JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

ROBOTICS AND AUTONOMOUS SYSTEM

Autonomous Path Planning on Mobile Robotics by ensuring collision avoidance and optimal route selection

[1]Minki Ghosh, [2]Dr. Subir Kumar Das
[1]M-Tech Student (CSE), [2]Associate Professor,
[1]Computer Science and Engineering,
[1]Supreme Knowledge Foundation Group of Institutions, Mankundu, India

Abstract: Robotics and autonomous systems represent a dynamic and interdisciplinary field that merges mechanical, electrical, and computer engineering to create intelligent machines capable of performing tasks with minimal human intervention. This paper explores the key principles, advancements, and challenges in robotics and autonomous systems. It delves into the design and development of robotic systems, ranging from industrial automation to mobile and humanoid robots, emphasizing their ability to perceive, interpret, and adapt to complex environments. The integration of artificial intelligence and machine learning techniques plays a pivotal role in enhancing the autonomy of these systems, enabling them to learn from experience, make decisions, and continuously improve their performance. The paper also discusses the importance of sensor technologies, such as computer vision and inertial sensors, in providing robots with the ability to sense and navigate their surroundings. Autonomous systems are increasingly applied in various domains, including healthcare, agriculture, manufacturing, and transportation. The paper highlights the societal impacts of robotics and autonomy, addressing ethical considerations, safety standards, and the potential for job displacement. Furthermore, it explores ongoing research in swarm robotics, collaborative human-robot interaction, and the development of ethical frameworks for autonomous decision-making. As robotics and autonomous systems continue to evolve, their role in shaping the future of technology and society becomes more pronounced. This abstract provides a comprehensive overview of the current state of the field, outlining the challenges and opportunities that lie ahead in the quest to create intelligent, adaptive, and ethically responsible robotic systems.

I. INTRODUCTION

The domain of robotics and autonomous systems has emerged as a transformative force, seamlessly blending mechanical, electrical, and computer engineering to create intelligent machines capable of autonomously navigating and interacting with their environment. At its core, robotics seeks to transcend the boundaries of manual human intervention, employing a sophisticated integration of hardware and software components. From industrial arms and drones to advanced humanoid robots, these systems encapsulate the convergence of mechanical structures, actuators, and sensors with cutting-edge algorithms and artificial intelligence techniques. Recent technological advancements, particularly in sensor technologies like computer vision, and inertial sensors, have bolstered the perceptual capabilities of robots, enabling them to comprehend and adapt to complex surroundings. The integration of machine learning algorithms further empowers these systems to learn from experience, adapt to novel scenarios, and continuously optimize their performance. As robotics technology advances, it extends beyond traditional industrial applications to encompass diverse fields such as healthcare, agriculture, and exploration. However, this evolution is not without its challenges. The societal impacts of robotics and autonomy raise ethical considerations, including questions about job displacement, accountability, and the ethical use of autonomous decision-making systems. This introduction sets the stage for a comprehensive exploration of the fundamental principles, technological innovations, and societal implications within the dynamic realm of robotics and autonomous systems.

II. Motivation

True autonomous robots are intelligent machines that can perform tasks and operate in an environment independently, without human control or intervention .This will help the humans to work less and get the maximum benefit out of everything. As we can see humans are also using animals for different tasks, as example we can say in agriculture field, bulls are using by the farmers. In this case mobile robotics can also perform a great task for both humans and animals. This things motivate me to make the overview on mobile robotics.[Ref-1]

III. Contribution

Path planning is a very long process, here my contribution would be to make everything simple and easy to understand the term for everyone. To make everyone understand that mobile robotics can really make humans life a lot easier.

1. Literature Review

1.1 Human-Robot Interaction It is becoming more common for humans and robots to share a workspace. This has led to the need for improved human-robot communication and for awareness by the robot of what can be expected of the people around it and, similarly, by the people of what can be expected of the robots. An aspect of interaction with robots that is not unique to mobile robots is teaching them the tasks they are expected to accomplish.[Ref-2,4]

1.2 Localization and mapping In order for the robot to navigate successfully, it must determine its position in the workplace. So localization together with perception and motion control are key issues in robot navigation. Localization is closely related to representation. If an accurate GPS system could be installed on a robot, the localization problem would be solved. The robot would always know where it was. But at the moment, this system is not available or is not accurate enough to work with. In any case, localization implies not only knowing the robot's absolute position on Earth but also its relative position with respect to a target. The control system also plays a role. If the robot intends to reach a specific location, it needs an environment model or map so that it can plan a path to reach the target. This means that localization is a broad issue that includes not only determining the absolute position of the robot in space but also building a map and determining the robot's position relative to the map. Therefore, sensors (perception system) are fundamental in the task of localization. Any inaccuracy and incompleteness of the sensor will affect the robot's localization. Also, sensor noise and aliasing reduces the useful information picked. Uncertainty and error must be minimized. They must help in mapping the robot and its environment.

