Research Article

Robot Trajectory Planning Based on the Energy Management Strategy

Mingkai Li

Guilin University of Electronic Technology, Guilin 541000, China

Correspondence should be addressed to Mingkai Li; 1900101010@mails.guet.edu.cn

Received 18 July 2022; Accepted 20 August 2022; Published 10 September 2022

Abstract

With the increasing demand for automated production technology in national defense, industry, agriculture, and other fields, the status and role of mobile robots are becoming more and more pivotal. The intelligent technology of robots is becoming more and more demanding in terms of reliability, stability, efficiency, and adaptability, and the autonomous navigation technology of mobile robots is facing new challenges. This paper introduces the basic principle of traditional A∗ algorithm, points out the problems of this algorithm such as cumbersome calculation, large turning angle, and unsmooth derived trajectory planning, and proposes an improved A∗ algorithm. The improved algorithm introduces a two-way alternating search mechanism to improve the efficiency of the search path, and the improvement of the heuristic function solves the problem that the two-way alternating search A∗ algorithm takes more time when there are obstacles perpendicular to the path on the way to the encounter. Simulation experiments in different raster environments prove that compared with the traditional A∗ algorithm, genetic algorithm, and simulated annealing algorithm, the improved A∗ algorithm can significantly improve the search path speed, while overcoming the disadvantages of many path turning angles and large turning angles and is an efficient and feasible algorithm for raster map environments.

1. Introduction

In recent years, only industries have flourished, and the robotics industry has received increasing attention, and its related technologies have received unprecedented attention [1]. The huge economic benefits embedded in the mobile robot industry have led countries around the world to carry out their own blueprints for the development of mobile robots; for example, in 2018, the United States issued the National Artificial Intelligence Research and Development Strategy, proposing that the United States will continue to increase scientific research investment in mobile robots in the future [2]; Japan also explored more extensively in the field of artificial intelligence, and then released the “Artificial Intelligence in Core Technology requirements,” indicating the important position of intelligent robots in Japan’s national strategy of artificial intelligence [3]; in order to accelerate the construction of a strong science and technology country, China also clearly pointed out in “China Smart 2025” to vigorously develop the intelligent robotics industry, putting forward new requirements to accelerate the promotion of mobile robotics in China [4].

In recent years, with the support of intelligent technologies such as Internet technology, computer technology, and hardware technology, China’s intelligent mobile robot industry has grown very rapidly, and since 2013, the development of China’s robot industry has entered an explosive period, and in 2016, China’s mobile robot industry has accounted for more than 30% of the global robot market, occupying the largest mobile robot market in the world. At present, the development of China’s mobile robotics industry has surpassed that of many developed countries [5], but the demand gap for new technologies in China’s mobile robotics industry is still very huge, so the core technology of mobile robots in China is still in urgent need of new breakthroughs.

Robotics involves multiple disciplinary technologies such as robot kinematics, artificial intelligence, and mechanics [6]; therefore, enhancing mobile robotics-related
present and future research in the field of mobile robots. Autonomous
navigation technology for mobile robots is the use of sensor technology and various advanced algorithms in a certain environment to complete robot localization and environment map building and to help mobile robots move safely and autonomously toward a target point.

Robot action and navigation is the core technology to be continuously developed, which consists of three main parts: localization, environment map building, and path planning, among which, trajectory planning is particularly important, and localization and environment map are the basis of path planning [7]. Trajectory planning is fundamentally to have a reliable action route and way for the robot to act autonomously, i.e., to provide the robot with a safe path from the starting position to the target position, which can avoid obstacles and try to ensure the optimality of the path in a certain environment for a certain index [8]. So far, path planning technology has been developed for more than fifty years, but for different planning requirements and some special scenarios of applications, there is still much room for improvement in the overall path planning in terms of comprehensiveness [9], and path planning algorithms are still in need of new breakthroughs in terms of advancedness, applicability, and efficiency. Today’s global robots are becoming more and more important with increasing sales, as shown in Figure 1.

