Self-Organized Multi-Robot Path Planning and Chain Distribution Based on Improved DWA and A* Fusion in Unknown Space

Bowen Ding[1] and Guoliang Wang[2,*]

[1]School of Information and Control Engineering, Liaoning Petrochemical University, Fushun 113001, China

[2]School of Information and Control Engineering, Liaoning Petrochemical University, Fushun 113001, China

Abstract: With the rise of autonomous driving in recent years, path planning has gained widespread attention. Traditional path planning needs to be based on a large amount of known information, which is not available for confined environments. Taking the complex indoor space where GPS cannot be used as the research background, the article designs a self-organised motion scheme for multi-intelligent body trolleys that includes exploration and path planning. By improving the DWA and A* algorithms, the multi-robot self-organisation achieves reasonable path planning, and the fusion of the two algorithms solves the contradictory problems of global planning being unable to avoid dynamic obstacles and local planning possibly falling into local optimum. After that, the pilot-following algorithm is added to guide the multi-intelligence body to operate in formation. By studying the constraints of hardware such as LiDAR and machine trolleys, the chain distribution of multiple intelligences is proposed to solve the problem of information loss caused by the discontinuous monitoring field of view. Eventually, when the carts are all in position, the whole area is covered and monitored using sensor fusion with multiple viewpoints. The feasibility of the explored scheme is verified by simulation experiments, and the feasibility and robustness of multi-sensor fusion is verified by specific hardware.

Keywords: Multi-intelligentsia, Formation, Obstacle avoidance exploration, Path planning, A*, DWA, Pilot following.

1. INTRODUCTION

With the rapid leap of science and technology and the booming development of artificial intelligence field, information visualisation technology is increasingly becoming a bridge connecting data and cognition, and its social demand is more and more extensive and urgent [1]. In this context, information perception system plays a pivotal role as the core hub. Tracing back to 1994, the United States took the lead in completing the development and application of the GPS global positioning system, a milestone achievement that led the revolution in positioning technology, made precise positioning an indispensable part of daily life, and also provided strong support for the huge work of path planning.

In recent years, more and more research on path planning has turned to indoor space. Facing the complex indoor environment and the layers of steel and concrete blocking the signals, it is not possible to achieve the accurate path planning task indoors by copying the outdoor mode of using GPS. Since the beginning of the 21st century to the present after unremitting exploration and innovation, indoor path planning technology has made significant progress, building a relatively mature solution system, but today's traditional path planning programme has a difficult to avoid the problem is the need for a large amount of known information or a high degree of cooperation with personnel, once the loss of pre-information collection will make it difficult for the intelligent body to move forward. For path planning in unknown areas, its potential application scenarios are extremely wide, spanning military strategic deployment, emergency rescue in small spaces, and even wildlife ecological monitoring and other key areas, showing its immeasurable value [2]. In view of this, the article precisely focuses on this demand gap, and innovatively proposes a multi-intelligent body self-organised path planning algorithm in unknown space.

The core of the technology framework is built by high-precision LiDAR sensors in collaboration with a set of self-organised robotic carts. These intelligent vehicles are capable of moving in autonomous formation, with the front vehicle leading the rear vehicle, exploring unknown environments efficiently by using the precise scanning capability of LIDAR, and also intelligently adjusting the deployment strategy according to the complex and changing environmental conditions to achieve penetration and scanning of unknown areas. On this basis, they can work together to accurately track and locate moving targets in the area, demonstrating excellent real-time response capability and positioning accuracy. The main challenges addressed in this paper have two main parts: firstly, how to use multiple intelligences to explore in an unknown space and maintain the avoidance of static as well as dynamic obstacles, and finally arrive at the target area or complete the exploration; secondly, how to use the linkage of multiple robots on the way of exploration to continuously monitor the space, and how to realise the joint work between multiple intelligences in the presence of signal interference.

In this paper, the efficacy and robustness of the proposed system is systematically verified through a

*Address correspondence to this author at the School of Information and Control Engineering, Liaoning Petrochemical University, Fushun 113001, China; E-mail: glwang@lnpu.edu.cn

E-ISSN: 2409-9694/24

well-designed simulation environment, complemented by detailed comparative analyses of test and experimental data. The experimental results show that the technology module of the system exhibits excellent obstacle avoidance and path planning capabilities when exploring unknown regions. Specifically, the module not only achieves a high degree of real-time responsiveness to ensure instantaneous decision-making, but also significantly optimises the generation strategy of obstacle avoidance paths, resulting in smoother and more efficient trajectory characteristics. This optimisation not only overcomes the limitations of traditional path planning in static environments, but also effectively solves the local optimal dilemma faced by path planning in dynamic environments, and realises the comprehensive optimisation of static and dynamic obstacles, thus comprehensively improving the adaptability and practicability of the path planning scheme. In summary, this study provides a strong support for improving the performance and reliability of path planning.

1.1. Domestic and International Research Status

This paragraph introduces the methods and origins of path planning and multi-intelligentsia collaboration, including the research results in recent years at home and abroad, the advantages and disadvantages between the methods and the improvement of the methods.

The widespread application of intelligent body technologies is profoundly reshaping the landscape of various industries, and the concept of collaborative multi-intelligent body localisation has continued to attract extensive attention and in-depth exploration in the research community since it was pioneered by Fox and Bekey *et al*. in 2000 [3]. Compared to single intelligences, although they are also capable of sensing the environment through path planning and sensor technologies, their capabilities are often limited by the limitations of a single detection by a camera or a laser. The rise of multi-intelligent body systems not only greatly enriches the dimension and depth of environment sensing, but also builds a more stable and reliable localisation network through distributed information sharing and strategy collaboration. This collaborative mechanism enables multi-intelligent bodies to respond quickly, collaborate efficiently, and complete tasks quickly and accurately in unknown areas.

For indoor path planning research is mainly divided into two parts, dynamic path planning based on local and global path planning based on static information. The traditional static path planning algorithms are Dijkstra, A*, D*, RRT [4], PRM [5], ant colony algorithm and so on. Dijkstra's algorithm, proposed in 1959, is a shortest path algorithm from one vertex to each of the remaining vertices, which solves the shortest path problem in the power graph. Later, by fusing the advantages of the best-first search algorithm and Dijkstra's algorithm, the A* search algorithm was first proposed by Peter Hart, Nils Nilsson, and Bertram Raphael [6] of Stanford Research Institute in 1968 to solve the static global path planning problem.

Traditional dynamic path planning mainly includes DWA, potential field method [7], etc. Fox [8] *et al.* proposed DWA, which samples and combines the velocities in the space to compute multiple moving routes and selects the optimal trajectory based on the robot's motion model as well as the current motion state. Rosmann *et al*. [9] proposed a multi-objective optimisation algorithm to ensure the smoothness of the robot's motion.

In the field of path planning, whether dynamic or static, a single method is often difficult to cope with the challenges of complex and changing scenes, which is directly related to the success of path planning. For the static global path planning mainly based on A* algorithm, which is the result of analysing and calculating the global information in advance, it is not able to respond in time to unexpected situations, and it is difficult to avoid dynamic obstacles, and it is not able to replace more suitable paths in the changing scenes. As for the local dynamic path planning based on DWA algorithm, although it satisfies the timely response to the dynamic scene by continuously calculating and deciding the next movement route, due to the loss of global information and the insufficient length of prediction, it will fall into the local optimal solution in the face of certain obstacles, resulting in the inability to avoid the obstacles and thus failing to arrive at the target position.

