, real-time navigation	Dynamic and complex environments	Adaptability to dynamic changes, combines RL for better decision-making	Complex to implement, requires extensive training and data

Table 1: Tabular Comparison of SLAM-Based Methods

IV.MACHINE LEARNING-BASED MOBILE ROBOT NAVIGATION PROTOCOLS

Reinforcement Learning (RL)

RL algorithms enable robots to learn optimal navigation strategies through trial and error by interacting with their environment. Techniques such as Q-learning, Deep Q-Networks (DQN), and Policy Gradient methods are commonly used to solve navigation tasks in complex and dynamic environments.

Supervised Learning

Supervised learning algorithms, particularly convolutional neural networks (CNNs), are employed to process sensory data (e.g., images, LiDAR) and make real-time navigation decisions. These models are trained on labeled datasets to recognize obstacles, identify paths, and make directional decisions.

Unsupervised Learning

Techniques like clustering and anomaly detection are used for mapping and environmental understanding. These algorithms help in identifying patterns and structures within the data, enabling robots to navigate unknown or partially known environments without extensive pre-mapping.

Sensor Fusion with Machine Learning

Combining data from multiple sensors (e.g., cameras, LiDAR, IMUs) using machine learning techniques improves the accuracy and reliability of the robot's perception. Sensor fusion models like Kalman Filters and Bayesian Networks are enhanced with machine learning for better environmental awareness and decision-making.

Deep Learning for Navigation

Deep learning models, particularly deep reinforcement learning (DRL), are used for end-to-end learning of navigation tasks. These models can handle raw sensory inputs and learn complex policies for obstacle avoidance, path planning, and goal-directed behavior.

Method	Authors	Key Features	Applications	Strengths	Limitations
Deep Reinforcement Learning	Mnih, V., Kavukcuoglu, K., Silver, D., et al. (2015) [14]	Human-level control, deep Q-networks, model-free learning	Complex navigation tasks	High adaptability, learns directly from raw sensory input	Requires extensive training data, computationally intensive
Continuous Control with Deep RL	Lillicrap, T.P., et al. (2016)[15]	Continuous action spaces, actor-critic methods, deterministic policy gradient	Real-time control tasks	Handles high-dimensional action spaces, suitable for real-time applications	Requires significant computational resources for training
Convolutional Neural Networks (CNNs)	Krizhevsky, A., Sutskever, I., & Hinton, G.E. (2012)[16]	Deep CNNs, ImageNet classification, feature extraction	Object recognition for navigation	High accuracy in image classification, effective for feature extraction	Requires large labeled datasets for training
Representation Learning	Bengio, Y., et al. (2013)[17]	Unsupervised learning, feature learning, deep architectures	Environmental understanding	Learns useful representations from raw data, improves generalization	Requires careful design and tuning of learning algorithms
Multi-Sensor Fusion	Chen, C., et al. (2017)[18]	Sensor fusion, Kalman filters, Bayesian networks	Autonomous driving, indoor navigation	Combines multiple sensor inputs for accurate perception, robust to sensor noise	Complexity in sensor integration, computationally intensive
Gradient Descent Optimization	Ruder, S. (2016)[19][19]	Overview of optimization algorithms, convergence analysis	General ML applications	Improves convergence of learning algorithms, enhances training efficiency	Requires careful selection of optimization algorithm based on problem characteristics
Deep Learning for Game Playing	Silver, D., et al. (2016)[20]	Deep neural networks, tree search, reinforcement learning	Strategic planning, decision making	Achieves superhuman performance in complex decision-making tasks	Computationally intensive, requires significant resources for training
EfficientNet	Li, W., (2020)[21]	Model scaling, compound scaling, efficient neural networks	Resource-constrained environments	Achieves state-of-the-art accuracy with fewer parameters,	Requires careful tuning of scaling parameters efficient for deployment in mobile robots

End-to-End Training of Visuomotor Policies	Levine, S., et al. (2016)[22]	End-to-end learning, deep visuomotor policies, reinforcement learning	Robot manipulation, autonomous navigation	Directly maps sensory input to motor actions, effective for complex control tasks	Requires extensive training data, challenging to train end-to-end models
Probabilistic Robotics	Thrun, S., Burgard, W., & Fox, D. (2005)[13]	Probabilistic models, Bayesian inference, SLAM	SLAM, localization, mapping	Robust to sensor noise, handles uncertainty in measurements	Computationally intensive, requires good initial guesses for convergence

