Enhancing 2D Path Search and PID Control for Autonomous Mobile Robots: A* Algorithm Optimization and Hardware Integration

Chengzhi Zhou

College of Electronic Information and Automation, Civil Aviation University of China,

Tianjin, China

Orangecz6@163.com

Abstract. Intelligent autonomous mobile robots find diverse applications across various fields. This article delves into the optimization of path search planning and implementation for these robots in specific scenarios. The study enhances the traditional right-hand principle's search strategy, integrating a forward-looking approach grounded in the goal-oriented law with A* algorithm path planning. Emphasizing the algorithm's lightweight and high efficiency, the study also considers its interaction with hardware systems. The real-world application scenario is abstracted into a two-dimensional planar environment. Collaborative efforts between hardware and software systems result in a comprehensive robot system, validated through tests ensuring swift navigation from the starting point to the endpoint along the planned path.

Keywords: mobile robot, A* algorithm, forward-looking search strategy, targeted search, path planning.

1. Background of the study

Intelligent autonomous mobile robots find versatile applications in diverse environments [1], spanning intelligent manufacturing, search and rescue, and healthcare [2]. Upon receiving a target position, these robots adeptly navigate complex surroundings, autonomously determining the optimal path through artificial intelligence algorithms such as A* and Dijkstra. Concurrently, they regulate motors for stable output, facilitating precise movement towards the target position. This not only conserves human resources but also ensures uninterrupted workflow, significantly enhancing efficiency and reducing the likelihood of errors. The robot's autonomous obstacle avoidance planning effectively mitigates issues of congestion and disorder associated with manual operations, fostering a more organized working process.

Intelligent autonomous mobile robots encounter persistent challenges in achieving comprehensive, efficient, and stable autonomous navigation. These challenges encompass generating effective paths, making appropriate decisions in uncertain and dynamic environments, and adapting swiftly to diverse application scenarios. While prior studies, such as Zhao et al. [3], emphasize the importance of multi-sensory systems in autonomous robot architecture, they lack details on path planning algorithms. Similarly, C. Özbaran et al. [4] focus on mechatronic system design for mobile warehouse robots without introducing algorithms. H. Pan et al.'s [5] improvement of the A* algorithm considers smoothing but overlooks path exploration strategies and the algorithm's lightweight and universal nature. This study addresses these gaps by optimizing the path exploration process, designing intelligent autonomous mobile robots for enhanced adaptability in complex environments, integrating lightweight algorithms with hardware systems, and ensuring practicality and feasibility at a low cost. The key innovations of this paper include:1. Abstraction of 2D environments from multiple application scenarios: Most extant research typically focuses on optimizing the behavior of autonomous robots within a single or specific set of application scenarios. Our study explores the feasibility of creating abstractions from 2D planar environments across diverse application scenarios.

1. Adaptable 2D Environments: Our study innovates by abstracting versatile 2D planar environments, transcending specific scenarios and optimizing computational efficiency.

2. Goal-Oriented Exploration: Introducing a refined forward-looking search strategy, our approach enhances exploration efficiency by consistently progressing toward the target.

3. Robust Perception and Responsiveness: Integrating infrared and gyroscope sensors, our system ensures effective environment monitoring. The STM32F405RGT6 central control unit and DRV8836DSSR dual H-bridge motor drive amplify the robot's perceptual and responsive capacities in complex environments.

The paper unfolds as follows: Section two outlines the abstraction of 2D planar environments across practical applications; Section three delves into the implementation of path search planning and motor control algorithms; Section four elucidates the overall system design framework and hardware circuit architecture; Section five furnishes experimental data, details on experimental environments, and final test outcomes; The concluding section, Section six, summarizes the key findings of the paper.

2. Simulation environment

To address diverse application scenarios, enhance problem-solving efficiency, and streamline research, we adopt scene abstraction. This method involves creating an intuitive and controllable grid model, similar to graphical pixels [7], customized for practical application contexts. By simplifying environmental representation and computational processing, we abstract the two-dimensional planar environment into a 16*16 grid, illustrated in Fig. 1. Each cell simulates the obstacle situation, visually indicating passable areas in green and obstacle areas in black. This strategic reduction of complex scenarios to a 2D plane effectively balances functionality and computational complexity [6].

