

# AGV path planning method based on memory precursor point and physical PID model

Cong Zhong <sup>1,\*</sup>, Mingzhen Li, Yulong Jiang  
Wuhan University of Technology, Wuhan, China

## ABSTRACT

Path planning is a core algorithm component of AGV equipment, and its quality directly affects the operating efficiency and accuracy of the equipment. The current mainstream path planning algorithm A\* and the traditional Dijkstra algorithm have the disadvantages of high computational complexity and limited accuracy. Aiming at the limitations of existing algorithms, this paper improves the Dijkstra-like algorithm by using the memorable precursor point list and introduces the kinematics model of the AGV car to optimize the PID control details. The comparison test proves that the improved algorithm can effectively improve the planning Efficiency and stability of equipment operation.

**Keywords:** AGV equipment, path planning, PID control, environment modeling, Dijkstra algorithm

## 1. INTRODUCTION

AGV (Automated Guided Vehicle) equipment refers to a transport vehicle equipped with automatic navigation devices such as electromagnetic or optical, which can drive along a prescribed navigation path and has various transfer functions. The world's first AGV was successfully developed by an American electronics company in the early 1950s, and it was used to transport goods in grocery warehouses. In 1955, British companies stopped using ground rails and invented an electromagnetic induction-guided driving method that buried wires underground. In the 1960s, European companies used computer technology to control AGV systems. AGV entered the flexible manufacturing production system from the warehouse, and AGV developed rapidly. In the mid- 1970s, it was gradually promoted into production equipment. With the development of modern logistics and manufacturing industry, AGV equipment is widely used in the field of logistics and transportation in factories, warehouses, hospitals and other places<sup>1-2</sup>.

Regardless of the driving method and mechanical design details, path planning is always one of the core issues of AGV equipment control<sup>3</sup>. The goal of path planning is to realize the autonomous navigation of the robot by choosing an appropriate path, avoiding obstacles, and optimizing path length. Due to the high degree of autonomy and flexibility of AGV equipment, as the use of equipment increases, its path planning problem becomes more and more complicated<sup>4</sup>. Although traditional path planning algorithms such as Dijkstra algorithm have been widely used, they have some limitations in complex environments, such as inability to deal with dynamic obstacles and long path search time. Therefore, how to improve the path planning algorithm to improve the efficiency and accuracy of path planning of AGV equipment has become one of the current research hotspots<sup>5</sup>.

Fazlollahtabar H studies the problem of real-time scheduling and path planning for autonomous guided vehicles. According to the work plan and the earliest delivery time rule, the station point and the AGV walking path are determined, and the AGV and available materials are considered for order processing, so as to obtain a collision-free path and minimum waiting time<sup>6</sup>. Yuan Z et al. proposed a two-level path planning algorithm to optimize the path of multiple AGVs. To make the path the shortest and reduce AGV path conflicts as much as possible, an improved A\* algorithm was first proposed<sup>7</sup>. Then a dynamic Rapid Exploration Random Tree (RRT) algorithm with kinematic constraints is proposed to obtain collision-free traversable paths in local grid graphs. Malopolski W proposed a new method to describe systems with one-way, two-way, or multi-lane flow paths<sup>8</sup>. Chaudhry IA addresses the need for simultaneous scheduling of machines and vehicles in Flexible Manufacturing Systems (FMS). A spreadsheet-based Genetic Algorithm (GA) method was proposed to solve this problem<sup>9</sup>.

\*2411814326@qq.com

This paper aims to study and improve the Dijkstra algorithm based on raster modeling . Based on the traditional algorithm, the memory and inheritance of the precursor point list can ensure the efficiency and versatility of the algorithm. At the same time, it aims at the defect that Dijkstra is easy to fall into local extremum. , introduce the typical physical model of AGV , and modify the PID parameters when the equipment is running according to the dynamic parameters provided in the model to meet the requirements of AGV equipment in practical applications, improve the efficiency and accuracy of path planning, and provide AGV equipment The practical application provides more reliable support.

## 2. ALGORITHM SCHEME DESIGN

The path planning method used in this paper is based on the basic principle of Dijkstra algorithm and its adaptive improvement and optimization. The general idea is shown in Fig. 1 below.