1.3 Path, trajectory, and motion planning Path planning is concerned with finding the best path in order for the mobile robot to reach the target without collision, thus allowing a mobile robot to navigate through obstacles from an initial configuration to another configuration. The temporal evolution of motion is neglected. No velocities and accelerations are considered. A more complete study, with broader objectives, is trajectory planning.[ref-5]

1.4 classical Approaches: Classical approaches to path planning have laid the foundation for subsequent research. This section discusses the following techniques:

- Potential Field Methods: These methods use attractive and repulsive forces to guide the robot towards the goal while avoiding obstacles. The limitations of potential field methods, such as local minima and sensitivity to parameter tuning, are highlighted.
- Cell Decomposition: Grid-based and graph-based cell decomposition methods divide the environment into cells to generate paths. The advantages and disadvantages of these techniques are explored.
- Voronoi Diagrams: Voronoi diagrams are employed to determine paths based on distance information. The utility of Voronoi diagrams in path planning is discussed.

1.5 Machine Learning Techniques: Machine learning techniques have shown promise in learning path planning policies and handling complex environments. This section covers the following techniques:

- Reinforcement Learning (RL): RL approaches, such as Q-learning and deep RL algorithms, are employed to learn path planning policies. The challenges and opportunities of RL in path planning are discussed.
- Deep Learning: Deep neural networks are used for end-to-end path planning, where raw sensor data serves as input. The advantages and limitations of deep learning in path planning are explored.[ref-6,7]

1.6 Optimization-Based Approaches: Optimization-based approaches leverage mathematical optimization techniques to find optimal paths. This section covers the following techniques:

- A* Algorithm: The A* algorithm and its variants, such as D* and D*-Lite, combine graph search and heuristics to find optimal paths. The strengths and limitations of A* algorithms are discussed.
- Dijkstra's Algorithm: Dijkstra's algorithm, a classic graph search algorithm, is relevant to path planning, particularly in static environments. The application of Dijkstra's algorithm and its variations is explored.
- Ant Colony Optimization (ACO): ACO algorithms are inspired by the foraging behavior of ants and have been applied to path planning. The use of ACO for finding optimal paths is discussed. [ref-8]

2. Problem Statement and Objective

2.1 Problem statement

The problem statement of mobile robotics path planning involves finding an optimal or near optimal path for mobile robot to navigate from a given starting point to a desired goal location in a dynamic environment. This problem is particularly challenging because the robot must take into account various factors such as obstacle, robot's physical capabilities, and potential constraints or limitations. In the case of a human's regular life, there are lots of things which can be risky as well as time consuming for the human being. Robot can make that so much easier and will also take less time to finish it. One robot is enough to do lots of work.[ref-9,10,11]

2.2 Objective

The objective of a path planning problem is to find an optimum or near-optimum path (safest, shortest, and smoothest) without colliding with a problem. For mobile robots, the aim of path planning is to find a feasible path in a specific environment. This path begins at the starting point (S) and ends at the target point (T). The objectives of mobile robotics path planning can vary depending on the specific application and requirements.

1. Collision Avoidance: One of the primary objectives of mobile robotics path planning is to ensure collision-free navigation. The path planning algorithm aims to generate a trajectory that avoids obstacles, static or dynamic, in the environment. By considering the robot's current position, desired destination, and the surrounding obstacles, the algorithm plans a safe and feasible path.

2. Optimized Trajectories: Another objective is to generate optimized trajectories that minimize travel time, energy consumption, or any other defined cost function. The path planning algorithm considers factors such as robot dynamics, environment characteristics, and task requirements to determine the most efficient path to the goal. Optimization criteria can vary depending on the specific application, ranging from minimizing distance traveled to maximizing energy efficiency.

3. Smooth and Stable Motion: Mobile robots should exhibit smooth and stable motion while navigating their paths. Jittery or jerky movements can impact robot stability, cause 13 wear and tear on mechanical components, and potentially disrupt any payload or sensor mounted on the robot. Path planning algorithms aim to generate trajectories that result in smooth and stable motion for the robot, ensuring better overall performance and safety.

4. Adaptability to Dynamic Environments: Mobile robots often operate in dynamic environments where the presence of obstacles or other agents can change over time. The path planning algorithm should be capable of adapting to such changes in real-time and generating updated paths accordingly. This objective involves incorporating sensor feedback, obstacle detection, and dynamic obstacle avoidance strategies to ensure the robot can successfully navigate through changing environments.[Ref-12,13]

3. Design and Methodology

The design and methodology of mobile robotics path planning involve several stages and approaches. Here is an overview of the design process and methodologies which are commonly used.