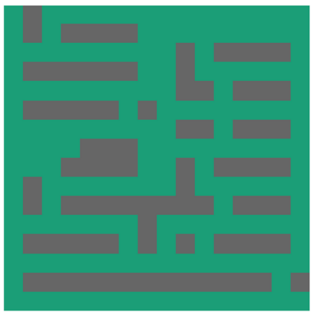

Fig.1 Two-dimensional planar environment

3. Algorithmic control

To attain efficient navigation and precise control for intelligent autonomous mobile robots, this research addresses both path search and planning, along with hardware control challenges. It incorporates forward-looking search to enhance traversal and exploration strategies with a more robust focus on goal-pointing. Recognizing the algorithm's lightweight nature, the A* algorithm is selected for path planning. The hardware control aspect employs the incremental PID algorithm for implementation.

3.1 Path search planning

During the path search planning process, we deploy an initial forward search strategy to traverse and explore, effectively transforming a physical map into a digital domain stored in the central control chip, containing the obstacle landscape of each location. Once the map is established, we employ the A* algorithm for effective path planning.

As a dynamic optimization technique, the forward-looking search strategy fundamentally involves simulating potential future states before executing an operation. In this study, we apply this technique to enhance the robot's traversal search strategy by proposing optimizations for commonly adopted

rules such as centripetal and right-handed laws. Similar modifications are suggested for the goalward law. The optimization process begins with designing the state and action spaces, defining the available actions at decision points, and the resulting state space these actions generate. These moves include advancing forward, executing left and right turns, with the state comprising the robot's specific coordinates on the two-dimensional map, forming the fundamental framework of the optimized search strategy. The key to optimizing the strategy lies in establishing an evaluation function, offering a quality metric for comparing different actions. In this research, enhancements are made to the left-hand and right-hand rules, with the distance from the target position selected as the primary evaluation criterion. This guides the movement towards the direction closer to the intended target.

$$h(n) = \sqrt{(\mathrm{x} - x_n)^2 \quad (\mathrm{y} - y_n)^2} \tag{1}$$

$$h(n) = |\mathrm{x} - x_n| \quad |\mathrm{y} - y_n| \tag{2}$$

We employ a combination of Manhattan and Eulerian distances in our approach. The Manhattan distance, calculated as the sum of the absolute values of the horizontal and vertical coordinates, generally yields paths with fewer bends but may exhibit localized detours when encountering obstacles. In contrast, the Eulerian distance, representing a diagonal distance, tends to avoid localized detours but introduces more twists [8]. Consequently, the Eulerian distance is applied when an obstacle is encountered at the decision point, as shown in (1). Conversely, the Manhattan distance is utilized when no obstacle is present at the decision point, as indicated in (2). The decision-making process is illustrated in Fig. 2.

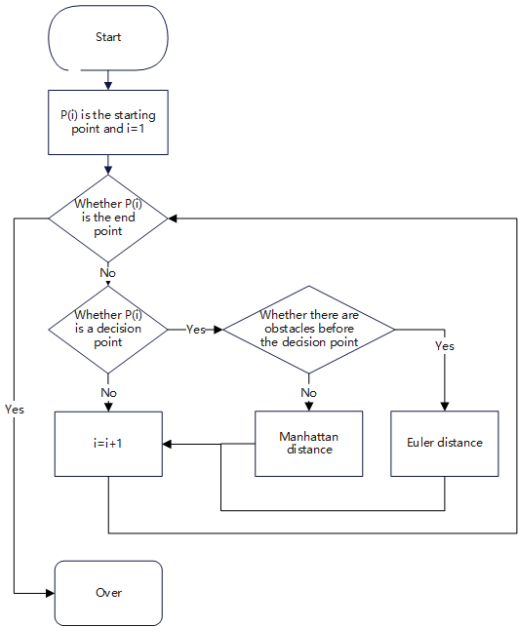

Fig.2 Decision Point Decision Flowchart

After the evaluation, the final action taken is based on the evaluation results, and the action that brings the robot closest to the goal is selected. The optimization strategy adheres to the guiding principle of progression towards the target, thereby enhancing the efficiency of the robot's exploration in intricate environments and managing to confine its operation within the stipulated computational complexity.