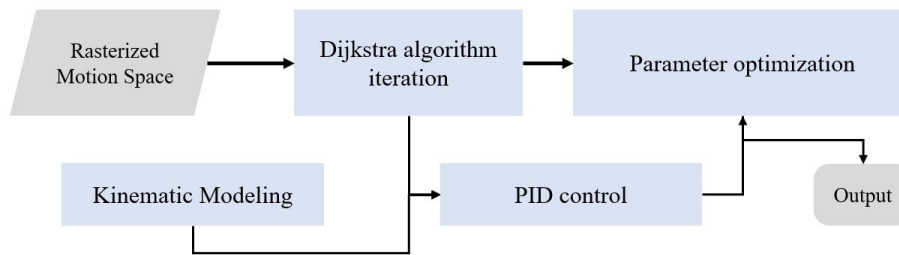


Figure 1. General idea of AGV path planning method.

Dijkstra's algorithm is an algorithm used to solve the single-source shortest path problem. Its basic principle is to gradually determine the shortest path from the source node to other nodes through a greedy strategy. In the Dijkstra planning of this article, we save the iterative precursor points of the same value by adding memory backtracking groups, and retain all the shortest paths through the relevant data of the precursor points.

During the moving process of the robot, it is necessary to continuously obtain the latest global path, and perform local path planning according to the current robot position, so as to ensure that the robot always drives in the correct direction and avoids collisions with obstacles <sup>10</sup>. When the algorithm obtains the latest global path, it traverses the relevant global path points and finds the point closest to the robot as the starting point of the local path. Then, according to the starting point of the local path and the index in the global path, a certain number of global path points can be sampled and selected as the local path <sup>11-13</sup>.

Once the initialized local path is obtained, it can be transformed into a series of continuous trajectory points using an interpolation method for the actual movement of the robot. Then, by establishing the AGV physical model, the driving trajectory of the planning algorithm is obtained. The lateral and longitudinal velocities of the robot are calculated from the corrected angular misalignment, limiting its rotational velocity. And through the PID adjustment parameters, the relevant driving information is sent to the equipment chassis, and the parameters are constantly adjusted during the movement to obtain a higher-quality trajectory and a smoother movement process.

In this paper, the Mecanum-wheel mobile robot is selected as a typical example of AGV equipment, and the forward kinematic equation and inverse kinematic equation of the corresponding robot are established. On the local path, we add motion optimization constraints such as maximum speed and maximum acceleration to ensure that the robot will not experience unstable or large overload behaviors such as overspeed or sudden braking during motion. By establishing the kinematic equations of the robot and adding motion optimization constraints, the effect and accuracy of path planning can be further improved, so that the robot can complete tasks more stably, safely, and efficiently in actual motion. By adding corresponding motion optimization constraints to adjust the driving parameters of the motor, the motion optimization on the local path is realized, thereby further improving the effect of the algorithm.

### 3. ALGORITHM IMPLEMENTATION DETAILS

#### 3.1 Grid modeling of motion space

Before path planning, one must first understand the working environment of the AGV and build a reasonable map model<sup>14</sup>. In the process of map construction, it is necessary to identify and model various obstacles in the environment, so that obstacles can be avoided, and an appropriate path can be selected during path planning. At the same time, it is also necessary to adjust the resolution of the map so that the characteristics of the environment can be described more accurately during path planning. The final map model constructed should be able to accurately reflect the actual situation of the environment, including various obstacles, the size of the channel, the height of the terrain and other information. In the AGV application scenario, we can simplify the model. Since the range of motion is limited to a plane, so we don't need detailed obstacle information, the most important task is to distinguish obstacles and passable road information in the motion area.

Considering the regularity of obstacles in the motion scene, we use the binarization method to perform rasterization operations on the motion space to facilitate the subsequent path planning process<sup>15</sup>.

Divide the map into a network structure of equal size, and mark obstacles and passable areas through  $\{0, 1\}$  set, black represents impassable obstacles, and white represents passable areas entering the path planning solution stage. Taking the  $30 \times 30$  grid area as an example, using the function  $P_n = \{x_n, y_n, t_n\}$ ,  $n \in \{0, 1, \dots, 900\}$ ,  $t_n$  carries the attribute information corresponding to the grid. An example of a motion space after rasterization is shown in Fig. 2 below.

