Application of Route Planning and Obstacle Avoidance System

Weiheng Wang[1*]

[1]Affiliated High School of South China Normal University International Department, Guangzhou, 510000, China

Abstract. Recently, research on route planning technologies and obstacle avoidance has gained increasing attention with the rapid development of artificial intelligence. These fundamental technologies are applied in operating systems to support robotic navigation and transfer tasks. Nonetheless, challenges such as lag and uncertainty in predictions remain unsolved, limiting the efficiency and stability of practical applications. To address these issues, scientists continue to explore optimization methods. This paper will introduce such technologies by explaining their definitions, working principles, and coding implementations. Furthermore, a design scheme will be presented as a practical reference, illustrating how these concepts can be applied dynamically. Through this discussion, the paper seeks to encourage further exploration and inspire potential solutions or revisions to overcome existing technical barriers. Moreover, the paper shares the basic theory for helping advanced learning of the technologies, not for the comprehensive application of the operating system, so any sharing of the program is to help appreciate the application of technologies.

1 Introduction

With the more complicated applications of the route planning system, the algorithms such as Dijkstra, A* are limited by the huge amount of calculated quantities. Modern technology has optimized the efficiency of algorithms through the following methods: hierarchical network, dividing the road network into different levels to significantly reduce the amount of calculation; contraction hierarchies, optimizing the heuristic search to further accelerate the search speed; and multi-objective programming, supporting complex planning across modes to provide the optimal combination solution. In addition, the application of artificial intelligence and machine learning is playing a crucial role in route planning with its immense computational power.

While applying the algorithm in the operating system, there are still some challenges that need to be exactly solved: First, in dynamic and unstructured environments, such as dense crowds, road construction zones, and adverse weather conditions, algorithms often struggle to process complex variables promptly, leading to instability in route planning. Secondly, high-precision route planning requires a large amount of computing resources. Thirdly, many algorithms perform well in specific experimental environments, but their performance tends

* Corresponding author: henry486w@gmail.com

to decline in new environments. The latest algorithms may be able to ensure the high accuracy of route planning in fixed scenes, but it's more difficult to conduct identically in different scenes, such as streets with pedestrians and vehicles.

Similarly, the obstacle avoidance technology probably faces the same problems. The high cost of calculation and an intricate environment with unpredictable situations impedes the university.

2 Route planning

2.1 Definition

Route planning is defined as the process of selecting a minimum cost route from a starting point to a destination through a transport network [1]. This problem is one of the most fundamental and challenging issues in the field of robotics. Ideally, a path planning algorithm should be able to identify a collision-free route whenever one exists and continue to find a suboptimal path when no optimal path is available. However, the route planning problem is NP-complete. Various computational results have been established for the complexities of the problem. In fact, generating collision-free paths is generally intricate, no matter how moderate the advanced robotic systems are. Therefore, contemporary route planning algorithms aim to achieve a balance between completeness (often adopting relaxed or probabilistic forms of completeness) and computational efficiency, with the objective of finding feasible paths for most practical scenes [2, 3].

2.2 Work principle

Route planning first requires clear input information: starting point, ending point, environmental map (map structure or configuration space), obstacle information, and optimization goals (such as distance, time, cost, etc.). Although the robot moves physically within its workspace, the route planning problem can be more effectively addressed by operating directly in the configuration space. To plan collision-free routes, it is necessary to partition the configuration space into two distinct sets: the set of configurations without collision, denoted as δ_{free}, and the set of configurations that result in collisions, denoted as δ_{obst} (configuration space obstacle region). The set of points in the workspace occupied by the robot in configuration q is represented by R(q), while the obstacle region in the workspace is denoted by ε.

The set δ_{obst} is defined as,

$$\delta_{obst} = \{q \in \delta \mid R(q) \cap \varepsilon = \emptyset\} \tag{1}$$

$$\delta_{obst} = \delta / \delta_{free} \tag{2}$$

Defining a free route as a continuous map from the unit interval (or convenient intervals that represent route length) as γ:[0,1] $\rightarrow \delta_{free}$, such that $q(0) = q_{init}$ and $q(1) = q_{goal}$ [2].

2.3 The A star (A*) algorithm

A* algorithm is one of the most famous route planning algorithms, which can be applied to configuration space as the value evaluates each cell in the configuration space. This algorithm utilizes heuristic searching sufficiently, based on the aim of searching for the shortest route,

$$f(v) = h(v) + g(v) \tag{3}$$

In this equation, h(v) is the heuristic distance of the cell to the g, g(v) is the length of the route from the initial cell to the goal cell, f(v) is the evaluation of each adjacent cell of the reached cell (the cell with the lowest value of f(v) is chosen as the next one in the sequence). The advantage of this algorithm is that the distances used as a criterion can be utilized and altered, offering flexible modifications of the basic principle [4].