1. Problem Formulation: The first step is to define the problem and its requirements. This includes specifying the robot's initial position, the desired destination, and any constraints or objectives. It is important to consider factors such as obstacle avoidance, optimality, real-time performance, and any task-specific requirements.

2. Environment Modeling: Create a model of the robot's environment, including obstacles, boundaries, and any relevant features. The environment model can be created using sensor data or predefined maps. Different representations, such as occupancy grids, point clouds, or polygonal models, can be used depending on the available sensor information and the complexity of the environment.

3. **Sensing and Perception:** Determine the sensors required for environment perception. This can include cameras, lidar, radar, or other range-finding sensors. Sensor data processing techniques, such as point cloud segmentation or object detection algorithms, may be employed to extract meaningful information about obstacles and their characteristics. 4. Path Representation: Choose a suitable representation to define the robot's path. This can include grid-based representations, graphs, splines, or potential fields. The path representation should capture the spatial information necessary for navigation and collision avoidance.[ref-15,16,1,18]

5. Motion Planning: Once a path is generated, the motion planning component determines how the robot should move along the path. This involves considering the robot's kinematics, dynamics, and any constraints on its motion. Motion planning algorithms, such as trajectory generation or control-based approaches, determine the control inputs required to achieve smooth and stable motion along the path.

6. Collision Avoidance: Implement collision avoidance techniques to ensure safe navigation. This can involve reactive approaches, such as potential fields or velocity obstacles, or predictive methods like model predictive control (MPC). Collision avoidance strategies should take into account the robot's current position, velocity, and the perceived obstacles to generate appropriate control actions.

7. Real-Time Adaptability: Account for real-time adaptability to handle dynamic environments or changing conditions. This may involve integrating feedback loops, sensor fusion techniques, or online preplanning strategies. The path planning algorithm should be able to quickly update the planned path and adjust the robot's trajectory based on sensor feedback or environmental changes. 8. Validation and Evaluation: Thoroughly test and evaluate the path planning algorithm using appropriate metrics. This can include metrics such as path length, computation time, and smoothness of motion, energy consumption, or task-specific objectives. Validation can be done through simulations, virtual environments, or physical testing in representative scenarios. 9. Optimization and Iteration: Based on the evaluation results, optimize and refine the path planning algorithm. This may involve tuning parameters, improving computational efficiency, or incorporating advanced techniques from related research areas. The design process should be iterative, allowing for improvements and adjustments based on practical implementation and performance feedback.[ref-19]

8. Validation and Evaluation: Thoroughly test and evaluate the path planning algorithm using appropriate metrics. This can include metrics such as path length, computation time, and smoothness of motion, energy consumption, or task-specific objectives. Validation can be done through simulations, virtual environments, or physical testing in representative scenarios.

9. Optimization and Iteration: Based on the evaluation results, optimize and refine the path planning algorithm. This may involve tuning parameters, improving computational efficiency, or incorporating advanced techniques from related research areas. The design process should be iterative, allowing for improvements and adjustments based on practical implementation and performance feedback.[ref-19]

10. Algorithm: Dijkstra's Shortest Path Algorithm

1. Initialization:

- Set the distance to the source node as 0, and the distances to all other nodes as infinity.
- Mark all nodes as unvisited.
- set the source node as the current node.

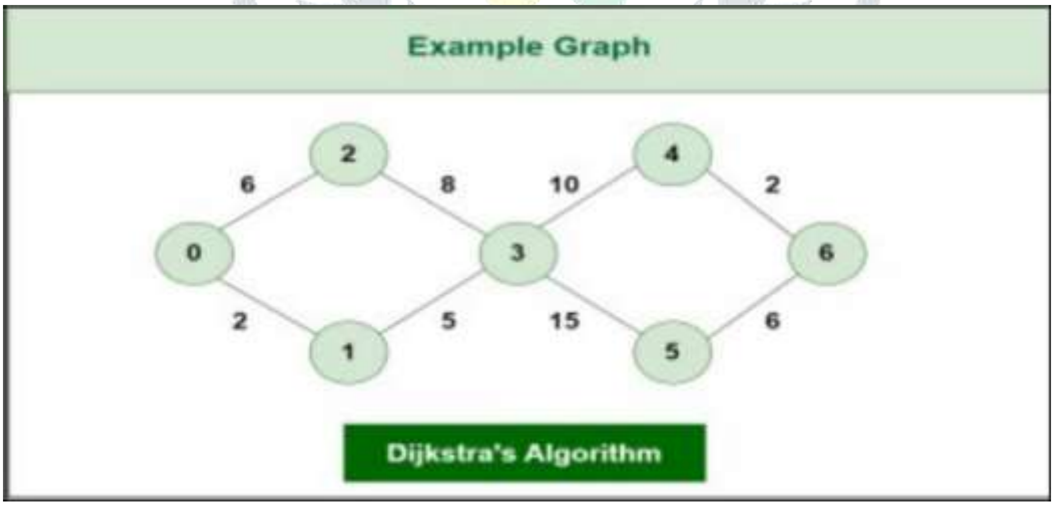
2. Iteration:

- for the current node, consider all of its neighbors (nodes directly connected by an edge).
- Calculate the tentative distance from the source to each neighbor through the current node.
- If the calculated distance is less than the previously known distance, update the distance.