At present, mobile robots can be seen in various fields instead of manual work, and the position of mobile robots in social production and services is becoming more and more important, while autonomous navigation, as the core requirement and basic embodiment of intelligent mobile robots, will naturally have higher and higher technical requirements, and path planning technology, as the core technical composition of autonomous navigation, has an important research value for the field of mobile robots. It goes without saying that the challenge of path planning technology innovation is also very great, as shown in Figure 2.

2. Research Background

The first robot that could "move," Shakey, was successfully introduced in 1972 under the leadership of Stanford professor Charlie Rosen [10]. The success of Shakey opened the beginning of research in the field of mobile robots and started the boom of research on autonomous navigation of mobile robots, which is a very important inspiration for the present and future research in the field of mobile robots.

With the new breakthroughs in sensor technology, artificial intelligence technology, and information technology in the 80s and 90s, the development of intelligent mobile robots was greatly accelerated, and a large number of scholars and funds were invested in the research and development of mobile robots, which made the development of mobile robots step into the explosive growth stage, and mobile robots also began to enter the industrialization stage, and various different kinds of mobile robots, a variety of different kinds of mobile robots, have sprung up into the public’s view, which can be classified according to the different functional roles, application environments, movement modes, and working conditions, as shown in Figure 3.

So far, mobile robots have been developed for more than 50 years and account for a very large market share in the global intelligent energy industry. The rapid development of mobile robots needs to rely on various hardware technologies and software technologies, and the rapid development of sensor devices and software technologies has, on the one hand, promoted the development of mobile robotics and, on the other hand, caused the problems of low reuse of R&D codes and incompatibility of new sensors due to the increasing complexity and diversity of mobile robot software platforms and hardware devices [11]; therefore, the development of mobile robots encountered considerable difficulties in terms of systematization, modularity, and iterability.

To solve this problem, the Robot Operating System (ROS), a mobile robot development system was born, which originated from the Switchyard AI project at Stanford University and is now taken over by Willow Garage [12]. The ROS system is characterized by open source, interfaces in multiple languages, rich functional components to accomplish communication between programs, upper-layer software and underlying hardware, and various visualization tools to facilitate development [13].

At present, the ROS system has experienced more than a decade of development, and it not only has a very superior mobile robot development ecological environment, which improves the code reuse rate of mobile robot development and reduces the development cycle but also provides a very active communication community, which creates a benign communication platform for the development of machine mobile robots and accelerates the technology sharing in the field of mobile robots, and now, the ROS system is the mainstream global mobile robot system development platform, and most of the robot development systems are ROS systems [14]. In recent years, in many enterprises and universities in China, the basic systems for mobile robot development are also ROS systems, and at present, China is the country with the most number of users of ROS systems in the global mobile robot development field, as shown in Figure 4.

At present, mobile robot functions have been more perfect, and product types have become more and more abundant, such as the service mobile robot PR2, which was developed by Stanford Research Institute and designed and manufactured by Willow Garage in 2010 [15]; several mobile robots applied to service, education, industry, and other fields were independently developed by Harbin University Robotics Group [16]; the first mobile robot was developed by Boston. The first bionic quadruped mobile robot that can be applied to complex scenarios, Big Dog, was developed by Boston Dynamics in 2005 [17]. At present, China’s bionic quadruped mobile robot research has also made great breakthroughs, and the Cyber Dog [18], independently developed by Xiaomi in 2021, is not only functional but also priced to be accessible in ordinary families.
Self-driving cars belong to the category of wheeled mobile robots, and self-driving technology is one of the core competencies of the future automobile market [19]. In the 1940s, foreign countries took the lead in launching research in the field of self-driving cars, and in the field of autonomous driving, China has achieved independent research and development of self-driving cars, equipped with unmanned vehicles on Baidu’s self-developed Apollo autonomous driving platform, and currently, unmanned vehicles developed based on this platform can reach Level 4 level of autonomous driving, which is at the world leading level [20].

