MANTECH Publications

International Journal of Robotics and Autonomous Systems
Volume 5, Issue 2, July-December, 2020

Navigating the Future: Simple Path Planning Algorithms for Autonomous Robots

Amitav Sengupta
Email Id: *amitav.sengupta@techlabs.org*

Riya Mehra
Email Id: *riya.mehra@robosim.ai*

Abstract

Path planning is a critical component in autonomous robotic navigation, enabling robots to determine an efficient route from their current position to a designated goal. This paper explores the core principles and practical applications of simple path planning algorithms, including Dijkstra's algorithm, A (A-star), and the Bug algorithm. By comparing their computational requirements, path efficiency, and suitability for different environments, this paper provides a foundational guide for students, researchers, and developers entering the field of mobile robotics. Emphasis is placed on simplicity, computational efficiency, and adaptability to real-time robotic systems with limited hardware resources.*

Keywords: *Path Planning, Robotics, Dijkstra, A*, Bug Algorithm, Autonomous Navigation*

INTRODUCTION

Robots are increasingly expected to operate in dynamic and unstructured environments, making autonomous navigation a crucial feature. Path planning allows robots to compute a collision-free trajectory from their start point to a target. This paper introduces simple path planning techniques that are efficient and easy to implement in hardware-constrained robotic platforms. Understanding these algorithms is fundamental to both academic research and industrial robotics applications.

PATH PLANNING CONCEPTS

- **What Is Path Planning?**

  Path planning is the computational process by which a robot determines an optimal or feasible path to travel from an origin to a destination. It involves assessing the environment, evaluating potential paths, and selecting the best path based on criteria such as distance, time, and safety.

- **Categories Of Path Planning**

  Path planning algorithms fall into two broad categories: global and local planning. Global planning assumes prior knowledge of the environment and attempts to find the optimal path. Local planning, by contrast, deals with real-time obstacles and sensor-based path calculation.

COMMON SIMPLE PATH PLANNING ALGORITHMS

- **Dijkstra'S Algorithm**

  This classic graph-based algorithm calculates the shortest path from a source node to all other nodes. It is widely used for grid maps but is computationally expensive when applied to large or dynamic environments.

- **A* (A-Star) Algorithm**

  An improvement over Dijkstra's algorithm, A* introduces heuristics to estimate the cost to the goal, significantly reducing computation time. It is optimal and complete, making it ideal for mobile robots in static maps.

- **Bug Algorithm**

  The Bug family of algorithms (e.g., Bug1, Bug2) uses minimal computation and sensor feedback to navigate toward a goal while avoiding obstacles. Though less efficient in path length, they are ideal for simple robots with limited processing capabilities.

COMPARISON OF PATH PLANNING ALGORITHMS

Table 1: Comparison of key path planning algorithms by type, efficiency, and practical use-case.

Algorithm	Type	Efficiency	Suitable For
Dijkstra's	Global	Low	Static, structured environments
A*	Global	High	Grid maps with known heuristics
Bug Algorithm	Local	Moderate	Dynamic, unknown environments

IMPLEMENTATION CONSIDERATIONS

When choosing a path planning algorithm for implementation, factors such as hardware limitations, memory usage, response time, and environmental dynamics must be evaluated. For example, A* may not perform well in high-speed scenarios where computation time is critical. On the other hand, Bug algorithms, although suboptimal in path length, perform well under limited sensing and computational environments.

APPLICATIONS IN REAL-WORLD ROBOTICS

Simple path planning algorithms are deployed in various real-world scenarios such as warehouse robots (A*), robotic vacuum cleaners (Bug Algorithm), and autonomous delivery systems (Dijkstra's). Their reliability and simplicity make them suitable for educational projects, low-cost robotics, and rapid prototyping in industrial robotics.

CONCLUSION

This paper has presented an overview of simple yet powerful path planning algorithms used in robotic navigation. While advanced techniques exist, these foundational algorithms remain relevant due to their clarity, efficiency, and low implementation cost. As the field of robotics continues to evolve, the importance of understanding these core strategies becomes even more critical for new learners and developers.

REFERENCES

1. E. W. Dijkstra, "A Note on Two Problems in Connexion with Graphs," NumerischeMathematik, vol. 1, no. 1, pp. 269–271, 1959.
2. P. E. Hart, N. J. Nilsson, and B. Raphael, "A Formal Basis for the Heuristic Determination of Minimum Cost Paths," IEEE Transactions on Systems Science and Cybernetics, vol. SSC-4, no. 2, pp. 100–107, 1968.
3. J. Latombe, "Robot Motion Planning," Springer Science & Business Media, 2012.
4. H. Choset et al., "Principles of Robot Motion: Theory, Algorithms, and Implementations," MIT Press, 2005.