Table 2: Pivotal Role of Machine Learning in Advancing Mobile Robot Navigation

V.PATH PLANNING ALGORITHMS FOR ROBOT NAVIGATION

Path planning is a crucial aspect of autonomous robot navigation, ensuring that robots can move from one point to another safely and efficiently. Here is an overview of several commonly used path planning algorithms, each with its unique approach and applications:

Algorithm	**Description**	**Applications**	**Strengths**	**Limitations**	**Reference**
Dijkstra's	A graph search algorithm that finds the shortest path between nodes in a graph, ensuring non-negative edge weights.	Widely used in static environments where the map is fully known.	Guarantees the shortest path, simple and easy to implement.	Computationally expensive for large graphs	[23]
A (A-star) Algorithm*	An extension of Dijkstra's algorithm that incorporates heuristics to prioritize paths, making it faster and more efficient.	Used in games, robotics, and any pathfinding problems requiring efficiency.	Faster than Dijkstra's due to heuristic function, finds optimal path.	Performance depends on the quality of the heuristic.	[24]
Probabilistic Roadmaps (PRM)	A sampling-based algorithm that constructs a graph (roadmap) of random points in the space and connects them to find a path.	Effective for high-dimensional configuration spaces, like robotic arms.	Efficient for high-dimensional spaces, good for complex environments.	May not find the optimal path, depends on the sampling quality.	[25]
Rapidly-exploring Random Trees (RRT)	Another sampling-based algorithm that rapidly explores the space by building a tree of possible paths.	Suitable for high-dimensional spaces and dynamic environments.	Efficiently explores large spaces, can handle dynamic obstacles	Path quality may be suboptimal, requires post-processing to smooth paths.	[26]
Potential Field Methods	Uses artificial potential fields where the robot is attracted to the goal and repelled by obstacles.	Real-time obstacle avoidance and path planning.	Simple to implement, good for real-time applications	Local minima can trap the robot, not always guaranteed to find a path.	[27]
Hybrid A Algorithm*	Combines the efficiency of A* with the ability to handle non-holonomic constraints of real robots.	Autonomous driving, mobile robot navigation in complex environments.	Considers the kinematic constraints of robots, produces feasible paths.	More complex and computationally intensive than standard A*.	[28]
Dynamic Window Approach (DWA)	A local path planning algorithm that considers the robot's dynamics and kinematic constraints.	Mobile robot navigation in dynamic environments.	Produces smooth and feasible paths, suitable for real-time applications.	May not find the global optimal path, only focuses on local navigation.	[29]

VI.CONCLUSION

Each SLAM-based method has unique strengths and limitations that make it suitable for specific applications. ORB-SLAM excels in feature-rich environments, Hector SLAM is ideal for structured indoor settings, Graph SLAM is robust for large-scale mapping, and RL-integrated SLAM offers adaptability in dynamic scenarios. The choice of method depends on the specific requirements of the navigation task and the environment in which the robot operates. This summary encapsulates the pivotal role of machine learning in advancing mobile robot navigation, showcasing the blend of innovative techniques and practical

applications that drive the field forward. Each path planning algorithm has its strengths and weaknesses, making them suitable for different applications. Traditional algorithms like Dijkstra's and A* are robust and reliable for static environments, while sampling-based methods like PRM and RRT excel in high-dimensional and complex spaces. Potential field methods and the Dynamic Window Approach are beneficial for real-time applications, and hybrid approaches integrate multiple techniques to handle specific constraints. Advances in deep reinforcement learning offer promising new directions, particularly for dynamic and unstructured environments.