In summary, from the current types, functions, application scenarios of mobile robots, and the development status of various types of mobile robots at home and abroad, the future development trend of mobile robots must be more and more inclined to be highly automated and intelligent, and the characteristics of bionic will also become more and more significant. As the functions of mobile robots become more and more perfect and the types of mobile robots increase, the applications of mobile robots in various complex or dangerous environments and in daily life will become more and more common.
3. Materials and Methods

3.1. Basic Theory

3.1.1. Simultaneous Robot Localization and Map Building Techniques. Simultaneous localization and mapping (also known as SLAM) is a collective term for localization and map building, a concept first proposed by Durrant-Whyte et al. In the field of autonomous navigation for mobile robots, the SLAM problem has been graphically described as “where am I” and “where am I going.” SLAM technology relies on various sensor technologies to obtain environmental information, and simultaneous localization and map building technologies are divided into laser simultaneous localization and map building and vision simultaneous localization and map building, of which
LIDAR is used for laser simultaneous localization and map building, while the latter is camera-based, as shown in Figure 5.

Robot localization is one of the purposes of SLAM technology, which refers to the use of motion sensors to obtain robot motion information and estimate the robot’s position and pose information from the robot’s overall and local motion information. At present, the positioning technology of mobile robots has reached a relatively advanced level, which can meet the positioning requirements of mobile robots in various complex environments. In addition, the positioning methods include relative positioning and absolute positioning, among which, relative positioning, i.e., positional tracking, often uses sensor information such as encoders, gyroscopes, and inertial sensors (IMU) to project the displacement and attitude information of mobile computing robots and is currently a common positioning method for mobile robots.

Map construction is also the purpose of SLAM technology, which refers to the use of sensors to obtain environmental information and build a map that precisely matches the actual environment, with the main purpose of providing a sufficiently rich and reliable judgment basis for mobile robots’ motion decisions. SLAM technology was first presented at the International Conference on Robotics and Automation (IEEE Robotics and Automation Conference) in the 1980s. Smith developed the Kalman filter method to address the problem of map building and achieved groundbreaking research results. The particle filtering method is a filtering method with nonlinear sequential importance sampling. At present, based on the two filtering algorithms, combining various optimization and improvement methods is the mainstream trend of research in the field of SLAM algorithm.

The key problems of SLAM technology mainly include two major categories, one is the environmental information consistency and the other is the map building drift problem, among which, the environmental information consistency problem is also known as the data correlation problem, that is, the determinacy of the sensor to acquire the sensory information of an object in the environment, which is simply the consistency of the environmental information acquired by the mobile robot when it is in different positions. This problem has a very important impact on the accuracy of mobile robot positioning. Some intelligent algorithms such as the neural network method and fuzzy algorithm are also used to solve this problem.

The map building drift problem refers to the problem that the environmental information recognized by the sensor in the previous frame is in error with the environmental information recognized in the next frame during the map building process, resulting in the inability to align the scale of the two environmental map frames. The common method to solve this problem is loopback testing, and currently, loopback testing methods include violent matching loopback testing method. SLAM technology for mobile robots needs to combine technologies from several disciplines, and future research trends in SLAM technology need to focus on multisensor fusion, recognition of dynamic environments, and multirobot cooperation.

3.1.2. Energy Management. The goal of energy management is to rationally distribute power between the engine and the motor to improve fuel economy while optimizing emission performance. In today’s robotic applications, the power to drive the robot is provided by the engine and the battery together or separately, using energy management strategies to keep the engine working in the high efficiency region. Since the motor can act as a generator, it can also act as an electric motor. For sustained hybrid robots, it is also necessary to ensure the maintenance of the battery charge (state of charge, SOC).

The energy management strategy distributes the torque between the engine and the motor in real time while satisfying the soft and hard constraints of the system. The rule-based energy management strategy ensures the operational efficiency of the system by setting threshold value control rules for the operating area of each component. These control variables used as threshold values usually include vehicle speed, motor torque, and battery SOC. These threshold values are usually determined after tuning the parameters. In addition, they can be combined with fuzzy logic control to improve the robustness of the algorithm.

3.2. Research Method for Mobile Robot Path Planning Techniques. In the process of autonomous mobility of mobile robots, path planning plays an important role in decision control, and in the field of mobile robot research, the problem has also been formulated as “how do I get...
there.” When a robot plans a path, it is important to ensure that the path is as safe as possible, efficient in planning time, and smooth. After the emergence of graph-based search algorithms, a variety of intelligent planning algorithms have emerged. Although significant research results have been achieved in the current field of path planning technology, there are still many problems and shortcomings in the comprehensive performance of planning algorithms in practical applications, and therefore, improvement and optimization of path planning algorithms for path planning is a popular current research trend. According to the different types of maps used, path planning can be divided into discrete-domain path planning and continuous-domain route planning, and currently, the latter is the mainstream division of path planning algorithms.