3. Formula: tentative_distance=current_distance+edge_weight

- Mark the current node as visited to avoid revisiting it.
- Set the selected node as the new current node and go back to step 2.
- The algorithm terminates when the destination node is marked as visited, or when all reachable nodes have been visited.[ref-19]

Example: Consider the below graph:



For this graph, we will assume that the weight of the edges represents the distance between Two nodes.

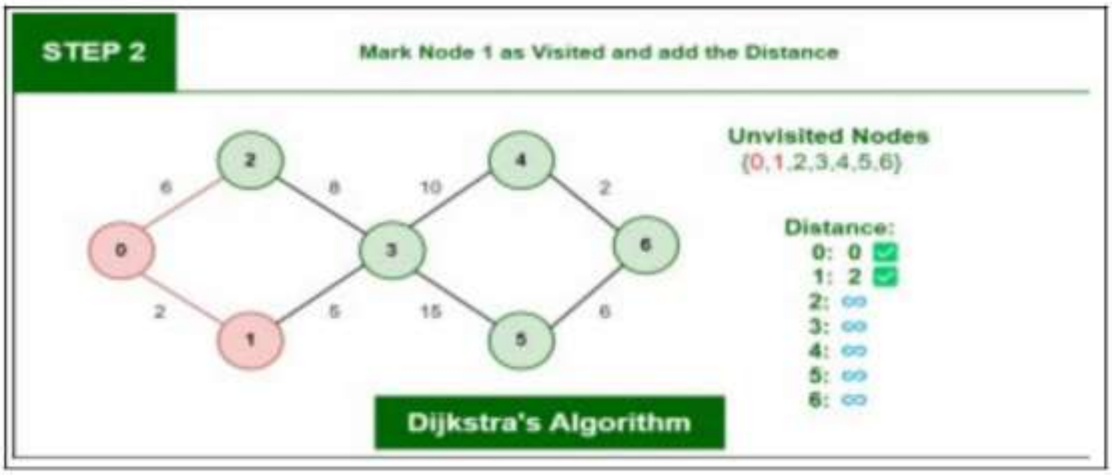
As, we can see we have the shortest path from,
Node 0 to Node 1, from
Node 0 to Node 2, from
Node 0 to Node 3, from
Node 0 to Node 4, from
Node 0 to Node 6.

Initially we have a set of resources given below:

- The Distance from the source node to itself is 0. In this example the source node is 0.
- The distance from the source node to all other nodes is unknown so we mark all of them as infinity.
Example: 0 -> 0, 1-> ∞,2-> ∞,3-> ∞,4-> ∞,5-> ∞,6-> ∞.
- We'll also have an array of unvisited elements that will keep track of unvisited or Unmarked Nodes.
- Algorithm will complete when all the nodes marked as visited and the distance between them are added to the path. Unvisited Nodes:- 0 1 2 3 4 5 6.Step 1: Start from Node 0 and mark Node as visited.