To address the above problems, many researchers have conducted a lot of studies. For example, Jiajun Xu *et al* [10] proposed a path planning method combining the improved A algorithm and the TEB algorithm. By introducing a virtual expansion region and optimising the inflection point selection strategy, the efficiency of the search and the security of the path are improved. In addition, the fusion of the TEB algorithm enables the method to achieve path planning in dynamic environments, but the method is mainly applicable to robots with Ackermann structures. Tao Zhang *et al.* [11] improved Algorithm A by extending its search neighbourhood to 25 directions and simplifying the 35 search directions to 9 to improve the operation efficiency, but this improvement may lead to ignoring the smaller obstacles around. Xu Wei *et al.* [12], on the other hand, combined the simulated annealing

algorithm with the A* algorithm to complete the static path planning and improved the trajectory evaluation function of the DWA algorithm, which ultimately achieved the local obstacle avoidance function. In this paper, the A* algorithm is improved to change the path planning using global information to local global path planning using known information, which is applicable to the path planning under unknown space mentioned in this paper. Moreover, this paper improves and optimises the DWA algorithm to find a dynamic path planning suitable for the self-organised formation movement of multiple intelligences in this paper by constructing different evaluation functions, so that each intelligent body considers the subsequent monitoring of the unknown space while participating in the obstacle avoidance, and dynamically combines the improved two methods to guide the trolley to avoid the obstacles as well as the dynamic global path planning, which is able to meet the requirements of avoiding the dynamic obstacle and at the same time jumping out of the local optimal solution as much as possible.

2. MODEL DESIGN AND OPTIMISATION

In this chapter, we detail the design and use of the multi-intelligent body self-organised motion and path planning model in the paper through several subsections, focusing on the pilot-following algorithm, the improved and fused DWA and A* path planning algorithms. A detailed description of how it can be improved and how it can be implemented is given.

2.1. Multi-Intelligent Body Collaboration and Path Planning

In this subsection it is presented in two main parts, one focuses on the design and application of the pilot-follow algorithm around multi-intelligence self-organising formations, and the other focuses on how to improve the A* and DWA algorithms and integrate the two to use this model as the main algorithm for guided path planning.

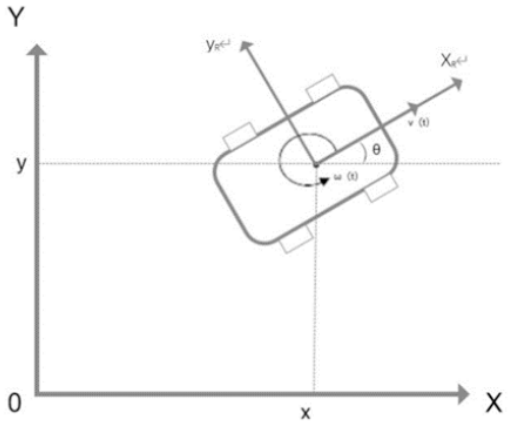


Figure 1: Intelligent Trolley Motion Parameters.

2.1.1. Pilot-Follow Method

In this paper, the pilot-following method is used to realise the formation of multiple intelligences in unknown space. For the pilot-following method, the first thing involved is the equations of motion of the multi-intelligent body, this paper plans to use the Mecanum wheel as the motion device of the intelligent body, and the motion model of the unmanned vehicle is shown in Equation 1

$$\begin{bmatrix} X(t+\Delta t) \\ Y(t+\Delta t) \\ v_x(t+\Delta t) \\ v_y(t+\Delta t) \\ \theta(t+\Delta t) \\ \omega(t+\Delta t) \end{bmatrix} = \begin{bmatrix} x(t) + v_x(t) \cdot cos(\theta(t)) \cdot \Delta t - v_y(t) \cdot sin(\theta(t)) \cdot \Delta t \\ y(t) + v_x(t) \cdot sin(\theta(t)) \cdot \Delta t - v_y(t) \cdot cos(\theta(t)) \cdot \Delta t \\ v_x(t) + a_x(t) \cdot \Delta t \\ v_y(t) + a_y(t) \cdot \Delta t \\ \theta(t) + \omega(t) \cdot \Delta t \\ \omega(t) + \alpha(t) \cdot \Delta t \end{bmatrix} \quad (1)$$

where $x(t)$ and $y(t)$ are the horizontal and vertical coordinates of the cart at the moment t respectively, $X(t+\Delta t)$ is the displacement of the cart in the X direction from the moment t to the moment $(t+\Delta t)$, Δt is a time interval, and similarly $Y(t+\Delta t)$ is the displacement of the cart in the Y direction;

$v_x(t)$ and $v_y(t)$ are the velocities of the cart in the x-direction (positive direction) and y-direction (perpendicular positive direction) of the cart at the moment t. $v_x(t+\Delta t)$ is the velocity of the cart in the x-direction at the moment $(t+\Delta t)$, and similarly $v_y(t+\Delta t)$ is the velocity in the y-direction;

$a_x(t)$ and $a_y(t)$ are the accelerations of the cart in the x and y directions;

$\theta(t)$ is the angle between the x-direction of the cart and the x-direction of the coordinate system at moment t, $\omega(t)$ is the angular velocity of the cart's rotation, and $\alpha(t)$ is the angular acceleration of the cart.

In the pilot-following formation exploration strategy, the lead vehicle acts as a pilot to lead the direction, while the rear vehicle shows a high degree of intelligence and adaptability, not simply copying the driving trajectory of the vehicle in front mechanically (*i.e.*, not pure trajectory following). Rather, it calculates a following path to follow the front vehicle according to its moving direction, distance, and various types of speeds, which not only needs to satisfy the function of following the rear vehicle, but also makes use of the path planning algorithm to realise obstacle avoidance and collision avoidance, which ensures the stability of the formation and greatly improves the safety and efficiency of the overall exploration task

Figure **2** is a schematic diagram of the positional relationship between the pilot car and the virtual car, where the position of the virtual car needs to be set by

the pilot car, while Figure **3** is a schematic diagram of the positional relationship between the pilot car, the virtual car and the following car, where the ultimate goal of the following car is to reach the position of the virtual car.

The final position to be reached by the follower is the position of the virtual robot R_V, and not the position of the pilot robot R_L

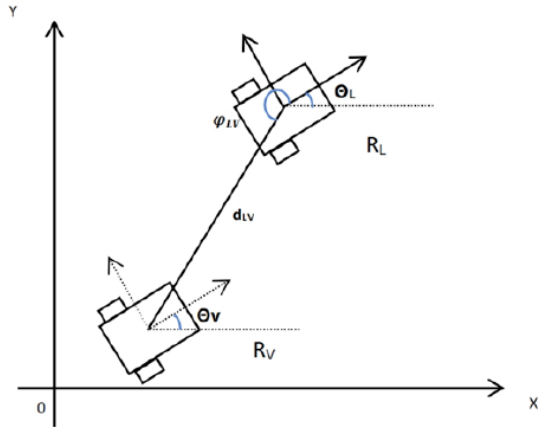


Figure 2: Pilot Vehicle and Virtual Vehicle Position Relationship.