Global path planning is mainly applied in relatively stable static environment; therefore, the requirement for real-time map is low and the requirement for map perverseness is high. The variety of global path planning algorithms is relatively rich, and they are divided into different search methods.

3.2.1. Algorithms Based on Graph Search. The graph search algorithm is to search an optimal global path from the starting point to the end point in the completed environment skeleton map (such as raster map, viewable map, and topological map) according to the different map information. The commonly used graph search algorithms include BFS, DFS, Dijkstra, and A*.

For the limitations of the algorithms, scholars at home and abroad had proposed many improvement methods, the A* algorithm has a certain purpose, and the planning efficiency of the algorithm is naturally higher. To further improve the performance of all aspects of the A* algorithm, LINM et al. introduced parent node information in the A* algorithm to reduce the number of extended nodes and shorten the planning time; Hua Hong et al. proposed a multiple A* algorithm, designed a new node management criterion, improved the search efficiency, performed path smoothing, and obtained paths that conform to the robot motion characteristics.

Such algorithms, first, perform cluster analysis of the data by certain algorithms, so as to obtain the central part of the hidden neural network, and then use the results of this step to perform calculations to figure out the width value of the number. The specific width values are calculated as follows (1)–(3):

$$\sigma_j = \frac{c_{xy}}{\sqrt{2h}}$$  \hspace{1cm} (1)

In the $c_{xy}$ formula, start calculating the maximum distance to the centroid, and $h$ the specific is the number of nodes.

After the input data are analyzed in the implied layer with the output layer for the relevant data, the output $x_i$ of the first node $j$ of the input sample in the implied layer is calculated by the following equations (2)–(3):

$$\phi(x_i, j) = \exp\left(-\frac{1}{2\sigma_j}x_i - c_j \right).$$  \hspace{1cm} (2)

In the formula, $c_j$ is the centroid of the node in the first layer, and the $\sigma_j$ is the width value of the node in the first layer.

The output of $x_i$ the first node of $j$ the input sample in the output layer is calculated by the following equation (3):

$$y_m = \phi(x_i, j) \times w_m.$$  \hspace{1cm} (3)

In the $w_m$ formula, $\phi$ is the function of the involved weights.

3.2.2. Sampling-Based Planning Algorithm. The sampling-based planning algorithm is to find the end point in the built environment skeleton map by probabilistically sampling a certain indicator (such as roadmap information and spatial pose) in order to continuously reduce the search space, which can be divided into comprehensive query method and single query method according to the order of building roadmap, and the PRM algorithm and RRT algorithm are typical sampling planning algorithms.

Probabilistic road map (PRM) is the first sampling-based planning algorithm, which was proposed by Lydia Kavraki and Jean-Claude Latombe in 1994, and then foreign scholars SWilmarth and NAMato proposed derivative algorithms for the drawbacks of the algorithm, respectively, MAPRM and OBPRM, and domestic scholars Hongqing Tian and Jianqiang Wang also integrated the artificial potential field method based on PRM, i.e., APFPRM fusion algorithm, which improved the planning efficiency of the algorithm and the overall safety of the path.

The decision speed let random tree algorithm (RRT) is based on the incremental sampling planning algorithm proposed by American scholars SMLaValle, and in 1998, the algorithm is mostly applied in the environment with many obstacles, with the increasing application requirements of the algorithm, many improvements of the RRT algorithm have been proposed, such as the RRT proposed by Kuffner and the LaValle connect algorithm, which improves the planning efficiency of the algorithm by speeding up the grow tree expansion; Weimin Zhang improved the RRT algorithm by taking the target about and target bias as the starting point, which solves the problem of low planning efficiency of the traditional RRT algorithm in narrow maps.