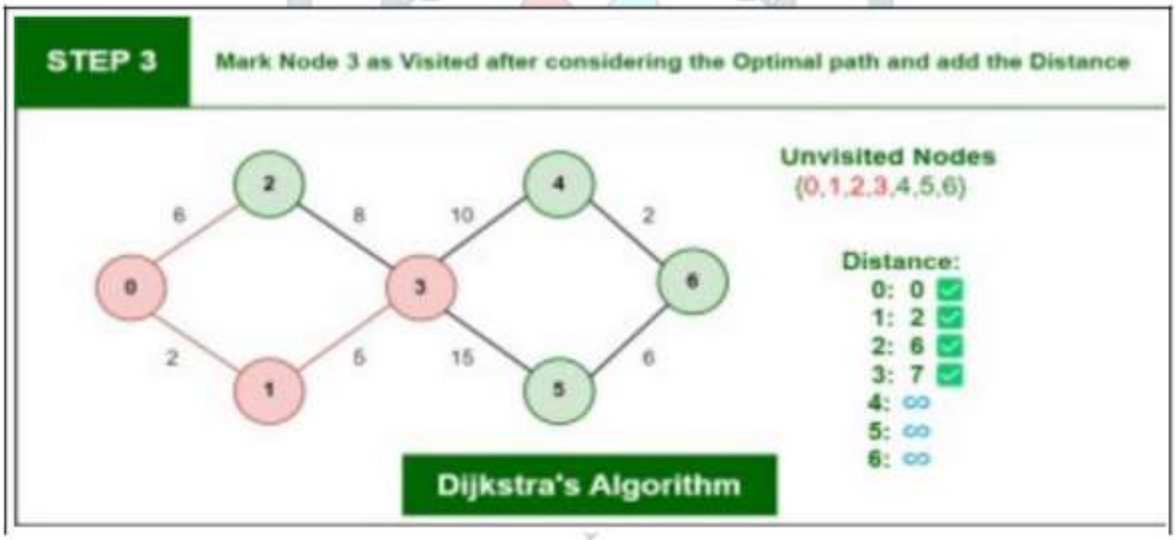
Step 2: Check for adjacent Nodes. Now we have two choices (Either choose Node1 with distance 2 or either choose Node 2 with distance 6) and choose Node with minimum distance. In this step Node 1 is the Minimum distance adjacent Node, so mark it as visited and add up the distance.

Distance: Node 0 -> Node 1 = 2



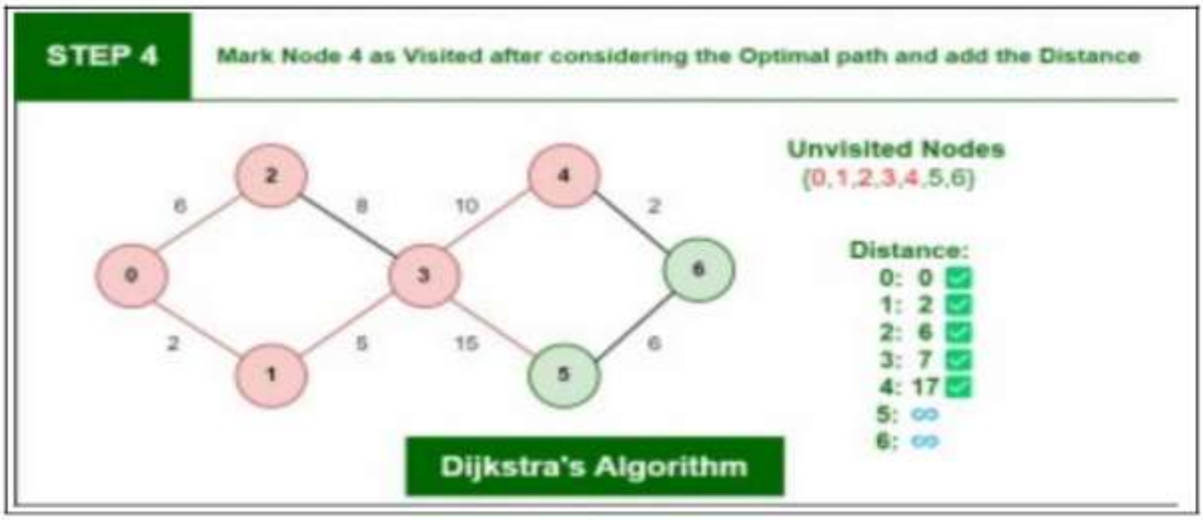
Step 3: Then Move Forward and check for adjacent Node which is Node 3, so marked it as visited and add up the distance, now the distance will be:

Distance: Node 0 -> Node 1 -> Node 3 = 2 + 5 = 7



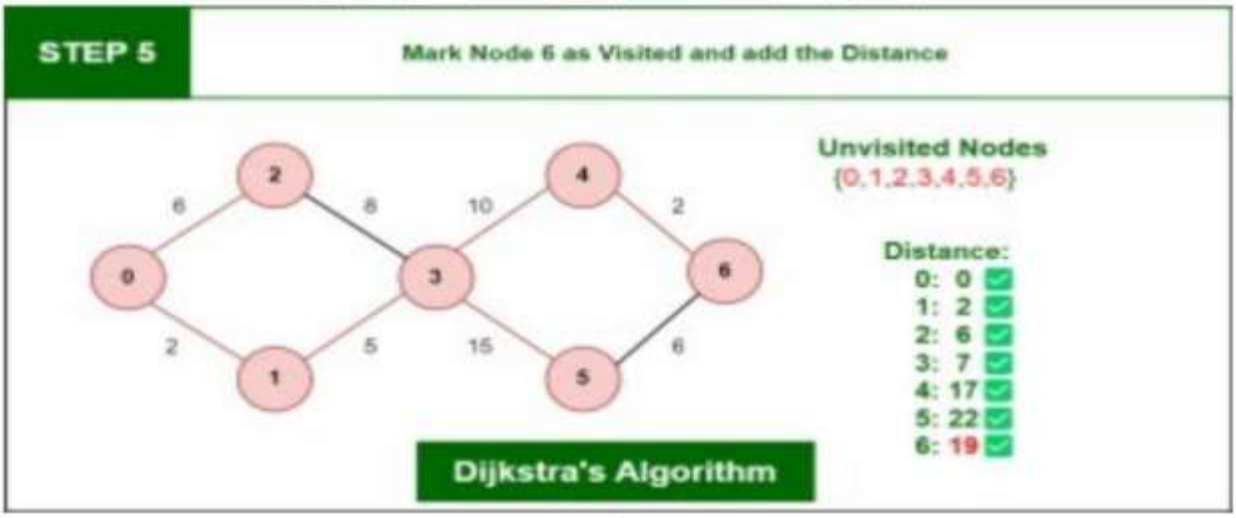
Step 4: Again we have two choices for adjacent Nodes (Either we can choose Node 4 with distance 10 or either we can choose Node 5 with distance 15 so choose Node with minimum distance. In this step Node 4 is the Minimum distance adjacent to Node, so mark it as visited and add up the distance.

Distance: Node 0 -> Node 1 -> Node 3 -> Node 4 = 2 + 5 + 10 = 17



Step 5: Again, Move Forward and check for adjacent Node which is Node 6, so marked it as visited and add up the distance, now the distance will be:

Distance: Node 0 -> Node 1 -> Node 3 -> Node 4 -> Node 6 = 2 + 5 + 10 + 2 = 19



So, the Shortest Distance from the Source Vertex is 19 which is the optimal one.

11. Optimization Criteria

There are many factors that must be considered in the optimization criteria for planning a mobile path. Three commonly used optimization criteria are listed in the following.

11.1 Path Length

The path length D [15,16] is defined as D = ∑n−1 q(xj+1 − xj)2 + (yj+1 − yj)2

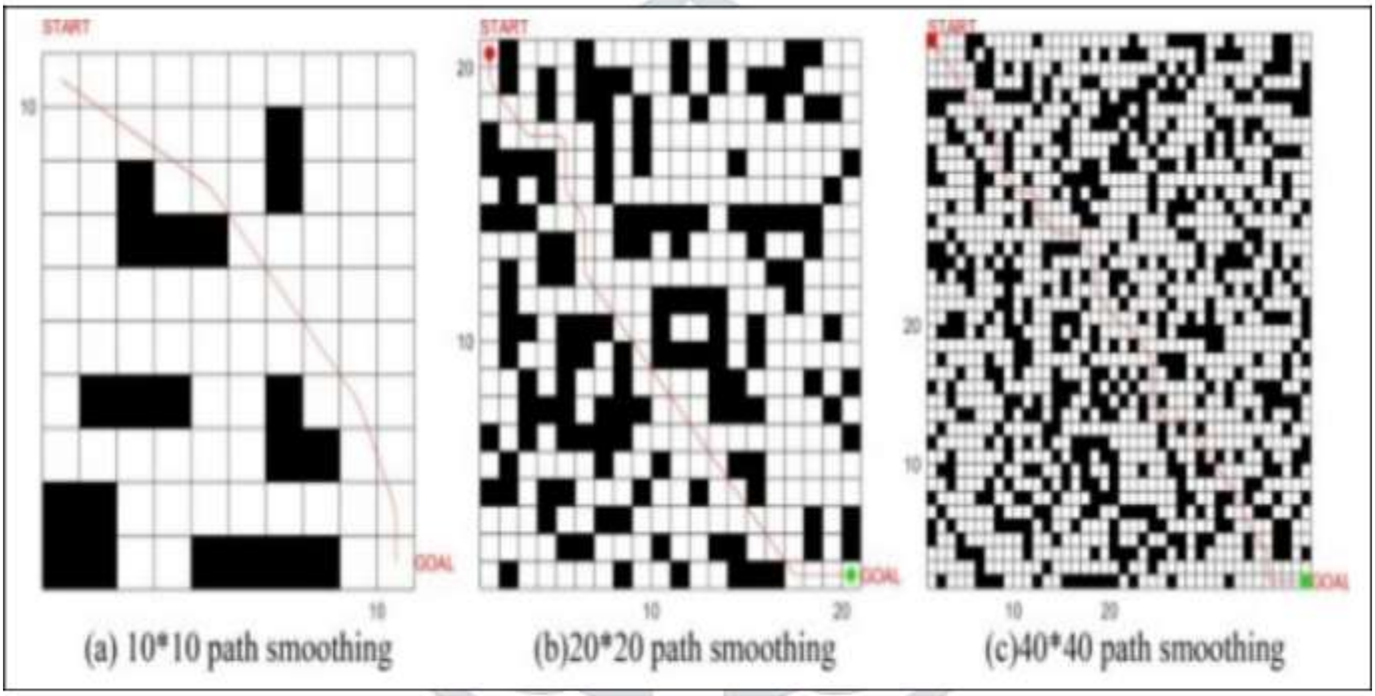
Where, xj and yj are the values of the X coordinate axis and Y coordinate axis of the nodes j.