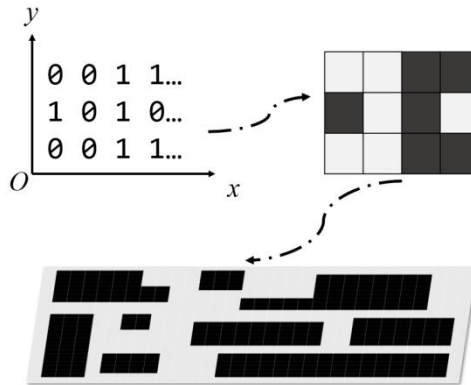


Figure 2. Schematic diagram of the motion space after the graph is rasterized.

#### 3.2 The introduction of the precursor point list of the algorithm scheme

As mentioned above, we first give the core operation process of the Dijkstra algorithm in Fig. 3.

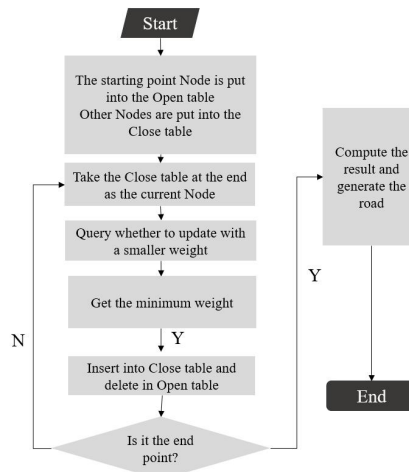


Figure 3. Dijkstra algorithm basic idea.

As shown in the figure, in a typical Dijkstra algorithm calculation process, the distance of the source node is first set to 0, and the distances of other nodes are set to infinity. Then, the algorithm selects the node closest to the source node among the unvisited nodes and updates the distance of the neighbor nodes of this node. If the updated distance is shorter than the original distance, update the distance of the neighbor node. The algorithm will continue to select the node closest to the source node among the unvisited nodes and update the distance of its neighbor nodes until all nodes are visited. In this way, when the algorithm is completed, each node has a distance value, and finally the system calculates and filters the length of the shortest path through the weighted distance value of the algorithm<sup>16-17</sup>.

In the traditional Dijkstra algorithm, our ultimate constraint goal is to obtain a distance-optimal path, so the main focus in the algorithm process is to obtain the path without giving too much reuse to the intermediate nodes in the iterative process. We need to save the shortest path and predecessor node information of each node. Specifically, in the iterative process of the algorithm, for each node, the shortest distance from the starting point to the node and the predecessor nodes of the node in the shortest path need to be saved.

In practical situations, there may be multiple nodes with the same shortest distance. To solve this problem, we introduce an improvement that uses memory backtracking groups to save the iterative predecessor points of all nodes with the same shortest distance. In this way, all the shortest paths can be preserved by saving the relevant data of the predecessor points.

Specifically, when we encounter multiple nodes with the same shortest distance during the iterative execution, we store all their predecessors in a memory backtracking group. This group can be viewed as a set of key-value pairs, where each key is a node with the same shortest distance, and the value corresponding to each key is the set of predecessors of the node. In subsequent operations, we can obtain the set of predecessor points for each node with the same shortest distance by checking the memory backtracking group. In this way, when we find the end point, we can obtain the set of predecessor points of the end point from the memory backtracking group, and trace back along these precursor points to obtain all the shortest paths to the end point.

The algorithm implementation is shown in the Fig. 4 below.

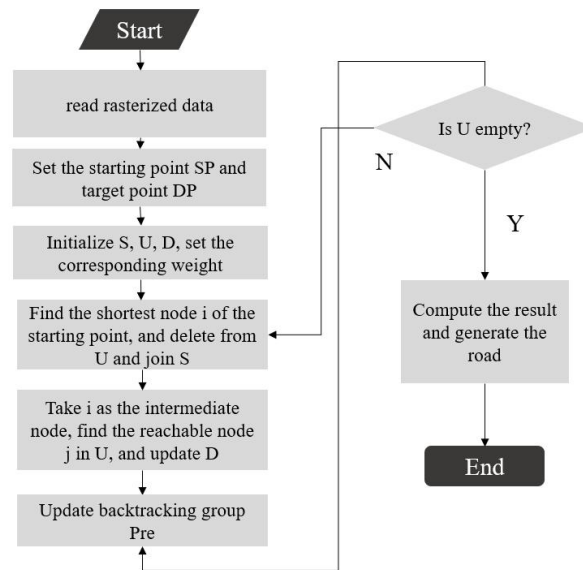


Figure 4. The algorithm implementation process of adding precursor table.