Path planning is carried out once the path exploration using the A* algorithm is completed. The A* algorithm, a heuristic search algorithm, amalgamates the completeness of Breadth-first Search and the efficiency benefits of Dijkstra's algorithm [9], thus optimizing the robot's path-planning process [10]. The next optimal node for expansion is selected by comparing the costs of nodes, iteratively continuing this process until reaching the final goal point.

The A* algorithm calculates the path cost mainly based on (1) and (3)

$$f(n)=g(n)+h(n). \quad (3)$$

Where (x_n, y_n) is the current node, (x, y) is the target node, f(n) is the total cost of the current node as shown in (3); g(n) is the actual cost from the starting point to the node; The estimated cost from the current node to the endpoint, denoted by h(n), is derived via a heuristic function that provides a cost estimation from node n to the target point, as depicted in (1) [11]. Fig. 3 illustrates the flow of the A* algorithm:

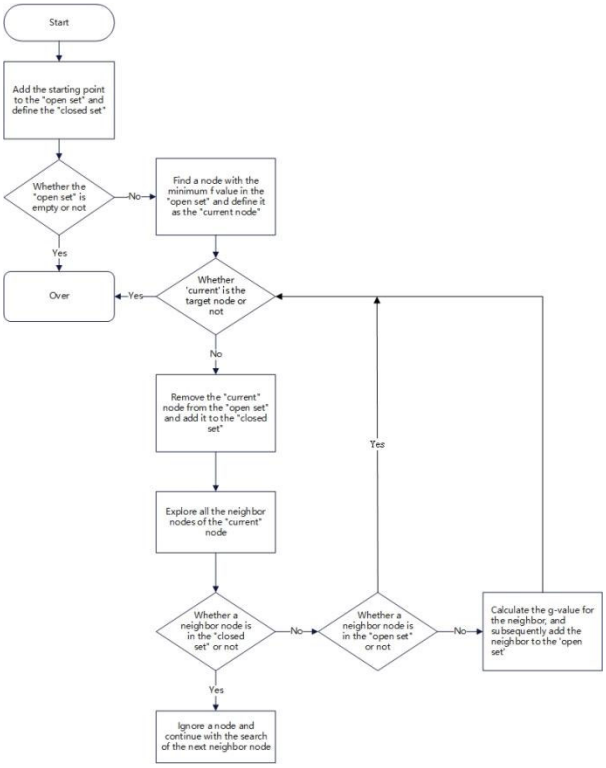

Fig.3 A* Algorithm flowchart

The forward-looking search strategy based on the combination of the toward-goal law and A* algorithmic path planning improves the traditional path search planning scheme by integrating the lightness and realizability of the algorithms. It achieves a balance between search efficiency and path quality.

3.2 Motor control algorithm

The motor control algorithm utilizes an incremental PID control algorithm [12]. The incremental algorithm requires historical deviations, during the kth control cycle, the deviations input during the k-1th and k-2th control operations are used [13]. The calculation yields Δu_k, as shown in (4), and the accumulation of these yields provides the PID output, as demonstrated in (5).

$$\Delta u_k = \quad \text{Kp} \quad \text{e(k-1)} + \text{Ki} * \quad \text{e(k)} + \text{Kd} * \quad \text{e(k)} - 2\text{e(k-1)} + \text{e(k-2)}). \quad (4)$$

$$u_k \quad u_{k-1} \quad \Delta u_k \quad (5)$$

e(k) is the error value between the current target and the actual situation, e(k-1) is the error value between the previous target and the actual situation, and e(k-2) is the error value between the antepenultimate target and the actual situation. Kp is the proportional coefficient, which immediately

responds to deviations in the proportional control part and determines the controller's sensitivity to errors. The larger the value of Kp, the faster the response speed, but there will inevitably be a steady-state error in proportional control. Ki is the integral coefficient, which adjusts the integral part of the control to reduce the system's steady-state error. However, it may result in an excessive degree of system overshoot. Kd is the derivative coefficient responsible for the derivative part of the control, efficiently managing and suppressing the system's overshoot [14].

