Hindawi
Journal of Electrical and Computer Engineering
Volume 2022, Article ID 5739765, 11 pages
https://doi.org/10.1155/2022/5739765

Research Article

Motion Route Planning and Obstacle Avoidance Method for Mobile Robot Based on Deep Learning

Jichao Cui and Guanghua Nie

Institute of Intelligent Engineering, Henan Institute of Technology, Xinxiang 453003, China

Correspondence should be addressed to Jichao Cui; jichaocui@hait.edu.cn

Received 11 May 2022; Accepted 17 June 2022; Published 19 November 2022

Academic Editor: Xuefeng Shao

In order to improve the motion route planning effect and obstacle avoidance effect of mobile robots, this paper combines the deep learning theory to analyze the motion route planning and obstacle avoidance process of mobile robots. According to the obstacle avoidance trajectory and constraints, this paper establishes a safe distance model for obstacle avoidance, then analyzes the braking process of the robot, and designs an improved safety model for obstacle avoidance. This model integrates two relatively mature safety models, complements their advantages and disadvantages, and comprehensively considers robot safety and the utilization of the motion path. According to the simulation test research, the robot based on deep learning proposed in this paper has a good motion route planning effect and obstacle avoidance effect and can effectively improve the autonomous motion effect of the robot.

1. Introduction

At present, mobile robots are widely used in industry, aviation, and other fields. Among them, the differential drive robot can perform the continuous and real-time autonomous movement on roads and in the wild and is an intelligent machine system that integrates environmental perception, dynamic decision-making, and execution. In the work of robots, it is inevitable to encounter different obstacle environments. In order to successfully complete the work content, the research on obstacle avoidance route planning is particularly important. Obstacle avoidance route planning is to search for a safe path from the initial point to the target point for the robot in an environment with obstacles. According to the different understanding of the environmental data, the corresponding algorithm can be used to find the most suitable path so as to quickly pass through all the obstacles ahead and successfully complete the movement task.

In the path planning method of the robot, the traditional methods include the artificial potential field method, Dijkstra algorithm, and A* algorithm. As the working environment of mobile robots becomes diverse, some traditional methods cannot meet their operating requirements in complex environments, and the emergence of intelligent algorithms overcomes the shortcomings of traditional methods [1]. At present, the path planning of most mobile robots adopts intelligent algorithms, such as the ant colony algorithm, particle swarm algorithm, and artificial fish swarm algorithm. Among them, the ant colony algorithm is widely used due to its advantages of fast convergence, good performance, and robustness of the algorithm. The traditional ant colony algorithm has a long search time and slow convergence speed and is prone to the problem of algorithm stagnation. Therefore, many researchers have proposed improved methods such as the elite ant system [2], maximum-minimum (MAX-MIN) ant colony system, and adaptive ant colony algorithm. Literature [3] established a local pheromone diffusion model in the algorithm, improved the heuristic function and pheromone volatilization factor, and improved the convergence speed of the algorithm. Literature [4] uses Manhattan distance instead of Euclidean distance as the distance calculation method, which introduces the direction angle heuristic factor and increases the angle between the direction vector of the current node and the next node and the direction vector of

the next node and the endpoint. The probability of the path improves the efficiency of the algorithm. Literature [5] combines the estimation function of the A* algorithm into the heuristic factor of the ant colony algorithm and improves the pheromone volatility factor, which makes the ant colony algorithm improve both the convergence and the optimization path. Literature [6] adds the distance between the current node and the next node in the heuristic function and uses the La-place distribution to improve the pheromone volatility factor, which prevents the algorithm from falling into the local optimum and accelerates the convergence. Literature [7] proposes to add an obstacle avoidance strategy in the early stage of the algorithm, use the principle of high-quality ant update to guide the update of pheromone, and perform secondary planning on the obtained path to make the obtained path better and improve the operating efficiency of the robot. Aiming at the shortcomings of the traditional ant colony algorithm, which is easy to fall into local optimum and slow to converge, the traditional ant colony algorithm is improved. First, an obstacle avoidance factor is added to the state transition probability formula to reduce the number of deadlocks in the process of ants searching for a path. The Manhattan distance is used instead of the Euclidean distance to calculate the distance between nodes to reduce the approximation caused by the square root. Error effects speed up the calculation. Then, the pheromone volatility factor is improved from a fixed value to a dynamic value that changes with the number of iterations, and the range of the pheromone is limited to avoid algorithm stagnation. After comparing the simulation results, the improved ant colony algorithm converges faster than the traditional ant colony algorithm, and it can also perform efficient and reasonable path planning in complex maps [8].