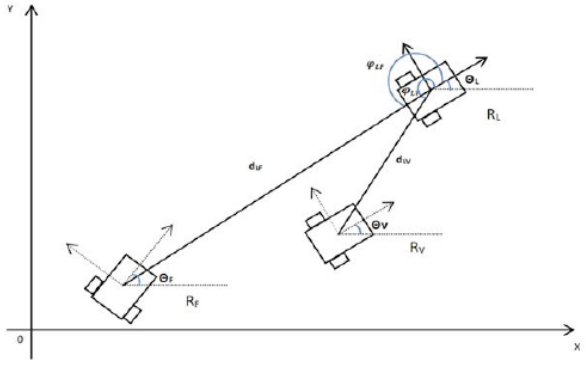


Figure 3: Pilot vehicle, virtual vehicle and follower vehicle position relationship.

Since the position of the virtual robot R_V needs to change with the change of the position of the navigator robot R_L, which is a relative following position, we need to project based on the position of the navigator *(xL, yL)* and the correspondence between the navigator robot and the virtual robot. The specific steps are as follows:

$$\begin{cases} x_V = x_L + d_{LV}\cos\ (\varphi_{LV} + \theta_L) \\ y_V = y_L + d_{LV}\sin\ (\varphi_{LV} + \theta_L) \\ \theta_V = \theta_L \end{cases} \tag{2}$$

x_V, y_V are the horizontal and vertical coordinates of the virtual robot; x_L, y_L are the horizontal and vertical coordinates of the pilot robot; d_{LV} is the length of the line connecting the pilot robot with the centre of the virtual robot; d_{LF} is the length of the line connecting the pilot robot with the centre of the following robot; φ_{LV} is the angle between the x-direction of the pilot robot and d_{LV}, and φ_{LF} is the angle between the x-direction of the pilot robot and d_{LF}; θ_L is the angle between the pilot is the angle between the x-direction of the pilot robot and the X-direction of the coordinate axis, and θ_V is the angle between the x-direction of the simulated robot and the X-direction of the coordinate axis.

Using the above results we can get the relationship between the virtual robot R_V and the following robot R_F, but for the multi-intelligent body movement involves a variety of problems such as perception, control, motion, environment, etc., so there must be a position and angle deviation between the virtual robot R_V and the following robot R_F, that is, the state error:

$$\begin{cases} x_{eVF} = x_V - x_F \\ y_{eVF} = y_V - y_F \\ \theta_{eVF} = \theta_{V\_}\theta_F \end{cases} \tag{3}$$

x_{eVF}, y_{eVF}, θ_{eVF} are the state errors of the virtual robot and the following robot, respectively; x_V, x_F, y_V, y_F are the horizontal and vertical coordinates of the virtual robot and the following robot, respectively; θ_V and θ_F are the observation angles of the virtual robot and the following robot, respectively.

The state errors of the virtual robot and the following robot in the global coordinate system are converted to a reference coordinate system based on the position of the following robot by means of a transfer matrix, resulting in a state error expression:

where e_x,e_y,e_θ are the errors in the x,y direction and the angle.

$$\begin{bmatrix} e_x \\ e_y \\ e_\theta \end{bmatrix} = \begin{bmatrix} \cos\theta_F & \sin\theta_F & 0 \\ -\sin\theta_F & \cos\theta_F & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_V - x_F \\ y_V - y_F \\ \theta_V - \theta_F \end{bmatrix} \tag{4}$$

$$\begin{cases} x_{eVF} = \cos\theta_F(x_V - x_F) + \sin\theta_F(y_V - y_F) \\ y_{eVF} = -\sin\theta_F(x_V - x_F) + \cos\theta_F(y_V - y_F) \\ \theta_{eVF} = \theta_{v\_}\theta_F \end{cases} \tag{5}$$

$$\begin{cases} x_{eVF} = d_{LV}\cos(\varphi_{LV} + \theta_L - \theta_F) - d_{LF}\cos(\varphi_{LF} + \theta_L - \theta_F) \\ y_{eVF} = d_{LV}\sin(\varphi_{LV} + \theta_L - \theta_F) - d_{LF}\sin(\varphi_{LF} + \theta_L - \theta_F) \\ \theta_{eVF} = \theta_L - \theta_F \end{cases} \tag{6}$$

$$\begin{cases} \dot{x}_{eVF} = V_L\cos(\theta_L - \theta_F) - V_F + \omega_L d_{LV}\sin(\varphi_{LF} + \theta_L - \theta_F) \\ \dot{y}_{eVF} = V_L\sin(\theta_L - \theta_F) - \omega_F x_{eVF} + \omega_L d_{LV}\cos(\varphi_{LF} + \theta_L - \theta_F) \\ \dot{\theta}_{eVF} = \omega_L - \omega_F \end{cases} \tag{7}$$

The collation yields (5); Substituting equation (2)(3) into (5) gives (6)

For the state error, it should be minimised or eliminated by controlling the follower, so the derivation of equation (6) is used to obtain the relationship

between the state error and the follower's linear and angular velocities equation (7)

ω_L,ω_F are the angular velocities of the pilot robot and the follower robot.

By substituting the equation of motion in (1), the controller is used to make $\lim_{t_o-t} \dot{x}_{eVF}$, $\lim_{t_o-t} \dot{y}_{eVF}$, $\lim_{t_o-t} \dot{\theta}_{eVF}$ closest to zero is sufficient. Due to the limited space of this paper does not design the controller using the Liyapunov function can prove convergence.

2.1.2. Route Planning and Design

Using the information collected by LiDAR for obstacle avoidance, this paper improves the DWA and A* algorithms and combines them to jointly guide the path planning.

(1) DWA Dynamic Window Approach

The core idea of the DWA algorithm is to determine a sampling velocity space in the velocity space (v,ω) that satisfies the hardware constraints of the mobile robot according to the current position state and velocity state of the mobile robot, and then predict the trajectory of the mobile robot in a certain period of time under these velocities, and evaluate the trajectory by an evaluation function, and finally select the velocity corresponding to the best evaluated trajectory as the Finally, the speed corresponding to the best evaluated trajectory is selected as the speed of the mobile robot, and so on until the mobile robot reaches the target point.

Trajectory scoring function: The trajectory scoring function is mainly used to calculate the score situation of multiple paths, using the cost score situation to select the best and most reasonable path, the ultimate goal of this paper is to hope to achieve the multi-intelligent body unknown space path planning and reasonable distribution of intelligent bodies, for the obstacle avoidance ability, anti-collision effect and obstacle avoidance of the smoothing degree of the requirements of the higher, the following is the evaluation function of this paper's improved DWA algorithm.

$$G = \partial \cdot g_{goal} + \beta \cdot g_{velocity} + \gamma \cdot g_{obstacle} + \delta \cdot g_{direction} \quad (8)$$

Eq. 8 where α, β, γ, δ are the weighting coefficients, and g_{goal}, $g_{velocity}$, $g_{obstacle}$, and $g_{direction}$ are the target score, smoothing score, obstacle score, and direction score, respectively.