11.2 Smoothness

The smoothness S [15] is defined as

S = α · 1 − DAl ! + β · 1 − Smin !
Nf − 1 Nf

where, α and β are weighted coefficients, DAl is the number of angles of deflection larger than the desired variable, Nf is the total number of path segments, Smin is the number of segments with the smallest number of path segments in the path.[ref-19]



(a) 10*10 path smoothing (b)20*20 path smoothing (c)40*40 path smoothing

Simulation of path smoothing in different raster environments.

11.3 Safety Degree

The safety degree SD [15] is defined as
SD = ∑n−1 Cj = (n 0, dj d≥ λ)
Where, di is the minimal distance between the i-th segment and its nearest obstacle, and λ is the threshold of the safety degree.

4. Results and interpretations

The results and interpretations of mobile robotics path planning can provide valuable insights into the performance and effectiveness of the algorithm in achieving its objectives. Here are some possible outcomes and interpretations:

1. Planned Paths: The most immediate result of mobile robotics path planning is the generation of planned paths from the robot's initial position to the desired destination. The paths can be visualized and analyzed to assess their quality, such as their smoothness, length, and adherence to the specified objectives. A well-designed path planning algorithm should generate paths that efficiently navigate the robot through the environment while avoiding obstacles.

2. Efficiency and Optimality: Path planning algorithms often aim to optimize certain criteria, such as minimizing distance traveled, reducing energy consumption, or maximizing task efficiency. The planned paths can be analyzed to evaluate the algorithm's success in achieving these objectives. If the paths are shorter, more energy-efficient, or result in faster task completion compared to alternative routes, it suggests that the path planning algorithm is effective in optimizing the desired criteria.

3. Real-Time Adaptability: Mobile robotics path planning may need to adapt to dynamic environments or changing conditions. The ability of the algorithm to handle such scenarios can be assessed by introducing modifications to the environment or obstacles during the robot's navigation. If the path planning algorithm successfully updates the planned path in real -time to avoid newly introduced obstacles or accommodate changing conditions, it demonstrates its adaptability and responsiveness.

4. Performance Metrics: Various performance metrics can be used to evaluate the path planning algorithm. These metrics may include computation time, computational complexity, memory usage, or convergence properties. By analyzing these metrics, one can gain insights into the algorithm's efficiency, scalability, and computational requirements. It helps determine if the algorithm can operate in real-time and handle complex environments without exceeding computational resources.

5. Task-Specific Evaluation: Depending on the application, specific task-specific metrics can be used to assess the performance of mobile robotics path planning. For example, in surveillance scenarios, metrics like coverage rate, observation quality, or response time can be used. In industrial settings, metrics related to productivity, collision rate, or throughput can be evaluated. Task-specific evaluations provide a more targeted assessment of how well the path planning algorithm fulfills the requirements of the specific application.[Ref-6,7]

5. Conclusion and Future Scope

Mobile robotics path planning plays a crucial role in enabling safe and efficient navigation for robots in various applications. The design and implementation of path planning algorithms involve considerations such as environment modelling, sensor integration, path representation, path search, motion planning, collision avoidance, real-time adaptability, and performance evaluation. By generating optimized paths and ensuring collision avoidance, path planning algorithms contribute to successful robot navigation. The future of mobile robotics path planning lies in the development of advanced algorithms, incorporating learning-based approaches, considering multi-robot collaboration, integrating human-robot interaction, enhancing robustness and adaptability, improving scalability and efficiency, and focusing on real-world deployment. Continued research and innovation in these areas will contribute to the advancement and widespread adoption of mobile robotics path planning techniques in various domains.

6. Acknowledgement

Robotics is a critical component of autonomous robot navigation. As mobile robots are interact and collaborate with humans in various domains, path planning will play a huge role in enabling seamless and effective human-robot cooperation.
We would like to thank Supreme knowledge foundation group of institution for sharing all the data regarding mobile robotics path planning used in this study.

7. References

1. Zhang, H. Y., Lin, W. M., & Chen, A. X. (2018). Path planning for the mobile robot: A review. Symmetry, 10(10), 450..

2. Viet, H. H., An, S. H., & Chung, T. C. (2013). Dyna-Q-based vector direction for path planning problem of autonomous mobile robots in unknown environments. Advanced Robotics, 27(3), 159-173..