Calculate the shortest distance,

$$d = | x1 - x2 | + | y1 - y2 | \tag{4}$$

The core of A* lies in guiding the search direction through a function. While ensuring the discovery of the optimal route, it significantly reduces unnecessary exploration, and its efficiency is much higher than that of blind search.

3 Obstacle avoidance

3.1 Definition

Obstacle avoidance refers to the ability of a mobile robot to prevent collisions with obstacles present in its task space. Even when the environment is pre-mapped, obstacle avoidance behaviour remains essential because accurate metric information is difficult to obtain, increasing the risk of collisions. In dynamically changing environments containing both motorial and static obstacles, obstacle avoidance becomes significantly more critical than in stationary settings, as the robot must respond promptly to newly detected and unpredictable conditions.

The central idea of obstacle avoidance is that, based on the robot's sensory observations, the self-organizing map should learn an effective collision avoidance behavior. These observations are gathered both from selected training examples and from the robot's own experiences during navigation. Specifically, a learning mobile robot can benefit from training examples and can also improve its performance autonomously through test and error when no external guidance is available [5].

3.2 Work principle

An Autonomous Mobile Robot (AMR) operates with minimal human intervention and is designed to follow predefined routes in either indoor or outdoor environments [6]. AMRs are equipped with advanced sensors that continuously collect environmental data, enabling real-time situational awareness. The true intelligence of an AMR lies in its ability to interpret sensor data and respond accordingly, though sensing obstacles is an essential capability. This real-time processing capability is enabled by simultaneous localization and mapping (SLAM) technology.

The artificial potential field (APF) method is extensively used in route planning issues. The APF method treats the robot as a particle moving within an artificial potential field. The field consists of an attractive component directed toward the target or global goal, and repulsive components associated with surrounding obstacles. The APF method offers a simple yet highly effective strategy for route planning issues and is particularly efficient for online motion trail generation. However, in its simple form, it exhibits many drawbacks, such as the oscillatory behaviour of robots in narrow spaces and sensitivity to local minima [7]. There is a modified APF model [8].

In the planar space, the modified attraction field function is defined as follows,

$$\mathrm{U_A} = \frac{1}{2}\mathrm{k_S}\mathrm{R_A}^2 \tag{5}$$

Also, it can be expressed as,

$$\boldsymbol{U}_{\mathrm{R}}(\mathrm{X}) = \begin{cases} 0.5\mathrm{Z}(\frac{1}{\mathrm{R_{rep}}} - \frac{1}{\mathrm{G_0}})^2 \mathrm{R_A}^{\mathrm{N}} & \mathrm{R_{rep}} \leq \mathrm{G_0} \\ 0 & \mathrm{R_{rep}} > \mathrm{G_0} \end{cases} \tag{6}$$

$\mathrm{R_A}$ represents the shortest distance between the robot and the target in the planar space, and $\mathrm{R_A}^{\mathrm{N}}$ is a regulative factor. The attractive force is derived as the gradient of the attraction field,

$$\mathrm{F_A} = -\nabla\left[\frac{1}{2}\mathrm{k_S}\mathrm{R_A}^2\right] = \mathrm{k_S}\mathrm{R_A} \tag{7}$$

Similarly, the repulsive force is obtained as the repulsion field's negative gradient, given by the following expression,

$$\boldsymbol{F}_{\mathrm{R}} = \begin{cases} \mathrm{Z} \times \boldsymbol{F}_{\mathrm{R1}}(\mathrm{X}) + \mathrm{M} \times \mathrm{Z} \times \boldsymbol{F}_{\mathrm{R2}}(\mathrm{X}) & \mathrm{R_R} \leq \mathrm{G_0} \\ 0 & \mathrm{R_R} > \mathrm{G_0} \end{cases} \tag{8}$$

As shown in Fig. 1, the total repulsive force $\boldsymbol{F}_{\mathrm{R}}$ is decomposed into two components in the modified model: F_{R1} and F_{R2}. F_{R1} denotes the component of the repulsive force along the line that connects the robot and the obstacle, while F_{R2} represents the component along the line between the robot and the target,

$$\begin{cases} \boldsymbol{F}_{\mathrm{R1}} = (\frac{1}{R_{\mathrm{R}}} - \frac{1}{G_0})\frac{R_{\mathrm{A}}{}^M}{R_{\mathrm{A}}{}^3} \\ \boldsymbol{F}_{R2} = (\frac{1}{R_{\mathrm{R}}} - \frac{1}{G_0})^2 R_{\mathrm{A}}{}^M \end{cases} \tag{9}$$