Active obstacle avoidance technology is the basis of unmanned driving technology, and path planning and tracking control are important components of active obstacle avoidance technology. In literature [9], obstacle avoidance path planning is based on polynomials, and a robust controller is designed for trajectory tracking control in the literature [10]. In literature [11], the tentacle algorithm is used to design the obstacle avoidance strategy, and the improved artificial potential field method is proposed in literature [12] to plan the obstacle avoidance path and verify the feasibility of the obstacle avoidance algorithm in different scenarios through hardware-in-the-loop experiments. Among the obstacle avoidance planning algorithms, polynomial programming has the characteristics of a simple algorithm, small computational complexity, and strong versatility and practicability. Path-following control algorithms mainly include optimal control, sliding mode control, robust control, fuzzy control, and model predictive control [13]. Model predictive control is an algorithm for rolling optimization control based on a simple model, which is widely used in the field of robot control.

The traditional obstacle avoidance path planning methods for multimobile robots mainly include the particle swarm optimization method, error feedback correction method, and end position parameter adjustment method. Literature [14] proposed robot obstacle avoidance and dynamic target tracking in unknown environments. Adding velocity difference and acceleration difference to the gravitational formula improves the flexibility of the robot's dynamic target tracking and its adaptability to the environment, but it takes a long time. Literature [15] proposed an exploration of obstacle avoidance algorithm for mobile robots based on mixed strategies by using the vector field histogram method combined with neural network algorithm to achieve basic obstacle avoidance, but this method requires a large number of sample training sets, and the adaptive ability is not strong.

As a typical implementation method of trajectory planning, Dijkstra's algorithm [16] is based on a breadth-first search strategy and generates the shortest path according to the increasing order of path length. In order to achieve algorithm acceleration, A* algorithm introduces heuristic value from node to target point in Dijkstra's cost estimation [17]. Since the environment in which the autonomous driving robot is located contains some unknown change information, the D* algorithm will use the unaffected node information for replanning so as to adapt to the trajectory planning of the dynamic environment [18]. Since the planning space is expressed in the form of a discrete grid, the planning result is a polyline segment with discontinuous curvature. Considering the constraints of the robot's steering mechanism, the results of the above algorithm are not suitable for direct execution by the robot. Therefore, by improving the expansion of nodes, the hybrid A* derived from the method and the state lattice derived from the improved node connection method are the latest development direction based on the search algorithm. The planning result is a group of arcs and straight line segments, the curvature of the planned path is still discontinuous, and the results based on the search method need further smoothing before the robot can execute [19]. Different from the search method, the sampling-based method represented by RRT (rapidly-exploring random tree) is to expand the trajectory nodes in a continuous feature space. In order to improve the real-time performance of the algorithm, Bi-RRT (bidirectional RRT), which is bidirectionally extended from the starting point and end point, and hRRT (heuristic RRT), which introduces a heuristic function to increase the probability of low-cost nodes being extended, are derived. However, the sampling-based algorithm always contains a random seed generator, so it inevitably contains tortuous nodes [20], making the planning result to need further smoothing.

The existing local dynamic programming methods include the artificial potential field method, genetic algorithm, fuzzy logic algorithm, ant colony algorithm, and bacteria algorithm. Although the local planning algorithm can plan the path in real-time according to the detected environmental information, it is easy to lose the target point due to the lack of global information, resulting in the phenomenon of target unreachability and oscillation.

This paper combines the deep learning theory to analyze the process of mobile robot motion route planning and obstacle avoidance and builds an intelligent model through simulation research to improve the robot motion planning and obstacle avoidance effect.

2. Research Con Safe Distance Model and Obstacle Avoidance Strategy

2.1. Commonly Used Local Route Planning Methods. When the robot changes its motion path at high speed, the trajectory it travels needs to be stable and gentle, satisfy the conditions of lateral acceleration constraints, and yaw angular velocity so as to improve the safety and stability of the robot for lateral obstacle avoidance. The planned path plays the role of navigation in the process of the robot's lateral obstacle avoidance, guiding the robot to transition from the current position to the target position. In the research of this paper, the starting point and the endpoint of the whole process of changing the motion path of the robot will be set according to the requirements based on the known robot dynamics model and motion conditions. Then, according to the proposed constraints, a collision-free, safe, and comfortable driving path is planned for the robot to meet the requirements of the robot for lateral obstacle avoidance. Several commonly used local route planning methods will be introduced below.