By using the memory backtracking group, all the shortest paths can be saved, and these paths can be found quickly when needed, which improves the efficiency and accuracy of Dijkstra's algorithm. The screening constraints depend on the kinematic modeling of the device, which is given below.

### 3.3 Kinematic modeling of typical AGV equipment

When the equipment moves under ideal conditions, we assume that the working environment is good, then the mecanum wheel will not idle due to the suspension of the device, and at the same time, the fuselage will not undergo large-scale deformation. We establish a global coordinate system and fix it on the equipment. The local coordinate system itself, decomposing its motion components, the velocity of the geometric center is represented by the following matrix:

$$v_t = [v_{tx} \ v_{ty} \ w]^T$$

Among them ,  $v_{tx}$  and  $v_{ty}$  are the horizontal velocity of the robot, and  $w$  is the angular velocity of the robot. Let the speed of the wheel be  $v_1, v_2, v_3, v_4$ ; the rotational speed be  $w_1, w_2, w_3, w_4$ ;  $r$  is the vector from the geometric center of the robot chassis to the wheel axis;  $v_r$  is the velocity component of the wheel axis along the direction perpendicular to  $r$ ;  $r_x$  and  $r_y$  are the projection distances of  $r$  on the x-axis and y-axis of the geometric center coordinate system. The velocity matrix of four mecanum wheels can be expressed as follows [15] .

$$\begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \end{bmatrix} = \frac{1}{R} \begin{bmatrix} 1 & 1 & r_x + r_y \\ 1 & -1 & -(r_x + r_y) \\ 1 & 1 & -(r_x + r_y) \\ 1 & -1 & r_x + r_y \end{bmatrix} \begin{bmatrix} v_{tx} \\ v_{ty} \\ w \end{bmatrix}$$

In the above formula,  $R$  represents the radius of the Mecanum wheel . To express the positive kinematic solution of the device, we need to solve the above formula to obtain the following positive kinematic equation.

$$\begin{bmatrix} v_{tx} \\ v_{ty} \\ w \end{bmatrix} = \frac{R}{4} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & -1 & -1 & 1 \\ r_x + r_y & r_x + r_y & r_x + r_y & r_x + r_y \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \end{bmatrix}$$

At the same time, the relative parameters of the local path and heading angle deviation can be obtained through the absolute coordinates of the space. Every time the algorithm selects a global path, it will intercept a section of path forward based on the local target point as the local path of the robot this time, and then calculate the lateral deviation, longitudinal deviation and angular error. Assume that the coordinates of the local target point are  $(x_g, y_g, \theta_g)$  in the map coordinate system , and the coordinates of the robot's current position are  $(x_c, y_c, \theta_c)$  .  $\Delta x$  is the absolute value of the longitudinal deviation between the local target point and the current position of the robot;  $\Delta y$  is the lateral deviation between two points;  $\Delta\theta$  is the heading angle deviation, that is, the angular deviation between the current position of the robot and the local target position; The distance between the position and the local target point can be calculated as follows.

$$\begin{bmatrix} \Delta x \\ \Delta y \\ \Delta\theta \\ \Delta l \end{bmatrix} = \begin{bmatrix} |x_g - x_c| \\ y_g - y_c \\ \tan^{-1} \left| \frac{y_g - y_c}{x_g - x_c} \right| \\ \sqrt{(x_g - x_c)^2 + (y_g - y_c)^2} \end{bmatrix}$$

Based on the error of the controlled system, by continuously adjusting the three parameter values of proportional P , integral I and differential D , the effective output value of the control variable can be calculated and then input into the system to realize the steady-state operation of the system as Fig. 5.

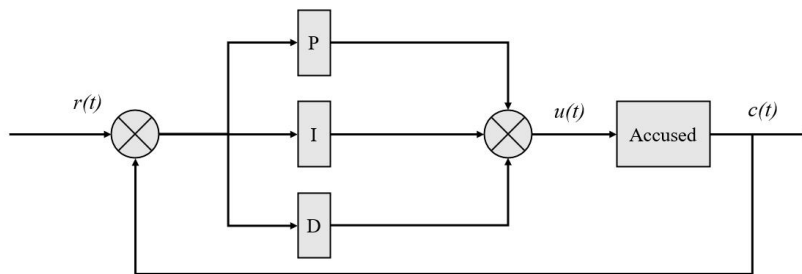


Figure 5. PID basic control principle.