1. The target score in Eq. 9 represents the distance between the robot's current state position and the target position, the system goal is to minimise this distance, the smaller the distance the higher the score, which is used to guide the robot towards the target position.

$$g_{goal}(v,\omega) = \alpha \cdot dist(p(v,\omega), goal) \quad (9)$$

$$\begin{cases} x(t+\Delta t) = \lambda(t) + v \cdot cos\big(\theta(t)\big) \cdot \Delta t \\ y(t+\Delta t) = y(t) + v \cdot sin\big(\theta(t)\big) \cdot \Delta t \\ \theta(t+\Delta t) = \theta(t) + \omega \cdot \Delta t \end{cases} \quad (10)$$

$$dist(p(v,\omega), goal) = \sqrt{(x(t+\Delta t) - x_{goal})^2 + (y(t+\Delta t) - y_{goal})^2} \quad (11)$$

$$g_{\text{velocity}}(v,\omega) = \beta \cdot (\Delta \dot{v}^2 + \Delta \dot{\omega}^2) \quad (12)$$

$$g_{\text{obstacle}}(v,\omega) = \gamma \cdot \left(\frac{1}{\text{min\_dist}(p(v,\omega), \text{obstacles}) + \varepsilon} \right) \quad (13)$$

dist(p(v,ω),goal) is the distance between the robot's predicted position trolley and the goal position goal at velocity v and angular velocity ω, which is calculated as shown in Equation 10:

Using Eq. 11 Euclidean distance calculation, the distance between the existing position of the trolley and the target position is obtained x_{goal},y_{goal} are the horizontal and vertical coordinates of the target position.

2. The smoothing score of Eq. 12 indicates the smoothness of the speed and angle changes during the movement of the intelligent body, and the system objective is to minimise the rate of change of speed and angle, and the smoother the change, the better, on the premise of being able to achieve the function.

$\Delta \dot{v}$ It's the rate of change of velocity.

$\Delta \dot{\omega}$ is the rate of change of angular velocity.

3. The obstacle score in Eq. 13 represents the distance score between the intelligent body and the obstacle, and the goal is to be able to keep the intelligent body and the obstacle within a certain distance interval as far as possible under the minimum requirement of obstacle avoidance, so that it will not be too far away from the obstacle, resulting in a reduction in the efficiency of obstacle avoidance, nor will it be close to the obstacle, resulting in the restricted field of view of the LiDAR exploration.

min_dist(p(v,ω),obstacles) is the minimum distance between the robot's predicted position and the nearest obstacle at velocity v and angular velocity ω; ε is a very small value used to avoid a zero denominator.

4. In the above formula (13) we can use a simple Euclidean distance calculation to make the cart as far away from the obstacle as possible to achieve the obstacle avoidance function, but in this paper to

achieve the function is not simply as far away from the obstacle as possible, the function achieved by the project in this paper also involves the distribution of multiple intelligences and the spatial monitoring, in order to achieve the best monitoring effect, it is not only need to be as far away from the obstacle as possible, need to take into account the close to the obstacle monitoring perspective will be narrowed too far away from the obstacle will be insufficient to the obstacle wrapping and other issues. In order to achieve the best monitoring effect, it is not only necessary for the robotic vehicle to be as far away from the obstacle as possible, but also need to take into account that the monitoring perspective will be narrowed when it is close to the obstacle, and it will be insufficiently wrapped around the obstacle when it is too far away from it. Therefore, this paper uses a new obstacle score formula, which is able to calculate the score by the distance from the obstacle in different intervals, and the score changes in different intervals are not the same.

$$g_{obstacles} = e^{-(\frac{min\_dist-\mu}{\lambda})^2} \tag{14}$$

The above formula is used to calculate the minimum distance min_dist from each point on the trajectory to the nearest obstacle, and then the exponential decay function is used to calculate the score, which decreases very quickly when min_dist is very small, and slower when min_dist increases.

μ is the proper distance desired to be maintained and is related to the number of dropping robots and the spatial complexity.

λ controls the rate of decay. A larger value of λ will make the score stay higher over a larger distance range, and adjusted according to the requirements for distance sensitivity, the larger λ is, the slower the score declines and the wider the scope of application; the smaller λ is, the faster the score declines and the narrower the scope of application. The use of this formula can make the robot trolley in a certain distance range, to achieve the obstacle avoidance function.

$$H(v,\omega) = \delta \cdot (-|angle\_diff(\theta_{goal}, \theta_{new})|) \tag{15}$$

$$\theta_{goal} = arctan(y_{goal} - y, x_{goal} - x)$$

$$\theta_{new} = \theta + \omega \cdot \Delta t$$

5. the direction score of formula 15 indicates the angle between the direction after avoiding obstacles and the direction of the target, the goal is to minimise this angle, the smaller the angle the higher the score, indicating that after avoiding obstacles can continue to move forward in the original direction. This ensures that the exploration process is efficient and does not cause repeated exploration by adjusting the direction multiple times in an unknown area.

θ_{goal} is the angle between the current direction of the robot and the direction towards the target position. θ_{new} is the angle between the robot's new direction at velocity v and angular velocity ω and the direction towards the target position. angle_diff($\theta_{goal,}\theta_{new}$) is the angle difference between θ_{goal} and θ_{new}.

Dynamic Window Constraints: In DWA, a dynamic window defines a set of available velocity commands that are defined by the following constraints.

Velocity constraints: The velocity range of the robot at the current moment is determined by the maximum velocity of the robot and the acceleration and deceleration constraints.

$$v_{\min} \le v \le v_{\max}$$

$$\omega_{\min} \le \omega \le \omega_{\max} \tag{16}$$

V_{min}, V_{max} is the minimum and maximum speed that can be achieved by the mechanical motion of the trolley, ω_{min},ω_{max} is the minimum and maximum angular velocity that can be achieved by the mechanical motion of the trolley, all are related to the specific model of the trolley and so on.

Dynamic constraints (17): in a small period of time after the current moment t (dynamic window), the velocity change is limited by the acceleration.

$$v_{\text{current}}\text{-}\dot{v}_b \cdot \Delta t \le v \le v_{\text{current}} + \dot{v}_a \cdot \Delta t$$

$$\omega_{\text{current}} - \dot{\omega}_b \cdot \Delta t \le \omega \le \omega_{\text{current}} + \dot{\omega}_a \cdot \Delta t \tag{17}$$

$$v \le \sqrt{2 \cdot dist(v,\omega) \cdot \dot{v}_b} \wedge \omega \le \sqrt{2 \cdot dist(v,\omega) \cdot \dot{\omega}_b} \tag{18}$$

$$(v^*, \omega^*) = arg \overset{max}{(v,\omega)} \in \text{DW}\, G(v,\omega) \tag{19}$$

In Eq. 17, V current, ω current are the velocity and angular velocity of the cart at the current moment, and $\dot{V}_b$ and $\dot{\omega}_b$ are the deceleration and deceleration angular velocity of the cart, and the purpose of this constraint is to ensure that the acceleration and deceleration ranges of the cart should be able to conform to the actual situation, and should not only satisfy the algorithmic requirements of the simulation.

Obstacle constraints (18): in the range of mechanical power of the machine trolley, the chosen trajectory can not collide with any obstacle, in a certain distance at a certain time to be able to stop or go around.

(dist(v,ω) in (18) is the distance from the obstacle calculated in the dynamic window.

Use (19) to select the optimal speed:

(v*,ω*) is the selected optimal speed command which is also the highest scoring speed command. arg max denotes finding the velocity command that maximises the objective function G(v,ω) (v,ω) is the set of velocity commands within the dynamic window.