The speed of the DC motor is set to r. At this time, the deviation of the feedback speed and the set speed is e(k), the system saves the deviation e(k-1) and the deviation e(k-2) of the last time, and these three inputs are calculated by incremental PID to get u_k. The system also saves the last PID output, u_{k-1}, so u_{k-1} plus the incremental u_k is the PID output u_k of the current control cycle, which outputs the voltage control output PWM wave to drive the DC motor, the encoder to continue to detect the running distance and speed and feedback control flow as in Fig. 4.

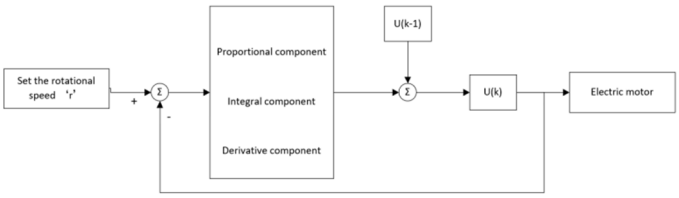

Fig.4 Flowchart of PID algorithm

4. Hardware circuit design

Fig. 5 displays the overall system design framework. The robot is powered by a 7.4V model airplane battery. Concurrently, a voltage conversion circuit, composed of boost and buck chips, arranges power supply to various modules, including the motor and infrared detectors[15].

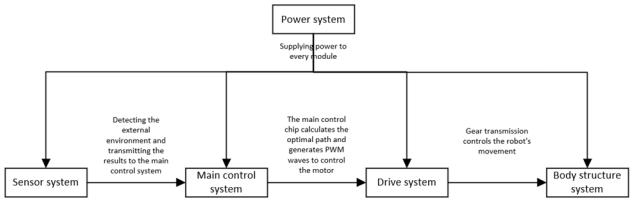

Fig.5 Hardware system design

The robot board was designed using AltiumDesigner and LCEDA Pro for PCB layout and wiring, as shown in Fig. 6.

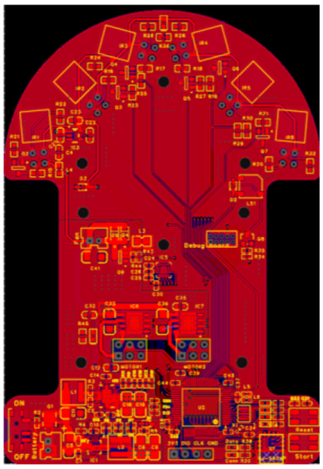

Fig.6 PCB design

4.1 Microcontrollers

The STM32F405RGT6 is used as the main control chip, which is a 32-bit processor based on ARM Cortex-M4 with a high operating frequency of 168Mhz, 512kb of flash memory, and 192kb of RAM. Coupled with a 12M crystal clock source, the STM32F405RGT6 acts as the primary component of the central control system, orchestrating various subsystem functions and implementing various algorithms and protocols [16]. The compact size of the chip and the ample availability of pins are sufficient to accommodate the requirements of the microcontroller portion of the circuit design, as illustrated in Fig. 7.

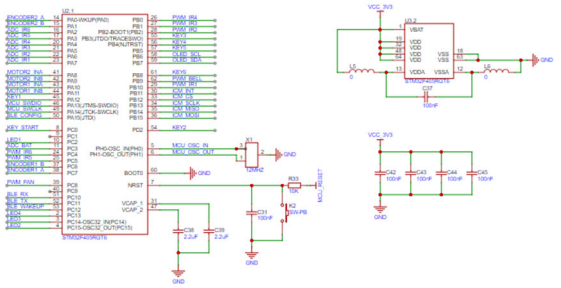

Fig.6 Microcontroller circuit

4.2 Sensor systems

The sensor system utilizes a paired set of infrared tubes and gyroscope sensors to monitor both the operational state of the vehicle and surrounding environmental conditions. The SFH4550 infrared emitter tube emits at peak strength at a wavelength of 860nm, while the central wavelength for the TPS601A infrared receiver tube is 800nm. Together, these constitute a pair of transceiver tubes. The infrared emitter releases infrared rays that, upon encountering obstructions such as maze walls, reflect back and are received by the receiver tube, where they are converted into electrical signals. The received infrared light is converted into electrical signals of varying strength, following the principle that closer objects produce stronger signals, while farther ones yield weaker signals. These signals are then conveyed to the microprocessor. The ICM-20602 sensor collects a stream of data, including the vehicle's current acceleration and angular velocity, which is then relayed to the central control system for processing.