The local path planning algorithm appeared relatively late, and the algorithm is mainly applied in variable dynamic environments, so it is necessary to build dynamic maps with high real-time performance; that is, it relies on higher sensor technology to identify dynamic obstacles in the environment and ensure the safety of mobile robots in dynamic environments. For example, the artificial potential field (also known as APF) method was proposed by Khatib in 1985. For example, foreign scholar Lazarowska proposed the DAPF algorithm, which constructs a discrete potential field based on the traditional APF to improve the planning efficiency and shorten the path length; domestic scholar Luo Qiang
et al. introduced the robot-target distance factor and tangent method in the repulsion function of the traditional APF algorithm, which solved the problem that the traditional APF algorithm cannot reach the end point.

Compared with the APF algorithm, the dynamic window approach (DWA) appeared much later, and naturally, the algorithm is more advanced. The DWA algorithm was born in 1997 and proposed by foreign scholars Dieter Fox and Sebastian Thrun. It is a local planning algorithm based on the sampling of motion variables, so the algorithm also belongs to the sampling planning algorithm, and the path planned by the DWA algorithm is more in line with the actual motion characteristics of mobile robots. The DWA is the most widely used algorithm in the field of local path planning for mobile robots.

To summarize the above, although there are many kinds of path planning algorithms, each algorithm has certain limitations, and so far, there is still no planning algorithm in the absolute leading position, so there is still some room for improving and improving the path planning technology of mobile robots. The future development of path planning algorithms should focus on the improvement of traditional algorithms, the integration of multiple algorithms, and the application of complex and multidimensional environments.

4. Results and Discussion

4.1. Improved Global Path Planning. The optimal path between the starting and ending points must be identified without running into any obstacles, given a mobile robot and an environment model. Establishing the environment model and the path planning algorithm are the two main components of global path planning, and the choice of algorithm has a significant influence on the effectiveness of the path planning process. The conventional A* algorithm has been explored and is advantageous due to its straightforward premise, high path planning efficiency, and high success rate when looking for the best path. Although numerous researchers have made some progress in the A* algorithm’s performance, they typically only enhance one aspect of the algorithm’s path planning effectiveness, path length, or path turning angle. In some instances, they also fail to fully take into account the robot’s safety in relation to the obstacle distance. In order to increase the robot’s visualization, path discrimination accuracy, overall efficiency, minimized redundant paths provide smoother global paths; thus, this work offers an updated A* algorithm. The modified A* algorithm is more effective than the A* method, the genetic algorithm (GA), and the simulated annealing process according to simulation data (SA).

4.1.1. Principle of A* Algorithm. The traditional A* algorithm searches for the target location by heuristics and can find the shortest path by using the edge cost (edgeweight) and Euclidean distance-based heuristics. Usually, the traditional A* algorithm searches for four neighboring nodes in each expansion. However, fixing and limiting each steering angle to 90° affects the search efficiency and increases the path length and turning angle. In order to increase the effectiveness and quality of the path design process, we use an 8-connected technique in this study. For each expansion, the number of neighboring nodes has risen from 4 to 8, and the steering angle is either 45° or 90°.

The classic A* algorithm’s flow diagram is more complex, and when it searches for paths using this technique, it simply looks for the best path nodes, which significantly reduces processing. The commonly used heuristic functions h include Euclidean distance function, Manhattan distance function, and Chebyshev distance function. The path calculated using the Euclidean distance function as the heuristic function in this study is closer to the actual path since the A* method searches eight nearby nodes in each expansion.

4.1.2. Raster Method Environment Modeling. Since the traditional A* algorithm is a raster traversal search algorithm, it treats the robot’s size as if it were a prime point when determining its actual path. This causes the robot to take more redundant paths, which increases its overall energy consumption and makes it easier to run into obstacles. The enhanced methodology described in this study uses an optimization strategy to overcome this issue by creating an enlarged obstacle layer surrounding the obstacles, and the number of layers of which may be changed depending on the size of the robot. The number of growing obstacle layers is assumed to be 1 as the mobile robot is modeled as a square grid with a side length of 1. This optimization technique maintains a set distance between the path and the barriers to save energy and enhance obstacle avoidance.