The lane-changing method based on the sine function model is a widely used route planning method. Its lane-changing trajectory is based on a sine function image, and its parameters are adjusted according to the actual lateral obstacle avoidance situation, as shown in Figure 1. The mathematical expression for this method is as follows:

$$y(x) = \frac{sy_t}{x_t} - \frac{y_t}{2\pi}\sin\left(\frac{2\pi}{x_t}\right). \tag{1}$$

This paper improves the problem that the endpoint curvature does not meet the conditions in the route planning method based on the sine function model. Because the B-spline curve has good properties such as geometric invariance, flexibility, and convexity, a method is proposed to express the desired trajectory of lateral obstacle avoidance with its optimized sine and cosine functions. Moreover, the polygons are controlled in the form of curve fitting to make the driving process of the robot more stable. In this method, a 3rd-order spline can be selected to simulate the planned path, and the splicing of segments is used to achieve the purpose of transitioning from the current position to the target position. The image of the route planning method of the B-spline curve is shown in Figure 2, and its mathematical expression is as follows:

$$C(u) = \sum_{i=0}^{n} d_i N_{i,k}(u). \tag{2}$$

In the formula, d_i is the control point, and $N_{i,k}(u)$ is the basic function of the B-spline curve.

However, this kind of route planning method has high requirements for the selection of control points. The selection of control points directly affects the function of the planning curve, and the whole calculation process is complicated, and it also requires more information and data and a larger theory to support it.

The route planning method based on trapezoidal acceleration mainly controls the variation range of the lateral acceleration of the robot in the process of changing lanes within the range of constraints. Its lateral acceleration curve with time is two positive and inverse trapezoids with exactly the same size and shape, and its image is shown in Figure 3.

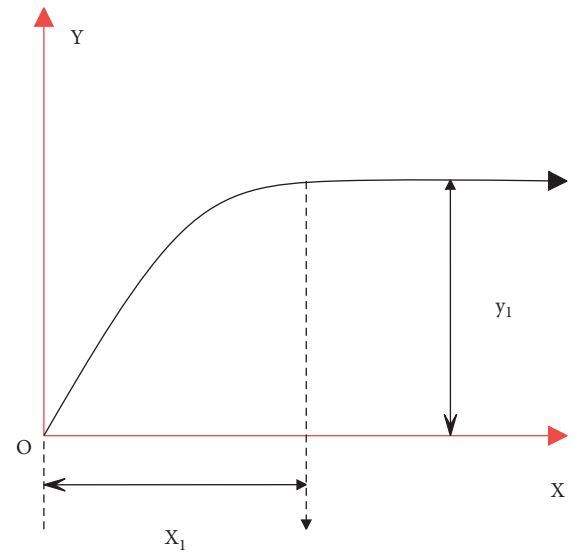

Figure 1: Image of the route planning method based on the sine function model.

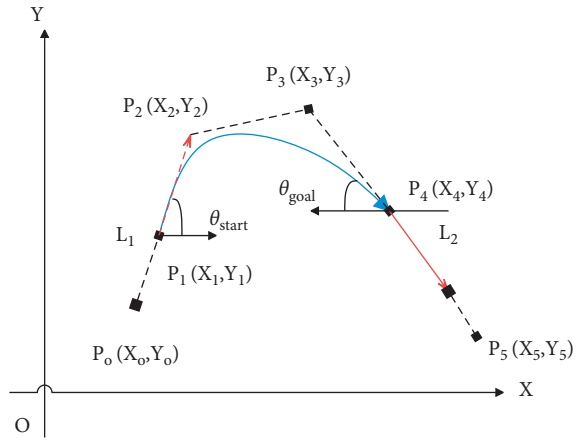

Figure 2: Image of route planning method based on B-spline curve model.

2.2. Lateral Obstacle Avoidance Trajectory and Safety Distance Model. When the robot travels on a multimotion path, it will choose different lane-changing methods according to the surrounding dynamic robot information. During the lane-changing process, the possible collision types mainly include robot rear-end collision, robot corner collision, and robot side scraping. Based on these three collision methods, different types of minimum safe distance models are established. We assume that the self-robot is M, s and F_0 are the front and rear robots of the current motion path, respectively, and L_d and F_d are the front and rear robots of the target motion path, respectively (as shown in Figure 4).

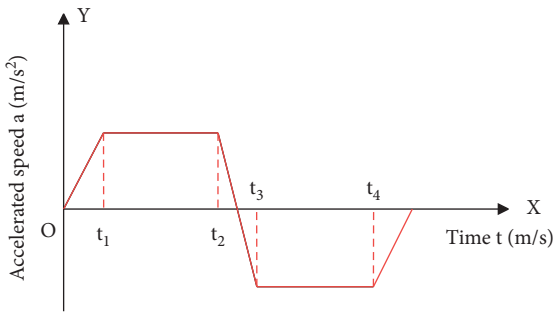

Figure 3: Image of the route planning method based on trapezoidal acceleration.