4.3 Motor drivers

The DRV8836DSSR chip, equipped with a built-in dual H-bridge driver circuit, enables the independent operation of two DC motors, thereby optimizing integration and minimizing footprint. The microcontroller accomplishes the motor's forward and reverse rotations by controlling the state of the four MOSFETs in the H-bridge circuit. The microcontroller also modulates motor speed via a PWM wave by adjusting the duty cycle of the voltage waveform. The H-bridges drive DC currents, delivering an output current up to 1.5A and mitigating the system's total power consumption by providing a maximum current of just 95nA in low-power mode. The chip's preset PWM pulse input interfaces are directly compatible with the microprocessor output [17]. Fig. 8 presents the motor drive section.

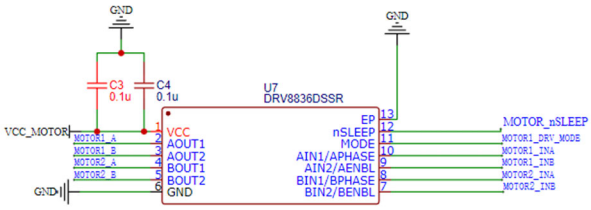

Fig.7 Motor drive circuit

5. Experimental result

In the simplified 2D planar environment, the simulation test for path planning, utilizing the aforementioned algorithmic process, with the origin at (0,0) and destination at (15,15), generates the illustrated path depicted in Fig. 9.

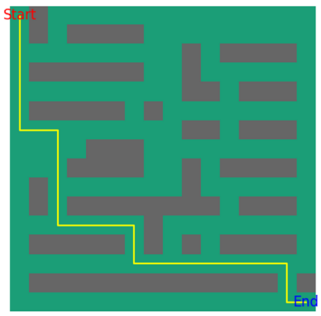

Fig.8 Algorithm simulation result

For the sensor system test, a collision on the vehicle body is detected, and the corresponding distance is set to zero when the detection distance falls below 2.5cm. Table 1 presents the actual measurements from the infrared sensor at four critical positions. The outcomes of the gyroscope tests are articulated in Table 2.

Table 1 Infrared test data

	Blocking voltage /mv	Conversion distance /m
Left infrared	3810	0.37
Right side infrared	3862	0.36
Left front infrared	3812	0.36
Right front infrared	3880	0.37

Table 2 Gyroscope test data

angle of rotation	Gyroscope deflection angle	inaccuracies
45°	44°	2.2%
90°	87°	3.3%

Fig. 10 illustrates the testing procedure executed on the fully assembled robot. The testing apparatus comprises the robot, a simulation map, and a timing device. The dimensions of the robot are 100mm by 80mm. The simulation environment consists of a versatile 1440mm by 1440mm lattice model constructed using baffles, which simulate a naturally varying working environment. A trio of tests were conducted, the results of which are compiled in Table 3.

Fig.9 Robot

Table 3 Measured data of the robot

No. of tests	Planning time (S)	Arrival time (S)	Number of touches	Human intervention or not
The first time	40	10	0	NO
The second time	43	12	0	NO
The third time	39	9	0	NO
average	40.6	10.3	0	NO

6. Conclusion

This paper initiates by delineating the application scenarios of intelligent autonomous mobile robots and abstracting a 2D planar environment for examination across various practical contexts. Prior to real-world experimentation, we validate the scheme's feasibility through software simulation. Subsequently, we expound on the core algorithm enhancing the path exploration process, guiding the rule of advancing toward the target with a predictive search strategy, and detailing the overall system layout and hardware circuitry. The experimental segment distills the simulated environment for evaluating the robot in practical implementation scenarios, such as intelligent manufacturing and healthcare services. This facilitates efficient path search planning and the autonomous motion of intelligent robots, effectively enhancing work productivity within these contexts.