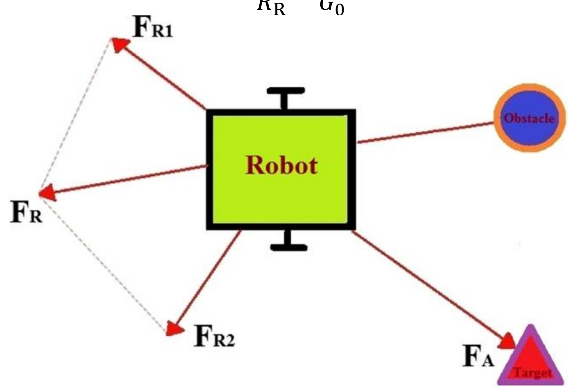

Fig. 1. The analysis of forces applied to a robot

The x-axis and y-axis components of both attractive and repulsive forces can be calculated as follows,

$$\boldsymbol{F}_{\mathrm{Rx}}\,(\mathrm{X}, \mathrm{X_{OB}}) = \boldsymbol{F}_{\mathrm{R}}\,(\mathrm{X}, \mathrm{X_{OB}})\,\cos\,\varphi \tag{10}$$

$$\boldsymbol{F}_{\mathrm{Ry}}\,(\mathrm{X}, \mathrm{X_{OB}}) = \boldsymbol{F}_{\mathrm{R}}\,(\mathrm{X}, \mathrm{X_{OB}})\,\sin\,\varphi \tag{11}$$

$$\boldsymbol{F}_{\mathrm{Ax}}(\mathrm{X}, \mathrm{X_D}) = \boldsymbol{F}_{\mathrm{A}}(\mathrm{X}, \mathrm{X_D})\cos\theta \tag{12}$$

$$\boldsymbol{F}_{\mathrm{Ay}}(\mathrm{X}, \mathrm{X_D}) = \boldsymbol{F}_{\mathrm{A}}(\mathrm{X}, \mathrm{X_D})\sin\theta \tag{13}$$

The forces along the X-axis and Y-axis are determined, and the angle δ between the resultant force and the X-axis is computed, where δ corresponds to the robot's steering angle,

$$\boldsymbol{F}_{\mathrm{x}} = \boldsymbol{F}_{\mathrm{Ax}}(\mathrm{X}, \mathrm{X_D}) + \boldsymbol{F}_{\mathrm{Rx}}(\mathrm{X}, \mathrm{X_{OB}}) \tag{14}$$

$$\boldsymbol{F}_{\mathrm{y}} = \mathrm{F_{Ay}}(\boldsymbol{X}, \mathrm{X_D}) + \boldsymbol{F}_{\mathrm{Ry}}(\mathrm{X}, \mathrm{X_{OB}}) \tag{15}$$

$$\begin{cases} \delta = \arctan\frac{\boldsymbol{F}_{\mathrm{y}}}{\boldsymbol{F}_{\mathrm{x}}}, \boldsymbol{F}_{\mathrm{x}} > 0 \\ \delta = \pi + \arctan\frac{\boldsymbol{F}_{\mathrm{y}}}{\boldsymbol{F}_{\mathrm{x}}}, \boldsymbol{F}_{\mathrm{x}} \leq 0 \end{cases} \tag{16}$$

$$\begin{cases} X_f = x + L \times \cos\delta \\ Y_f = y + L \times \sin\delta \end{cases} \tag{17}$$

The next position of the robot, denoted by X_f and Y_f, is determined based on the step size L. A regulative factor RAM has been introduced in the modified model. This regulative factor serves to adjust the behavior of the system under different conditions. Specifically, if the robot fails to arrive at the target and the parameter M lies within the interval (0, 1), then FR1 becomes increasingly large as the robot approaches the target, effectively allowing only FR2 and FA to guide the convergence route. When M is a positive real number, the robot avoids encountering local minima or target inaccessibility issues [8].

4 Application

4.1 Introduction of an intelligent vehicle

Core Technology used in the vehicle:

1. Camera. The camera assists the vehicle in detecting the surrounding environment and recognizing the obstacles.
2. Lidar. It is used to locate the obstacles precisely, recognize the distance between our vehicle and the obstacles, and avoid them.
3. McNamur wheels. This type of wheel is specialized to allow the vehicle to turn around in tight spaces since the vehicle needs to adapt to all kinds of environments in its work areas.
4. GPS module. This module is deployed to locate the vehicle itself accurately while moving to the destinations.

By using GPS, the robot can receive data to formulate the best route to reach the destination. Through lidar and camera, the robot can prevent hitting objects continuously by utilizing data. Moreover, the robot should drive itself without any control and exactly follow the planned route in expectation.

As shown in Fig. 2, the structure of the vehicle is clearly exhibited. In addition, the product's size is designed for use on campus.