The PID controller is composed of proportional, integral, and differential coefficients, which are expressed as  $K_p$ ,  $K_i$  and  $K_d$  respectively. The PID controller completes the adjustment of the system by controlling the deviation between the input and output results, and  $u$  represents the controlled quantity. Then the controller is expressed by the following formula.

$$u(t) = K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{de(t)}{dt}$$

In the example of this paper, to facilitate the iterative implementation of the algorithm, the discrete form of PID needs to be adopted, and the formula is as follows.

$$v = K_p e(k) + K_i \sum_{j=0}^k e(j) + k_d (e(k) - e(k-1))$$

In summary, through the kinematics model of the AGV trolley, the motion parameters required for local control can be obtained, and the real-time error adjustment control is performed through the PID algorithm to ensure the stability of the motion.

#### 4. ALGORITHM EFFICIENCY AND STABILITY TESTING

In this section, we evaluate the performance of the algorithm by testing the effect and accuracy of path planning. Firstly, the path generation time of the algorithm including the map rasterization operation is tested, and the path length is compared with the traditional algorithm<sup>18</sup>. Secondly, it is necessary to monitor the kinematic characteristics of the AGV car on the local path, including important parameters such as speed, acceleration and steering angle, and record these data in real time during the movement. Through the analysis of these data, we can understand whether the trajectory of the AGV car meets the design requirements and evaluate the overall effect of the algorithm according to the experimental results to guide the subsequent optimization work.

We prepare 8 groups of AGV scene atlases with the same size. These scene atlases include different types of scenes, such as open space, closed space, complex obstacles, etc., to evaluate the performance of the classic Dijkstra algorithm and the improved Dijkstra algorithm in different scenes. During the running of the algorithm, the time for the classic Dijkstra algorithm and the improved Dijkstra algorithm to generate paths in each scene atlas are recorded, and their performances are compared in Fig. 6.

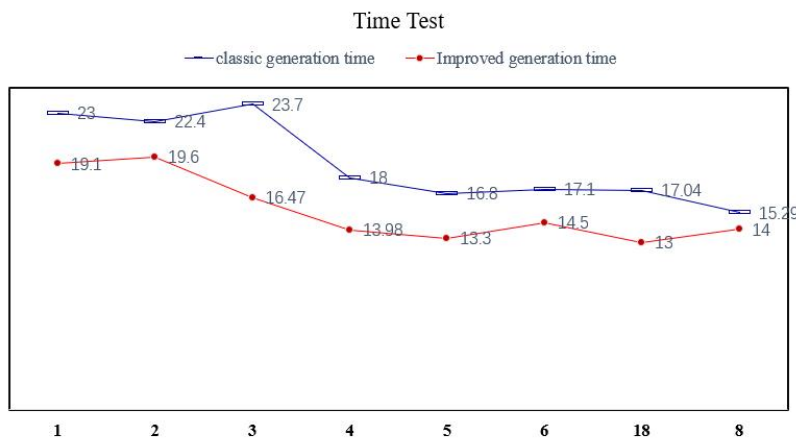


Figure 6. Average graph path generation time.

Provide the basis for the parameter input of our AGV motion simulation. We build the simulation model of AGV and assign Expresso script to the Mecanum wheel as Fig. 7.

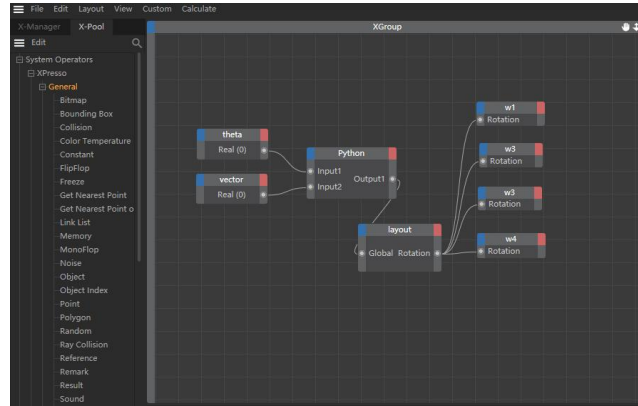


Figure 7. Espresso script parameter passing process.