References

1. *K. A. A. Mustafa, "Towards continuous control for mobile robot navigation: A reinforcement learning and slam based approach." University of Twente, 2019.*
2. *X.-T. Truong and T. D. Ngo, "Toward socially aware robot navigation in dynamic and crowded environments: A proactive social motion model," IEEE Trans. Autom. Sci. Eng., vol. 14, no. 4, pp. 1743–1760, 2017.*
3. *R. S. Shokendra Dev Verma, Kirti Bhatia, Shalini Bhadola, "A Detailed Overview of Mobile NavigationProtocols," Int. J. Innov. Res. Comput. Commun. Eng., vol. 10, no. 6, pp. 5678–5684, 2022.*
4. *H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: part I," IEEE Robot. Autom. Mag., vol. 13, no. 2, pp. 99–110, 2006.*
5. *M. J. Mataric, "Behaviour-based control: Examples from navigation, learning, and group behaviour," J. Exp. Theor. Artif. Intell., vol. 9, no. 2–3, pp. 323–336, 1997.*
6. *A. Garulli, A. Giannitrapani, A. Rossi, and A. Vicino, "Mobile robot SLAM for line-based environment representation," in Proceedings of the 44th IEEE Conference on Decision and Control, 2005, pp. 2041–2046.*
7. *V. Nguyen, A. Harati, A. Martinelli, R. Siegwart, and N. Tomatis, "Orthogonal SLAM: a step toward lightweight indoor autonomous navigation," in 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2006, pp. 5007–5012.*
8. *E. H. C. Harik and A. Korsaeth, "Combining hector slam and artificial potential field for autonomous navigation inside a greenhouse," Robotics, vol. 7, no. 2, p. 22, 2018.*
9. *G. Sepulveda, J. C. Niebles, and A. Soto, "A deep learning based behavioral approach to indoor autonomous navigation," in 2018 IEEE international conference on robotics and automation (ICRA), 2018, pp. 4646–4653.*
10. *Y.-H. Kim, J.-I. Jang, and S. Yun, "End-to-end deep learning for autonomous navigation of mobile robot," in 2018 IEEE International Conference on Consumer Electronics (ICCE), 2018, pp. 1–6.*
11. *J. K. Wang, X. Q. Ding, H. Xia, Y. Wang, L. Tang, and R. Xiong, "A LiDAR based end to end controller for robot navigation using deep neural network," in 2017 IEEE International Conference on Unmanned Systems (ICUS), 2017, pp. 614–619.*
12. *R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "ORB-SLAM: a versatile and accurate monocular SLAM system," IEEE Trans. Robot., vol. 31, no. 5, pp. 1147–1163, 2015.*
13. *M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "FastSLAM: A factored solution to the simultaneous localization and mapping problem," Aaai/iaai, vol. 593598, 2002.*
14. *V. Mnih et al., "Human-level control through deep reinforcement learning," Nature, vol. 518, no. 7540, pp. 529–533, 2015.*
15. *S. Gu, E. Holly, T. P. Lillicrap, and S. Levine, "Deep reinforcement learning for robotic manipulation," arXiv Prepr. arXiv1610.00633, vol. 1, p. 1, 2016.*
16. *A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," Adv. Neural Inf. Process. Syst., vol. 25, 2012.*
17. *Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," IEEE Trans. Pattern Anal. Mach. Intell., vol. 35, no. 8, pp. 1798–1828, 2013.*
18. *C. Xiang et al., "Multi-sensor fusion and cooperative perception for autonomous driving: A review," IEEE Intell. Transp. Syst. Mag., 2023.*
19. *S. Ruder, "An overview of gradient descent optimization algorithms," arXiv Prepr. arXiv1609.04747, 2016.*
20. *D. Silver et al., "Mastering the game of Go with deep neural networks and tree search," Nature, vol. 529, no. 7587, pp. 484–489, 2016.*
21. *B. Zoph et al., "Rethinking pre-training and self-training," Adv. Neural Inf. Process. Syst., vol. 33, pp. 3833–3845, 2020.*
22. *S. Levine, C. Finn, T. Darrell, and P. Abbeel, "End-to-end training of deep visuomotor policies," J. Mach. Learn. Res., vol. 17, no. 39, pp. 1–40, 2016.*
23. *E. W. Dijkstra, "A note on two problems in connexion with graphs," in Edsger Wybe Dijkstra: His Life, Work, and Legacy, 2022, pp. 287–290.*
24. *P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," IEEE Trans. Syst. Sci. Cybern., vol. 4, no. 2, pp. 100–107, 1968.*
25. *L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," IEEE Trans. Robot. Autom., vol. 12, no. 4, pp. 566–580, 1996.*
26. *S. M. LaValle and J. J. Kuffner, "Rapidly-exploring random trees: Progress and prospects: Steven m. lavalle, iowa state university, a james j. kuffner, jr., university of tokyo, tokyo, japan," Algorithmic Comput. Robot., pp. 303–307, 2001.*
27. *O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," Int. J. Rob. Res., vol. 5, no. 1, pp. 90–98, 1986.*
28. *D. Dolgov, S. Thrun, M. Montemerlo, and J. Diebel, "Practical search techniques in path planning for autonomous driving," Ann Arbor, vol. 1001, no. 48105, pp. 18–80, 2008.*
29. *D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," IEEE Robot. Autom. Mag., vol. 4, no. 1, pp. 23–33, 1997.*