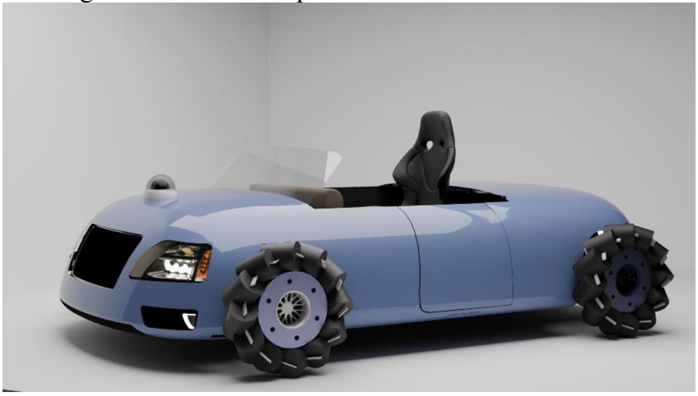

Fig. 2. 3D model of intelligent vehicle

4.2 Simulation of A* algorithm

With the assistance of the code on GitHub, a basic simulation of route planning can be implemented in MATLAB. As shown in Fig. 3, the simulation of the A* algorithm can show how it scans the map and figures out the optimal route to reach the goal.

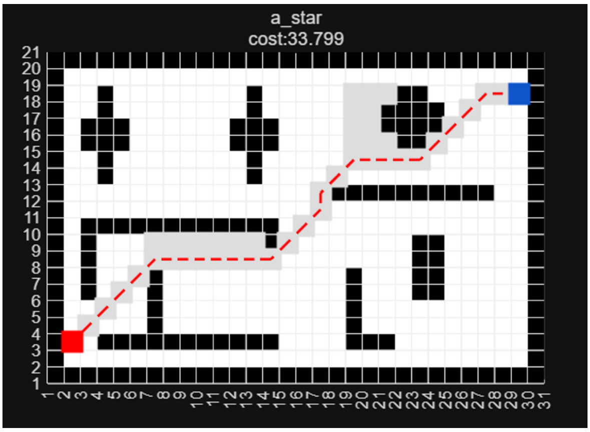

Fig. 3. Simulation of A* algorithm

5 Conclusion

This paper reviews the foundational theories and practical applications of route planning and obstacle avoidance systems in robotics. Through the analysis of classical algorithms such as A* and artificial potential fields, it is demonstrated that while these methods can effectively generate feasible routes, they are constrained by high computational costs and limited adaptability in dynamic environments. Integration of sensors such as LiDAR, cameras, and GPS is identified as a crucial component for achieving accurate environmental perception and real-time navigation. Then, an elementary design of intelligent vehicle and a MATLAB simulation of the A* algorithm illustrate the transition from theoretical models to practical implementation. However, possible challenges—such as performance instability in unstructured environments, limited adaptability across diverse scenarios, and the trade-off between efficiency and completeness—continue to hinder broader deployment.

Overall, this study emphasizes both the potential and current limitations of route planning and obstacle avoidance technologies. Future advancements will rely on developments in artificial intelligence, sensor fusion, and adaptive optimization, which are anticipated to enhance the expansibility and reliability of autonomous navigation systems, including the two core technologies above.

References

1. O. Sharma, N.C. Sahoo, N.B. Puhan, Recent advances in motion and behavior planning techniques for software architecture of autonomous vehicles: A state-of-the-art survey, Engineering Applications of Artificial Intelligence, 101, 104211 (2021)
2. F. Dellaert, S. Hutchinson, Robotics book (JupyterBook, Rowan Cockett, 2023)
3. K. Z. Haigh, J. R. Shewchuk, M. Veloso, Route planning and learning from execution, in Proceedings of the Preprints of the AAAI 1994 Fall Symposium on Planning and Learning: On to Real Applications, New Orleans, LA (1994)
4. F. Duchoň, A. Babinec, M. Kajan, P. Beňo, M. Florek, Tomáš Fico, L. Jurišica, Path Planning with Modified a Star Algorithm for a Mobile Robot, Procedia Engineering, 96, 59-69 (2014)

5. H. A. Gabbar, J. Ren, P. C.K. Hung, Modeling and simulation for connected and automated vehicle (CAV) deployment and performance evaluation, Smart Cities Policies and Financing, 34, 481-510 (2022)
6. M. B. Alatise, G. P. Hancke, A review on challenges of autonomous mobile robot and sensor fusion methods, 8, 39830 – 39846 (2020)
7. F. Bounini, D. Gingras, Hervé Pollart, D. Gruyer, Modified artificial potential field method for online path planning applications, 6, 180-185 (2017)
8. S.M.H. Rostami, A.K. Sangaiah, J. Wang, et al., Obstacle avoidance of mobile robots using modified artificial potential field algorithm, J Wireless Com Network, 70 (2019)