In the running parameters tab of Espresso, we record the kinematic parameters such as the linear velocity and angular velocity of the McNerg wooden wheel converted by the script. By running the AGV model in the simulator and recording the total running time of the car to evaluate the performance of the model. At the same time, we collect and analyze the motion data in the simulator to evaluate the motion characteristics of the simulated AGV car, such as speed, acceleration, steering angle, etc. Finally, we will summarize the collected data and draw the corresponding statistical chart as following Fig.8 and Fig. 9.

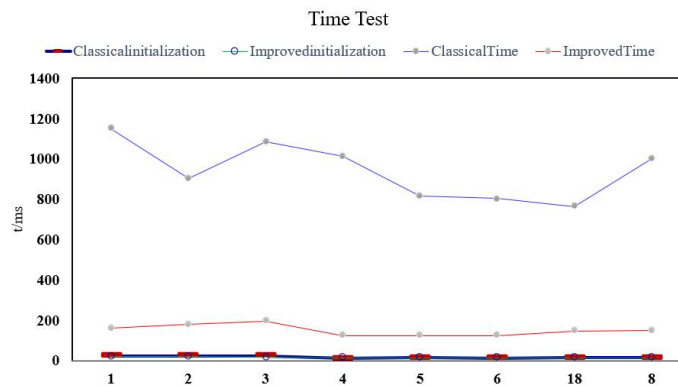


Figure 8. AGV model running time.

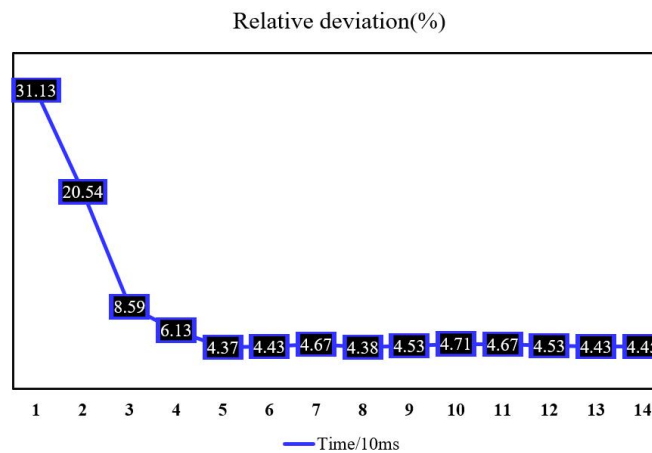


Figure 9. PID control situation.

We can draw relevant conclusions from the above figure. In terms of generation time, since the same grid modeling method is basically adopted, the motion space generation time of the improved algorithm is basically the same as that of

the classical algorithm. In terms of path generation speed, although the improved Dijkstra algorithm needs to sacrifice part of the space complexity as the map complexity increases due to the record of the precursor point list containing the nodes before the iteration, it can generate better quality in a relatively brief time path of. At the same time, because the actual situation of the kinematics of the AGV equipment is fully considered through the PID control method during the operation of the equipment, the AGV equipment of the algorithm in this paper can move with a relatively stable self-pose when moving, avoiding the robot's A large change in attitude when turning can greatly shorten the navigation time and improve the success rate of navigation at the same time.

## 5. CONCLUSION

Dijkstra algorithm is a classic single-source shortest path algorithm, which can find the shortest path from one source node to other nodes in the weighted graph, but the classic Dijkstra algorithm will have certain limitations due to the characteristics of the algorithm itself in the application scenario of AGV equipment. When applying the algorithm to representative AGV equipment with Mecanum wheel as the chassis, there are problems such as poor global path ability when navigating, too large turning radius, and unfriendly local path control performance. In view of these problems, this paper proposes an improvement scheme for application scenarios. By introducing a memorable list of precursor points, the optimal path under the same weight is recorded, and then the parameter constraint conditions in line with the kinematic control stability are added according to the established AGV equipment kinematics model, to optimize the quality of the global path generated by the algorithm.

In the process of realizing AGV navigation, local path control is a very important part. It can fine-tune the movement attitude of AGV equipment to ensure the stability and accuracy of the equipment during driving. In order to achieve local path control, we introduced the PID control method, through the monitoring and feedback of the attitude of the car, and the adjustment of the speed of the car according to the error signal, so that the car can maintain stability and precision during driving. PID control can be finely adjusted according to the actual situation to adapt to different working environments and application scenarios, thereby improving navigation efficiency and safety, and realizing the optimization of navigation efficiency and safety.