There are three integers in the lattice that exist in representing f(n), g(n), and h(n), and the corresponding spots in the lattice are at the top left, bottom left, and bottom right. The traditional A* method plans the g, h, and f values of each node. Every node also includes a pink arrow pointing to its parent node. The path is initially plotted from the starting location in one direction toward the goal point. Since the 8-neighborhood search algorithm is used in this research, there are three unsearched sites that can be chosen for the following step. These locations are spread to the bottom right of the beginning point. To get the orange raster route, which corresponds to the blue path, the f(n) of these three locations is calculated, and the point with the smallest f(n) is chosen as the next site, which is the current lowest generation value node. In the traditional A* algorithm one-way pathfinding process, after searching a portion of the nodes toward the target point, the algorithm then begins to search the nodes close to the starting point again, which will result in a lower obstacle avoidance rate, longer running times, better energy consumption, and decreased overall robot performance. This is because the searchable nodes close to the starting point have high h values (far from the target point) and low g values (only a few points are accumulated); after the traditional A* algorithm has planned toward the target point for a while, the search nodes gradually approach the target point, and the h values decrease (close to the target point),
while the $g$ values gradually increase with accumulation, causing the algorithm to restart.

4.1.3. Improving the $A^*$ Algorithm. The conventional $A^*$ method is a one-way search that takes a lot of time to plan paths since it searches a lot of pointless nodes back and forth. In order to address this issue, the traditional $A^*$ algorithm is given a two-way alternating search mechanism in this paper. This mechanism alternately searches the path from the starting point and the target point until they intersect, and it can significantly cut down the amount of time spent searching for pointless nodes and increase the effectiveness of path planning. The Euclidean distance function is employed as the heuristic function for both the forward search and the reverse search. The forward estimation function targets the reverse current node, and the reverse estimation function targets the forward current node.

By exchanging the respective target points of the forward and reverse searches, we employ the technique of alternating the best current node in this study to make sure that the bidirectional search comes together close to the geometric center of the beginning point and the destination point. The alternating bidirectional searches the $15 \times 15$ raster map’s path which is planned using the $A^*$ algorithm. Continue iterating until the encounter ends.

The two-way alternating search’s precise methodology here is a description of the path search $A^*$ algorithm: (1) create the environment map first; (2) initialize the forward search and reverse search open lists, closed lists, and father lists, adding the starting point to the forward search open list and the target point to the reverse search open list, respectively; (3) determine whether the forward search and reverse search open lists are not empty; if so, go to next step; if not, the path search fails because the forward or reverse search has exhausted all of the nodes that it is capable of searching but has not yet met; (4) change the total cost $F$ function, which determines the beginning point to the reverse current node to the reverse current node, and the greedy two-way BFS algorithm replaces the two-way alternating search $A^*$ method. However, the planned route may not always be the best one. When $g(n) = 0$, the cost function transforms into a heuristic function $h$ that calculates the forward current node to the reverse current node, and the greedy two-way BFS algorithm replaces the two-way alternating search $A^*$ method. The $A^*$ algorithm is converted into a two-way Dijkstra algorithm, which seeks for a short path but inefficiently calculates a lot of nodes. The two-way $A^*$ will only search for the best path node without expanding any other nodes, which will greatly reduce the amount of computation and result in very high pathfinding efficiency, improving the heuristic function of the two-way alternating search $A^*$ algorithm. Ideally, $h(n)$ is exactly equal to the cost from the forward current node to the reverse current node.

The estimated surrogate for the bidirectional alternating search between the forward current node and the reverse current node is the straight-line distance between the two points, which is usually less than the actual path cost. The $A^*$ algorithm takes into account the proportion of the actual path cost with the position between the forward and reverse current nodes. The estimated value should be smaller when the forward current node is far from the reverse current node because the estimated generation value is much lower than the actual generation value. Conversely, when the forward current node and the reverse current node gradually get closer to one another, the estimated value should be smaller because the estimated generation value is getting closer to the actual generation value. In this sense, the cost function’s heuristic function, $h(n)$, is weighted by exponential decay in this paper.
When the forward current node and the reverse current node are separated by a great distance, $h(n)$ is greater and the weight $e$ is greater, causing the two nodes to soon collide. The estimated cost is closer to the actual cost when the forward current node and the reverse current node are close to one another.