The DW is a search space of velocity commands defined taking into account the current state of the robot and its kinematic capabilities (e.g., maximum acceleration and maximum velocity).

G(v,ω) is the evaluation function that assesses the merit of velocity commands based on several factors.

(2) Exploratory A* global planning algorithm combined with DWA dynamic window method

The combination of the A* algorithm and the DWA dynamic window method for autonomous navigation and obstacle avoidance in complex and unknown environments can significantly improve the path planning capability and real-time obstacle avoidance performance of mobile robots. The core idea of this combination strategy is to use the global planning advantage of the A* algorithm to solve the problem of the DWA algorithm falling into the local optimum due to the insufficient prediction length or short step size in the obstacle avoidance process, and to use the local dynamic adaptation of the DWA algorithm to solve the problem of collision with the moving target or the target that suddenly enters the robot due to the failure to consider the dynamic changes in the planning of global routes by the A* algorithm. The DWA algorithm is able to solve the problem of collision with moving targets or sudden intruders caused by the inability of the A* algorithm to take dynamic changes into account in the global route planning.

However, for this project, given that traditional A* algorithms rely on complete environmental map information, this project innovatively introduces an exploratory A* algorithm. The algorithm works in an unknown environment, acquires local information about the surrounding environment in real time through sensors such as LiDAR, and dynamically sets temporary target points within its scanning range. With the continuous exploration and advancement of the mobile robot, these temporary target points are constantly updated, guiding the robot to gradually approach the global target, while achieving effective exploration and monitoring of the unknown space.

This strategy not only overcomes the limitation that traditional A* algorithms cannot be directly applied in unknown environments, but also improves the robot's ability to explore autonomously in unknown environments.

1. Select a local target location P_{local} within the scanning range, and use the A* algorithm to plan the path from the current location $P_{current}$ to the target location P_{local}.

$$f(n) = g(n) + h(n) \tag{20}$$

f(n) is the total cost of node n. $g(n)$ is the actual cost from the starting point $P_{current}$ to node n. $h(n)$ is the heuristically estimated cost from node n to the local goal location P_{local}.

When the trolley reaches the local target position or when the target position is found to be unreachable during the process, the target position is updated using the traditional Euclidean distance as a heuristic function, but there is no need to perform calculations to obtain it during data collection using LiDAR, and the measured distance can be used directly.

$$P_{local} = arg \max_{P \in scan(P_{current}, R)} \|P - P_{current}\| \tag{21}$$

scan($P_{current}$,R) denotes the set of reachable points within the scanning range centred on the current position $\mathrm{P_{current}}$ with radius R. Using the above formula to update a new target position, this process is repeated to achieve the purpose of exploring the unknown space

2. Selection of local target points

$$p* = select_{local_{goal}} \cdot (P_{local}, current\_position) \tag{22}$$

P_{local} is the local goal position found in the A* algorithm, and p* is the local goal position found in the path of the A* algorithm in the DWA algorithm, both of which belong to the state of being dynamically updated all the time.

Figure **4** shows the multiple obstacle avoidance paths constructed by the robotic vehicle using DWA in the face of an obstacle.

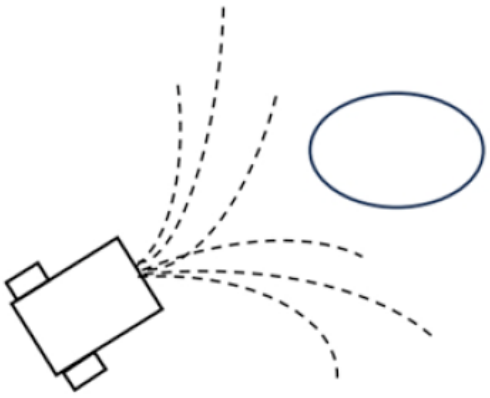


Figure 4: Schematic diagram of path selection.

2.2. Distributed Design

This section describes how multi-intelligentsia can adopt a distribution strategy by judging the environment when implementing exploratory path planning in an unknown space, and achieve continuous monitoring of the unknown environment space through the chained distribution designed in this paper.

Given the inherent randomness and complexity of unknown space exploration, especially for those closed environments that are difficult to reach or unsuitable for human exploration, traditional vehicle formations relying on a single signal transmission mode face a high risk of communication interruption and localisation loss. In order to effectively address this challenge, this project innovatively conceives a chained cooperative exploration architecture, which deeply integrates the advantages of physical connectivity and wireless communication technologies to ensure stable and reliable exploration and localisation capabilities even under extreme conditions.

Specifically, we design a chained exploration system consisting of multiple LiDAR-guided carts. These trolleys are connected to each other by means of physical-visual links, forming an invisible, flexible, and resilient exploration chain. This design ensures that each new vehicle is precisely placed within the direct monitoring range of its predecessor, which not only greatly reduces the risk of communication The (23),(24) constraints ensure that the former robot can be within the field of view of the latter robot, and when the former robot is about to leave the field of view of the latter robot, the former robot that is exploring and following will stop where it is and start monitoring the area. Interruption due to signal degradation or interference, but also enables real-time and precise tracking of each vehicle's position, thus avoiding any single node from being ‘lost’ in the exploration process.

In addition, the layout strategy of the chain structure gives the system seamless coverage for global monitoring. Through continuous and uninterrupted line-of-sight connections, the entire exploration chain can maintain all-round, dead-angle-free monitoring and positioning of the surrounding environment, which is crucial for precise navigation and path planning in complex and unknown spaces. This architecture not only improves the efficiency and safety of exploration missions, but also provides a high-quality, high-density spatial information base for subsequent data analysis and spatial modelling.

This function is mainly implemented as a judgement and constraint approach, whereby uninterrupted judgement is used to find situations that meet the constraints and execute the command.

1. Vision constraints

Constraints (23),(24) in which θ_{L_F} is the relative angle between the front robot and the back robot, θ_{view} is the angle of the field of view, and i is the robot number, starting from the last robot with 0, and so on.

The (23),(24) constraints ensure that the former robot can be within the field of view of the latter robot, and when the former robot is about to leave the field of view of the latter robot, the former robot that is exploring and following will stop where it is and start monitoring the area.

2. Following distance range

The drobot in (25) is the furthest distance that the machine trolley can detect, and the above equation ensures that the robot has to stay within the monitoring range without obstacles blocking its view, and once it is out of range, the previous machine trolley stops and starts monitoring the area.

$$\left|\theta_{L\_F}\right| < \theta_{view} \tag{23}$$

$$\theta_{L_F} = arctan(y_{i_{+1}} - y_i, x_{i+1} - x_i) - \theta_i \tag{24}$$

$$\sqrt{(x_{i_{+1}} - x_i)^2 + (y_{i+1} - y_i)^2} < d_{robot} \quad \mathrm{i} \in [0,\mathrm{n}] \tag{25}$$

3. Distribution strategy

In this project, we design and implement a highly collaborative robotic convoy system that explores deep into unknown space in the form of a serialised queue. The convoy is arranged in a strict one-line serpentine formation, led by a pilot vehicle, and enters the target area sequentially, ensuring order and safety during the travelling process.