3. Liu, Y. H., & Arimoto, S. (1991, January). Proposal of tangent graph and extended tangent graph for path planning of mobile robots. In Proceedings. 1991 IEEE International Conference on Robotics and Automation (pp. 312-313). IEEE Computer Society.

4. Henrich, D. (1997). Fast motion planning by parallel processing–a review. Journal of Intelligent and Robotic Systems, 20, 45-69..

5. Habib, M. K., & Asama, H. (1991, November). Efficient method to generate collision free paths for an autonomous mobile robot based on new free space structuring approach. In Proceedings IROS'91: IEEE/RSJ International Workshop on Intelligent Robots and Systems' 91 (pp. 563-567). IEEE.

6. Chen, D. Z., Szczerba, R. J., & Uhran, J. J. (1997). A framed-quadtree approach for determining Euclidean shortest paths in a 2-D environment. IEEE Transactions on robotics and automation, 13(5), 668-681..

7. Kavraki, L., & Latombe, J. C. (1994, May). Randomized preprocessing of configuration for fast path planning. In Proceedings of the 1994 IEEE International Conference on Robotics and Automation (pp. 2138-2145). IEEE..

8. Nayeem, G.M.; Fan, M.; Akhter, Y. A Time-Varying Adaptive Inertia Weight based Modified PSO Algorithm for UAV Path Planning. In Proceedings of the 2021 2nd International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST), Dhaka, Bangladesh, 5–7 January 2021; IEEE: New York, NY, USA, 2021; pp. 573–576. [Google Scholar] [CrossRef]

9.Dai, H.; Chen, D.; Zheng, Z. Effects of Random Values for Particle Swarm Optimization Algorithm .Algorithms2018,11, 23. [Google Scholar] [CrossRef]

10. Maina, R.M.; Lang'at, P.K.; Kihato, P.K. Collaborative beamforming in wireless sensor networks using a novel particle swarm optimization algorithm variant.Heliyon2021,7, e08247. [Google Scholar] [CrossRef] [PubMed]

11. Jingjing, H.; Xun, L.; Wenzhe, M.; Xin, Y.; Dong, Y. Path Planning Method for Mobile Robot Based on Multiple Improved PSO. In Proceedings of the 2021 40th Chinese Control 25 Conference (CCC), Shanghai, China, 26–28 July 2021; IEEE: New York, NY, USA, 2021; pp. 1485–1489. [Google Scholar] [CrossRef]

12. Liu, W.; Wang, Z.; Yuan, Y.; Zeng, N.; Hone, K.; Liu, X. A novel sigmoid-functionbased adaptive weighted particle swarm optimizer.IEEE Trans. Cybern.2019,51, 1085–1093. [Google Scholar] [CrossRef] [PubMed]

13. Liang, B.; Zhao, Y.; Li, Y. A hybrid particle swarm optimization with crisscross learning strategy.Eng. Appl. Artif. Intell.2021,105, 104418. [Google Scholar] [CrossRef]

14. Phung, M.D.; Ha, Q.P. Safety-enhanced UAV path planning with spherical vector-based particle swarm optimization.Appl. Soft Comput.\2021,107, 107376. [Google Scholar] [CrossRef]

15. Guo, X.; Ji, M.; Zhao, Z.; Wen, D.; Zhang, W. Global path planning and multi-objective path control for unmanned surface vehicle based on modified particle swarm optimization (PSO) algorithm.Ocean Eng.2020,216, 107693. [Google Scholar] [CrossRef]

16. Fu, Y.; Ding, M.; Zhou, C. Phase angle-encoded and quantum-behaved particle swarm optimization applied to three-dimensional route planning for UAV.IEEE Trans. Syst. Man Cybern.-Part A Syst. Hum.2011,42, 511–526. [Google Scholar] [CrossRef]

17. Xu, Z.; Zhang, E.Z.; Chen, Q.W. Rotary unmanned aerial vehicles path planning in rough terrain based on multi -objective particle swarm optimization.J. Syst. Eng. Electron.2020,31, 130–141. [Google Scholar] [CrossRef]

18.Thammachantuek, I.; Ketcham, M. Path planning for autonomous mobile robots using multi-objective evolutionary particle swarm optimization.PLoS ONE2022,17, e0271924. [Google Scholar] [CrossRef] [PubMed]

19. Sathiya, V.; Chinnadurai, M.; Ramabalan, S. Mobile robot path planning using fuzzy enhanced improved Multi-Objective particle swarm optimization (FIMOPSO).Expert Syst. Appl.2022,198, 116875. [Google Scholar] [CrossRef]