When the forward current node meets the reverse current node, $h(n)$ is close to zero and the weight $e$ is close to one, ensuring that the forward path intersects the reverse path.

The simulation’s experimental findings explain the classic $A^*$ method, the bidirectional alternating search $A^*$
algorithm, and the bidirectional alternating search. The bidirectional alternating search uses the A* algorithm and raster maps of 15 by 15 and 100 by 100 pixels. In comparison to the bidirectional alternating search, the A* algorithm, and the standard A* algorithm, the A* algorithm with improved heuristic function \( h(n) \) examines fewer path nodes although paths will be extended. The two-way alternating search A* algorithm with improved heuristic function in the 1515 raster map reduces the average search time by 64.6 percent and 44.0 percent, respectively, and the turn angle by 33.3 percent when compared to the two-way alternating search A* algorithm, but the path length increases by 20.9 percent when compared to the traditional A* algorithm. In the 100100 raster map, the turning angle is reduced by 86.7 percent than the two-way alternating search A* algorithm, but the path length is increased by 19.6% than the traditional A*. It demonstrates that when there are obstacles in the way of the encounter that cannot be easily avoided, the two-way alternating search A* algorithm with improved heuristic function \( h(n) \) can effectively reduce the search path time and path turning angle of the two-way alternating search A* algorithm. However, the path length will increase.

### 4.2. Robot Path Planning Energy Management Research

In this paper, in addition to improving the A* algorithm for global path planning, two algorithms are used to study the robot path planning energy management. First, the overall performance of the two algorithms is compared over two years with 1000 iterations. It is found that the energy consumption of the improved A* algorithm is much higher than that of the traditional A* algorithm after continuous iterations in the case where the initial energy consumption of the A* algorithm prevails, and therefore, the improved A* algorithm has a comparative advantage in robot path planning energy management. Therefore, the improved A* algorithm has a greater advantage in robot energy management, as shown in Figure 7.

This paper also makes a relevant comparison between the first two algorithms introduced in this paper and the algorithm studied in this paper. The comparison mainly involves four aspects, which are energy management efficiency, energy consumption index, path optimization effect, and portability. It is found that both the improved A* algorithm studied in this paper and the A* algorithm have good results, as shown in Figure 8.

### 5. Conclusion

The fundamental idea behind the conventional A* algorithm is introduced in this study, along with some of its drawbacks, including its laborious calculation process, huge turning angles, and unreliable trajectory planning. A better A* method is also suggested. The two-way alternating search mechanism introduced by the improved algorithm increases the effectiveness of the search path, and the heuristic function improvement addresses the issue that the two-way alternating search A* algorithm requires more time when obstacles perpendicular to the path are present on the way to the encounter. Simulation tests in various raster environments show that the improved A* algorithm outperforms the traditional A* algorithm, genetic algorithm, and
simulated annealing algorithm in terms of search path speed while overcoming the drawbacks of numerous path turning angles and large turning angles. This makes it a practical and effective algorithm for raster map environments.

**Data Availability**

The dataset is available upon request.

**Conflicts of Interest**

The authors declare no conflicts of interest.

**References**

[1] F. Peng, L. Zheng, Z. Duan, and Y. Xia, “Multi-objective multi-learner robot trajectory prediction method for IoT mobile robot systems,” *Electronics*, vol. 11, no. 13, Article ID 2094, 2022.

[2] M. Badrikouhi and M. Bamdad, “Smooth trajectory planning based on direct collocation method for cable-driven parallel robots with central spine,” *Mechanics of Solids*, vol. 57, no. 3, pp. 652–670, 2022.

[3] Y. Luo, J. Lu, Q. Qin, and Y. Liu, “Improved IPS path optimization for mobile robots based on angle-propagation theta algorithm,” *Algorithms*, vol. 15, no. 6, p. 198, 2022.

[4] X. Hou, Z. Li, S. Guo, L. Shi, H. Xing, and H. Yin, “An improved backstepping controller with an LESO and TDs for robust underwater 3D trajectory tracking of a turtle-inspired amphibious spherical robot,” *Machines*, vol. 10, no. 6, p. 450, 2022.