When the robot car at the end of the queue reaches the preset conditions or makes a decision based on real-time data analysis, it automatically or manually triggers a stop command, and then switches to the monitoring mode. The execution of this key action marks the full launch of the circular judgement and constraint mechanism within the fleet. The mechanism aims to dynamically adjust the behavioural strategy of each trolley based on its specific position, environment sensing data and the exploration progress of the lead trolley.

Specifically, as the tail vehicle stops and switches to the monitoring state, each vehicle in front of it responds in turn, entering a round-robin judgement process driven by a predefined algorithm or on-the-fly decision logic. Once a vehicle meets a specific stopping condition, it stops travelling and switches to the monitoring state, continuing to provide the necessary

environmental information to support the overall exploration task.

This process is carried out recursively until every robot car in the fleet stops moving according to the established strategy and transforms into a monitoring unit, or the pilot car successfully completes the exploration of the whole unknown space. This design not only embodies a high level of automation and intelligence, but also significantly enhances the fleet's adaptability and flexibility in the face of complex environments, providing valuable practical experience and technical support for future large-scale, high-efficiency robotic exploration missions.

3. EXPERIMENTS AND SIMULATIONS

The content of this chapter is mainly to show the performance of all the models and algorithms mentioned above in the experiments, and at the same time the optimised and improved algorithms are compared with the original traditional algorithms, so as to better reflect the advantages and rationality of the new algorithms.

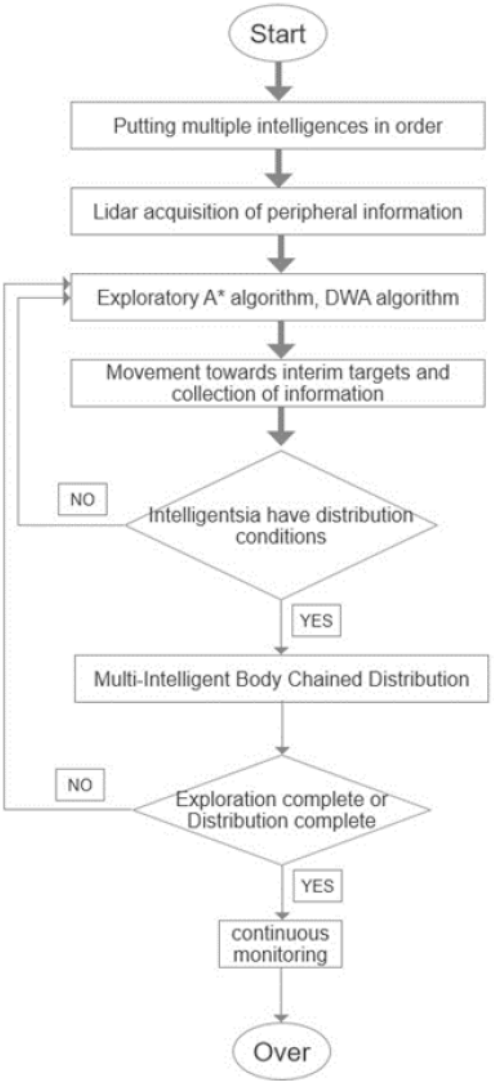


Figure 5: Method Overview Diagram.

3.1. Simulation Effect

3.1.1. Extended A Algorithm*

This paper extends the traditional A* algorithm and conducts a comparison experiment with the traditional algorithm with 25% obstacle coverage in the same 300*300 raster map. In this experiment, the traditional A* algorithm and the exploratory A* algorithm extended in this paper set the same starting point and end point, and give the same obstacles to compare the processing effect of the two algorithms. This paper also compares the exploratory A* algorithm with known endpoints and the A* algorithm with completely unknown endpoints. Figure **6** shows the results of the traditional A* algorithm, which clearly shows that the traditional A* algorithm can reasonably plan a shortest path through all kinds of known information. Figure **7** shows that under the condition of unknown endpoint information and obstacle information, the endpoint position is finally reached through exploratory path planning, and the insufficiency of information will lead to different paths to reach the target position each time. It can be seen that although there is a large amount of unknown information interference, the extended A* algorithm in this paper has already avoided the repeated exploration of paths as much as possible, reducing the waste of resources and lowering the efficiency.

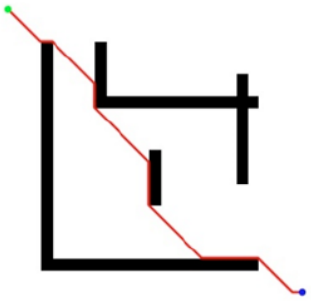


Figure 6: Traditional A* algorithm.

Figure **7** shows the path planning using the exploratory A* algorithm after the approximate direction of the end position is known. The simulation diagram is shown below:

The results of the specific analyses are presented below:

The traditional A* algorithm in Figure **6** requires the use of all the obstacle information, including the start and end positions, which is a one-time path planning and a single applicable scenario. In the expanded exploratory path planning in Figure **7**, due to the lack of information will make the beginning is not necessarily towards the end position to explore, but the subsequent due to the angle, the continuous improvement of the

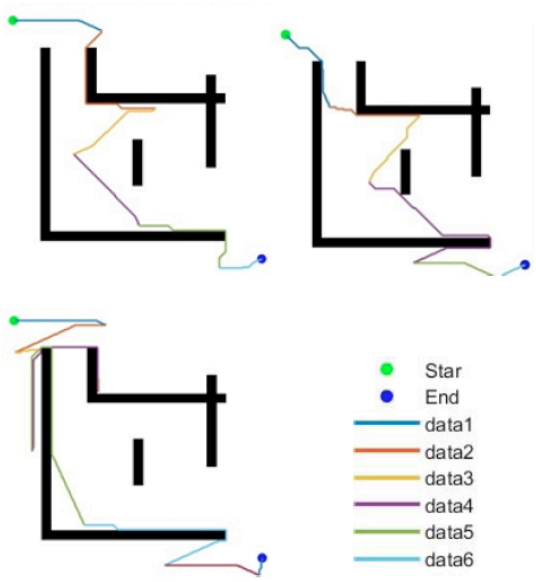


Figure 7: Exploratory A* Algorithm for Unknown Endpoint Information.

information, will continue to advance towards the end position to finally reach the goal position. Through the performance calculation comparison of traditional A* algorithm in reference [13] and the data in Table **1**, it can be seen that the response time of the expanded A* algorithm in this paper through the exploration in the case of unknown all the information is only increased by about 0.2 seconds per 100 metres on average, while the expanded algorithm in this paper realizes the A* exploration in the case of unknown information, so the increase in the response time belongs to the acceptable range.