In terms of effect testing, we established the simulation model of the AGV equipment, and passed the kinematic parameters of the real-time calculation through the Espresso label and simulated the motion of the simulation equipment through the corresponding parameters and recorded the parameters and the overall running time Measure the application effect of the improved algorithm.

After the algorithm efficiency and stability detection, it can be concluded that by introducing a memorable precursor point list and PID control, we can make the AGV equipment more accurate and reliable during the navigation process in the actual scene and improve work efficiency and work quality. Through these improvements , we can make the driving process of the equipment more stable, reduce the driving error caused by attitude changes, and improve the stability and accuracy of the AGV equipment when moving on a local path. The optimization of navigation efficiency and safety has been realized as much as possible.

## REFERENCES

- [1] Wang, Y., Research on dynamic path optimization of trailer-type multi-AGV system based on time window [D]. Jilin University (2018).
- [2] Tai, Y. P., Xing, K. X., Lin, Y. G., et al., Research on multi-AGV path planning method [J]. Computer Science, (62): 84-87 (2017).
- [3] Sun, S. K., Research on multi-AGV conflict-free and deadlock-free path planning based on CROTPN in logistics center [D]. Harbin Institute of Technology (2018).
- [4] Le-Anh, T., Koster, D., A review of design and control of automated guided vehicle Systems[J]. European Journal of Operational Research, 171(1):1-23 (2006).
- [5] Fazlollahtabar, H., Saidi-Mehrabad, M., Methodologies to optimize automated guided vehicle scheduling and routing problems: a review study. Journal of Intelligent & Robotic Systems, 77, 525-545 (2015).



- [6] Fazlollahtabar, H., Hassanli, S., Hybrid cost and time path planning for multiple autonomous guided vehicles[J]. *Applied Intelligence*, 48, 482-498 (2018).
- [7] Yuan, Z., Yang, Z., Lv, L., A Bi-Level Path Planning Algorithm for Multi-AGV Routing Problem[J]. *Electronics*, 16(3): 2-10 (2020).
- [8] Malopolski, W., A sustainable and conflict-free operation of AGVs in a square topology [J].*Computers & Industrial Engineering*, 26(9): 1128-1133 (2018).
- [9] Chaudhry, I. A., Mahmood, S., Shami, M., Simultaneous scheduling of machines and automated guided vehicles in flexible manufacturing systems using genetic algorithms[J]. *Journal of Central South University of Technology*, 18(5): 1473-1486 (2011).
- [10] Yang, H., Qi, J., Miao, Y., Sun, H., Li, J., A new robot navigation algorithm based on a double-layer ant algorithm and trajectory optimization[J]. *IEEE Transactions on Industrial Electronics*, 66(11):8557-8566 (2018).
- [11] Wang, L., Xu, H. X., Cao, J. H., et al., Improved ant colony-genetic algorithm for information transmission path optimization in remanufacturing service system[J]. *Chinese Journal of Mechanical Engineering*, 31(6):106-117 (2018).
- [12] Zhang, S., Liao, Z. H., Wu, Y., et al., Mobile robot path planning compares wavefront and A\* algorithms[J]. *Journal of Physics: Conference Series*, 1948(1):012046 (2021).
- [13] Chang, L., Shan, L., Jiang, C., et al., Reinforcement based mobile robot path planning with improved dynamic window approaching unknown environment[J]. *Autonomous Robots*, 45(1):51-76 (2021).
- [14] Fox, D., Burgard, W., The dynamic window approach to collision avoidance[J]. *IEEE Robotics & Automation Magazine*, 4(1):23-33 (1997).
- [15] Zhao, M., Zheng, Z. Y., Mo, Q. F., et al., Path Planning Method for Mobile Robots Based on Improved Artificial Potential Field Method [J]. *Computer Application Research*, 37(S2):66-72 (2020).
- [16] He, L. N., Lou, P. H., Qian, X. M., et al., Collision-free path planning for automatic guided vehicles based on time window[J]. *Computer Integrated Manufacturing Systems*, 16(12):2630-2634 (2010).
- [17] Wang, Z. Y., Zeng, G. H., Huang, B., et al., Global Optimal Path Planning for Robots Based on Improved A\* Algorithm [J]. *Computer Applications*, 39(09):2517-2522 (2019).
- [18] Li, T. T., Ning, P. F., Niu, P. J., Factory AGV Safety Path Planning Based on Improved Genetic Algorithm [J]. *Combined Machine Tool and Automatic Processing Technology*, (03): 160-163 (2020)