[5] K. Singhal, V. Kumar, and K. P. S. Rana, “Robust trajectory tracking control of non-holonomic wheeled mobile robots using an adaptive fractional order parallel fuzzy PID controller,” *Journal of the Franklin Institute*, vol. 359, no. 9, pp. 4160–4215, 2022.

[6] F. Wang, Z. Wu, and T. Bao, “Time-jerk optimal trajectory planning of industrial robots based on a hybrid WOA-GA algorithm,” *Processes*, vol. 10, no. 5, Article ID 1014, 2022.

[7] B. Ali and D. J. Lee, “Deep reinforcement learning-based adaptive controller for trajectory tracking and altitude control of an aerial robot,” *Applied Sciences*, vol. 12, no. 9, 2022.

[8] R. Hou, J. Niu, Y. Guo et al., “A novel resolution scheme of time-energy optimal trajectory for precise acceleration controlled industrial robot using neural networks,” *Actuators*, vol. 11, no. 5, p. 130, 2022.

[9] K. Takahashi, E. Tano, and M. Hashimoto, “Feedforward-feedback controller based on a trained quaternion neural network using a generalised HR calculus with application to trajectory control of a three-link robot manipulator,” *Machines*, vol. 10, no. 5, p. 333, 2022.

[10] J. Jin and L. Qiu, “A robust fast convergence zeroing neural network and its applications to dynamic Sylvester equation solving and robot trajectory tracking,” *Journal of the Franklin Institute*, vol. 359, no. 7, pp. 3183–3209, 2022.

[11] G. Xu, H. Zhang, Z. Meng, and Y. Sun, “Automatic interpolation algorithm for NURBS trajectory of shoe sole spraying based on 7-DOF robot,” *International Journal of Clothing Science & Technology*, vol. 34, no. 3, pp. 434–450, 2022.

[12] J. Bai, J. Du, T. Li, and Y. Chen, “Trajectory tracking control for wheeled mobile robots with kinematic parameter uncertainty,” *International Journal of Control, Automation and Systems*, vol. 20, no. 5, pp. 1632–1639, 2022.

[13] J. J. Zhang, Z. L. Fang, Z. Q. Zhang, R. Z. Gao, and S. B. Zhang, “Trajectory tracking control of nonholonomic wheeled mobile robots using model predictive control subjected to Lyapunov-based input constraints,” *International Journal of Control, Automation and Systems*, vol. 20, no. 5, pp. 1640–1651, 2022.

[14] Y. You, Z. Yang, T. Zou et al., “A new trajectory tracking control method for fully electrically driven quadrobot,” *Machines*, vol. 10, no. 5, p. 292, 2022.

[15] Z. Wu, J. Chen, T. Bao, J. Wang, L. Zhang, and F. Xu, “A novel point-to-point trajectory planning algorithm for industrial robots based on a locally asymmetrical jerk motion profile,” *Processes*, vol. 10, no. 4, p. 728, 2022.

[16] W. Tang, C. Cheng, H. Ai, and L. Chen, “Dual-arm robot trajectory planning based on deep reinforcement learning under complex environment,” *Micromachines*, vol. 13, no. 4, p. 564, 2022.

[17] L. F. Carlos, D. Diaz, H. B. Jesus, A. D. Nancy, and L. F. Michel, “A metaheuristic optimization approach for trajectory tracking of robot manipulators,” *Mathematics*, vol. 10, no. 7, 2022.

[18] M. G. Krishnan and A. Sankar, “Image space trajectory tracking of 6-DOF robot manipulator in assisting visual servoing,” *Automatica*, vol. 63, no. 2, pp. 199–215, 2022.

[19] Z. Song, T. Ma, K. Qi, E. Spyrakos-Papastavridis, S. Zhang, and R. Kang, “A trajectory tracking control based on a terminal sliding mode for a compliant robot with nonlinear stiffness joints,” *Micromachines*, vol. 13, no. 3, p. 409, 2022.

[20] Z. Zhu, T. Wu, G. Zhang, S. Yu, M. Wang, and J. Yuan, “Trajectory planning of free-floating space robot based on dual-mode switching,” *Journal of Aerospace Engineering*, vol. 35, no. 3, 2022.