3.1.2. Improved DWA algorithm and integration of exploratory A*

In this paper, the traditional DWA algorithm is improved and jointly guides the path planning by combining with the exploratory A* algorithm mentioned above, and the feasibility and reasonableness of the algorithm is verified by establishing the overall multi-intelligent body obstacle environment to simulate the above algorithm. The dynamic obstacle avoidance effect, path planning and the self-organised distribution

Table 1: Performance Parameters of Several A* Algorithms

Explore Segments	1	2	3	4	5	6	Total Duration
Search time(s)							
Legacy A*[14]	0.0882	0.0798	0.0774	0.0832	0.0798	0.0865	0.4949
Improvement A*(1)	0.0466	0.3895	0.0889	0.0200	0.0923	0.0139	0.6512
Improvement A*(2)	0.0439	0.0162	0.0529	0.4716	0.0601	0.0090	0.6537

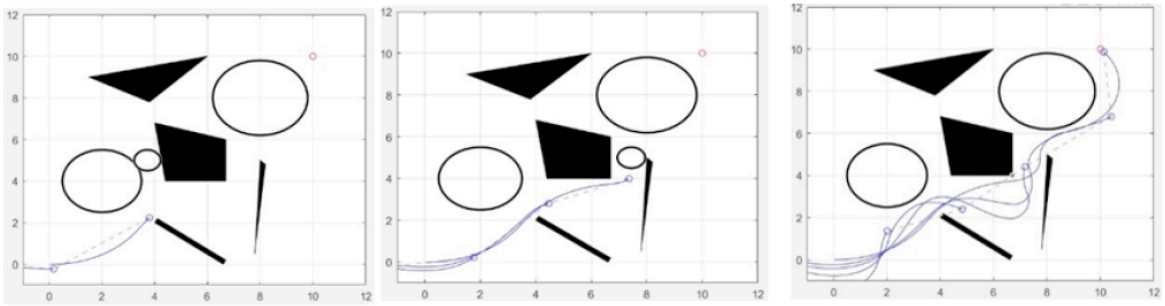


Figure 8: Dynamic Obstacle Avoidance and Distribution by Fusing Improved DWA Algorithm with A* Algorithm.

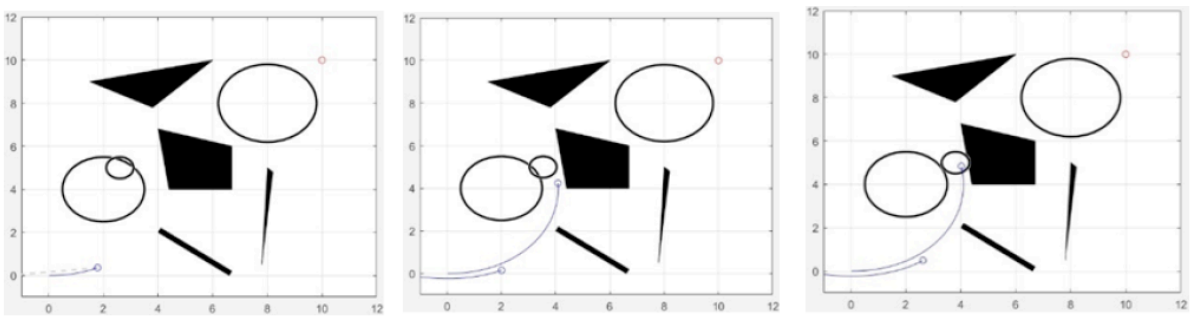
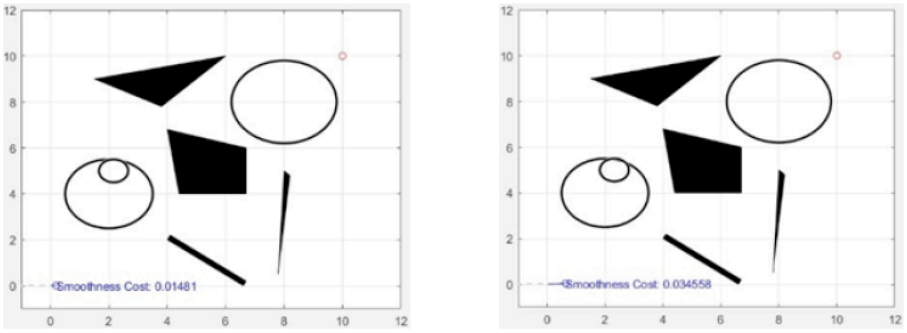


Figure 9: A* algorithm for obstacle avoidance and distribution.



(**A**) (**B**)

Figure 10: Smoothness cost.

of the subsequent intelligences of the multi-intelligentsia are demonstrated in Figure **8**.

In this paper, the same 150*150 raster map with 45% of obstacles layout is established, Figure **8** is the dynamic path planning which integrates the improved A* algorithm and DWA algorithm, and Figure **9** compares with the static path planning using only A* algorithm and smoothing in the literature [14], it can be clearly seen through the above picture that the multi-intelligence body in Figure **8** is under the leadership of the head vehicle in the left channel to avoid the dynamic obstacles, while in Figure **9** the head vehicle in channel one directly passes through the dynamic obstacles, and changes the path through the right channel to finally reach the target location, according to the comparison of the A* algorithm in the literature [14]. Dynamic obstacles, change the path through the right channel to finally reach the target location, while in Figure **9**, the head vehicle in the channel one directly through the dynamic obstacles, according to the comparison of the A* algorithm in the literature [14] straight through the dynamic obstacles, proving once again that the use of A* algorithms can not be made dynamic response, reflecting the importance and necessity of fusion algorithms. And compared with the traditional DWA algorithm in the literature [14] due to the traditional DWA algorithm does not take into account the problem of smoothness, so it leads to a high cost of smoothness, after comparing this paper's improved DWA algorithm in the smoothness index to improve about 50%, so that multi-intelligence can better bypass the obstacles, as shown in Figure **10**, the lower part of the Smoothness Cost is the cost of smoothness (A) for the improved DWA algorithm in this paper and (B) is the smoothness cost of traditional DWA algorithm in literature [14].

At the same time for the multi-intelligent body distribution problem mentioned in this paper, through

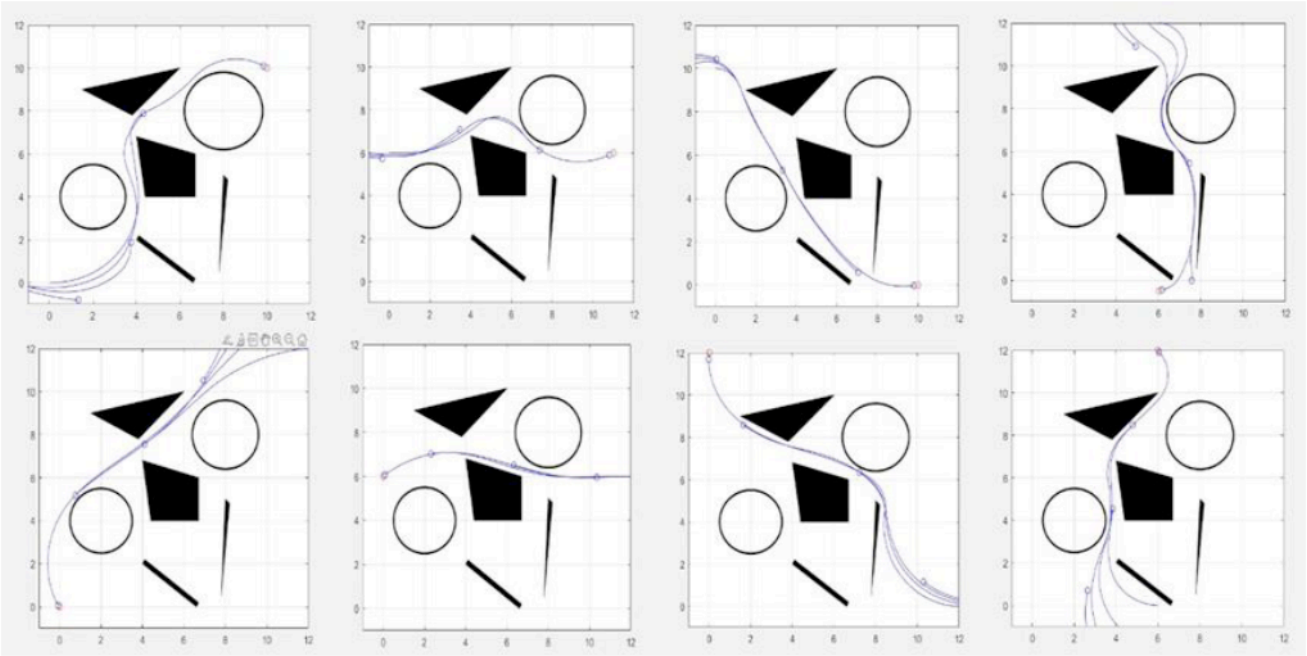


Figure 11: Results of repeated experiments in different directions.