References

[1] C. Alias, I. Nikolaev, E. G. Correa Magallanes and B. Noche, "An Overview of Warehousing Applications based on Cable Robot Technology in Logistics," 2018 IEEE International Conference on Service Operations and Logistics, and Informatics (SOLI), Singapore, 2018, pp. 232-239.

[2] Teng WANG, Jing PAN, Lu DONG, et al. Key technologies and applications of intelligent guiding robots for epidemic prevention[J]. Chinese Journal of Intelligent Science and Technology, 2021, 3(2): 187-194.

[3] T. Zhao, "Architecture and Integrated System of Autonomous Intelligent Robot Based on Multi-perception," 2023 Asia-Europe Conference on Electronics, Data Processing and Informatics (ACEDPI), Prague, Czech Republic, 2023, pp. 78-81.

[4] C. Özbaran, S. Dilibal and G. Sungur, "Mechatronic System Design of A Smart Mobile Warehouse Robot for Automated Storage/Retrieval Systems," 2020 Innovations in Intelligent Systems and Applications Conference (ASYU), Istanbul, Turkey, 2020, pp. 1-6.

[5] H. Pan, C. Guo and Z. Wang, "Research for path planning based on improved astart algorithm," 2017 4th International Conference on Information, Cybernetics and Computational Social Systems (ICCSS), Dalian, China, 2017, pp. 225-230.

[6] J. Yuan, Y. Li, H. Pan, Z. Cui and Y. Liu, "3D Traffic Scenes Construction and Simulation based on Scene Stages," 2018 Chinese Automation Congress (CAC), Xi'an, China, 2018, pp. 1334-1339.

[7] Z. Wang and X. Xiang, "Improved Astar Algorithm for Path Planning of Marine Robot," 2018 37th Chinese Control Conference (CCC), Wuhan, China, 2018, pp. 5410-5414.

[8] Geng Hongfei, Shen Jianjie. Improvement and Verification of A~* Algorithm for AGV path Planning [J]. Journal of Computer Applications and Software,2022,39(01):282-286.

[9] Dijkstra, E.W. A note on two problems in connexion with graphs. Numer. Math. 1, 269–271 (1959).

[10] H. Wang, J. Zhou, G. Zheng and Y. Liang, "HAS: Hierarchical A-Star Algorithm for Big Map Navigation in Special Areas," 2014 5th International Conference on Digital Home, Guangzhou, China, 2014, pp. 222-225.

[11] P. E. Hart, N. J. Nilsson and B. Raphael, "A Formal Basis for the Heuristic Determination of Minimum Cost Paths," in IEEE Transactions on Systems Science and Cybernetics, vol. 4, no. 2, pp. 100-107, July 1968.

[12] C. Qi, S. MingJian and W. LinGao, "The constant current control of anti-scaling descaling device based on incremental digital PID controller," The 27th Chinese Control and Decision Conference (2015 CCDC), Qingdao, China, 2015, pp. 4948-4952.

[13] X. Guo, Z. Li and G. Sun, "The Robot Arm Control Based on RBF with Incremental PID and Sliding Mode Robustness," 2019 WRC Symposium on Advanced Robotics and Automation (WRC SARA), Beijing, China, 2019, pp. 97-102.

[14] T. -Y. Wang and C. -D. Chang, "Hybrid Fuzzy PID Controller Design for a Mobile Robot," 2018 IEEE International Conference on Applied System Invention (ICASI), Chiba, Japan, 2018, pp. 650-653.

[15] Z. Gu, M. Wang and H. Peng, "Design of Integrated Robotic System for Object Recognition and Intelligent Grasping based on STM32," 2022 IEEE 4th International Conference on Power, Intelligent Computing and Systems (ICPICS), Shenyang, China, 2022, pp. 46-50.

[16] C. Wang, Q. Gao and J. Tang, "Research on Wheeled Inspection Robot Drive System," 2019 IEEE 4th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), Chengdu, China, 2019, pp. 1648-1652.

[17] C. Zhao and Z. Hua, "Design of Motor Speed Control System Based on STM32 Microcontroller," 2022 International Conference on Computation, Big-Data and Engineering (ICCBE), Yunlin, Taiwan, 2022, pp. 274-276.