different constraints make the multi-intelligent body between the chain distribution, as shown in Figure **8** grey dotted line connected, not only to achieve the physical sense of the visual connection has been to solve some complex environment signal interruption, but also through wireless communication to achieve large-scale data transmission, through a large number of different angles of the different starting point end point can also be a reasonable Distribution results Figure **11**.

CONCLUSION

In this paper, for complex indoor environments where GPS cannot be utilised, a multi-intelligent body based self-organised path planning and exploration scheme is designed, focusing on improving the A* and DWA algorithms and fusing the two to address the shortcomings of global and local planning. With this scheme, the limitations of traditional path planning methods in unknown environments are successfully addressed, and autonomous formation movement and area coverage monitoring of multiple intelligences are achieved.

After the fusion of the improved A* algorithm and DWA algorithm not only makes up for the functional defects of static path planning on the inability to avoid dynamic obstacles, but also improves the planning ability of dynamic obstacles on long paths, through simulation experiments and comparisons, fully proved the feasibility of the algorithm and advantages. Compared with the traditional A* algorithm, it not only achieves a large reduction of information demand in function, but also expands the application scenarios to satisfy most of the path planning in the case of lack of information, and at the same time ensures that the growth of the response time is not more than 30% to keep it in an acceptable and reasonable range, and also avoids the duplication of the exploration and redundancy of the paths through the constraints of the path exploratory lengths.

For the traditional DWA algorithm, the length of the planned path is short, and for the path planning of the whole director's path, there is always the dilemma of falling into the global optimal solution, thus failing to find a reasonable path, and the traditional DWA algorithm, due to its simple evaluation index, can only control the robot to complete the simplest obstacle avoidance and converge to the target position, and it does not take into account the degree of smoothness of the route, and the need to reserve space to prevent temporary dynamic obstacles from appearing in the visual dead zone when avoiding obstacles. space to prevent temporary dynamic obstacles from appearing in the visual dead zone. The improved DWA algorithm in this paper not only constructs a new set of cost evaluation indexes, but also adjusts the motion and obstacle avoidance of multi-intelligent body trolley clusters through the constraint system, so that the intelligent bodies can take into account the overall monitoring effect in the smoother obstacle avoidance and path planning process without causing too large monitoring dead space or missing obstacles, and the smoothing cost score increases by 0.5 percentage points in comparison with that of the traditional DWA algorithm. comparison, the smoothing cost score rises by about 50%.

ACKNOWLEDGMENTS

This work was supported by the Open Project of Key Field Alliance of Liaoning Province under Grant No. 2022-KF-11-03, the Educational Department Foundation of Liaoning Province under Grant No. JYTMS20231435, the National Natural Science Foundation of China under Grant No. 62073158.

DECLARATION OF COMPETING INTEREST

The authors declare no conflict of interest.

REFERENCE

[1] Gershon N, Eick SG, Card S. Information visualization. Interact. 1998; 5(2): 9-15. https://doi.org/10.1145/274430.274432

[2] Tang J, Chen Y, Jaakkola A, Liu J, Hyyppä J, Hyyppä H. NAVIS-An UGV indoor positioning system using laser scan matching for large-area real-time applications. Sensors. 2014 Jul 4; 14(7): 11805-24. https://doi.org/10.3390/s140711805

[3] Menegatti E, Behnke S, Zhou C. Humanoid soccer robots. Robotics and Autonomous Systems. 2009 Jul 31; 57(8): 759-60. https://doi.org/10.1016/j.robot.2009.03.003

[4] Noreen I, Khan A, Habib Z. Optimal path planning using RRT* based approaches: a survey and future directions. International Journal of Advanced Computer Science and Applications. 2016; 7(11). https://doi.org/10.14569/IJACSA.2016.071114

[5] Li Q, Xu Y, Bu S, Yang J. Smart vehicle path planning based on modified PRM algorithm. Sensors. 2022 Aug 31; 22(17): 6581. https://doi.org/10.3390/s22176581

[6] Seet BC, Liu G, Lee BS, *et al*. A-STAR: a mobile ad hoc routing strategy for metropolis vehicular communications. In: Networking 2004: Technologies, Services, Protocols; Performance of Computer and Communication Networks; Mobile and Wireless Communications. 3rd Int IFIP-TC6 Conf; 2004; May 9-14; Athens, Greece.

[7] Wang N, Dai J, Ying J. UAV formation obstacle avoidance control algorithm based on improved artificial potential field and consensus. International Journal of Aeronautical and Space Sciences. 2021 Dec; 22(6): 1413-27. https://doi.org/10.1007/s42405-021-00407-6

[8] Ogren P, Leonard NE. A convergent dynamic window approach to obstacle avoidance [J]. IEEE Trans Robot, 2005; 21(2): 188-195. https://doi.org/10.1109/TRO.2004.838008

[9] Rösmann C, Hoffmann F, Bertram T. Timed-elastic-bands for time-optimal point-to-point nonlinear model predictive

control[C]//2015 Eur Control Conf. IEEE, 2015: 3352-3357. https://doi.org/10.1109/ECC.2015.7331052

[10] XU J, XIN S J, Deng YZ. Mobile robot path planning based on the fusion of improved A and TEB algorithm [J]. Metrol Meas Tech, 2022; 49(05): 26-30.*

[11] Zhang T, Chen Z, LI Y M, *et al*. Research on robot obstacle avoidance based on improved A algorithm and dynamic window method [J]. Instrum Tech Sens, 2023; (04): 102-106.

[12] Xu W, Tang J W, Zhang C. Path planning of mobile robot based on improved A∗ and dynamic window algorithm [J]. Computer Simulation, 2023; 40(03): 447-452.

[13] Liao G, Wu Y. Improved robot path planning with A and DWA fusion[C]//2023 7th Int Conf Autom Control Robot. Kuala Lumpur, Malaysia: IEEE; 2023: 14-19. https://doi.org/10.1109/ECC.2015.7331052

[14] Qian W, Peng J, Zhang H. Research on Mobile Robot Path Planning Based on Improved A* and DWA Algorithms. InInternational Conference on Computer Engineering and Networks 2023 Nov 3 (pp. 105-118). Singapore: Springer Nature Singapore. https://doi.org/10.1007/978-981-99-9239-3_10

Received on 15-08-2024 Accepted on 23-09-2024 Published on 01-10-2024

https://doi.org/10.31875/2409-9694.2024.11.04