In this sports environment, there are four main types of safety distances that need to be considered:

(1) During the lane change process, the M robot may encounter corner collision, rear-end collision, and side scraping with the L_0 robot.

(2) In the process of changing lanes, the M robot may collide and rear-end with the F_0 robot.

(3) During the lane change process, the M robot may have rear-end collision, corner collision, and side scraping with the L_d robot.

(4) During the lane change process, the M robot may have a corner collision and side scraping with the F_d robot.

In the research process of this subject, in order to facilitate the calculation of the minimum safe distance model, the motion environment of the robot's lane-changing process has been simplified. This paper makes 4 ideal assumptions for the robot's lateral obstacle avoidance process:

(1) Since the shape and volume of the robot and the robot wheel have no effect on the research results during the lane-changing process, the robot and the robot wheel are treated as rigid bodies

(2) We assume that the former robot has been driving in a straight line on the current motion path, and the lateral acceleration and lateral displacement are always 0

(3) The robot independently selects and changes the motion path, and there are no interfering obstacles behind the current motion path and the front and rear of the target motion path

(4) During the process of lateral obstacle avoidance of the robot, the friction force of the road surface is large enough, and the robot rotates in pure rolling motion without sliding

The local route planning trajectory used in this paper is a quintic polynomial route planning algorithm, which is characterized by using a relatively simple calculation process and a smooth and gentle planned route, and can achieve the purpose of the robot's lateral obstacle avoidance and meet

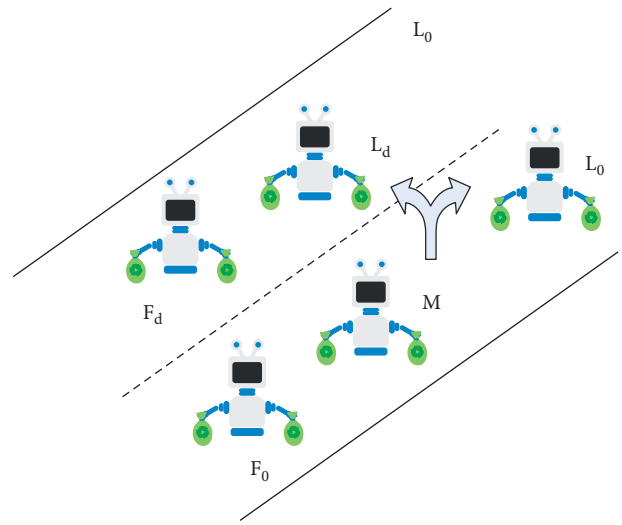

Figure 4: Schematic diagram of the surrounding robots at the time of lane change.

the constraints of robot dynamics and environment. The general form of a quintic polynomial is as follows:

$$y_{\text{ref}} = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + a_4 x^4 + a_5 x^5. \tag{3}$$

In the formula, x is the longitudinal coordinate of the robot in the process of lateral obstacle avoidance, y_{ref} is the lateral coordinate of the robot in the desired trajectory, and a_0`a_1, a_2, a_3, a_4, and a_5 are the five coefficients of the quintic polynomial, respectively.

The position state of the robot at the beginning of the lateral obstacle avoidance control is set as (x_1, y_1), and the position state of the robot at the end of the lateral obstacle avoidance control is set as (x_2, y_2).

From the boundary conditions, we get $y_1 = 0, \dot{y}_1 = 0, \ddot{y}_2 = 0; y_2 = y_e, \dot{y}_2 = 0, \ddot{y}_2 = 0$.

When it is substituted into formula (3), the solution is

$$\begin{aligned}
a_0 &= 0,\\
a_1 &= 0,\\
a_2 &= 0,\\
a_3 &= 10\frac{y_2 - y_1}{(x_2 - x_1)^3} - \frac{6\dot{y}_1 + 4\dot{y}_2}{(x_2 - x_1)^2} - \frac{3y_1 - \ddot{y}_2}{2(x_2 - x_1)},\\
a_4 &= 15\frac{y_2 - y_1}{(x_2 - x_1)^4} - \frac{8\dot{y}_1 + 7\dot{y}_2}{(x_2 - x_1)^3} - \frac{1.5y_1\ddot{y}_2}{(x_2 - x_1)^2},\\
a_5 &= 10\frac{y_2 - y_1}{(x_2 - x_1)^5} - 3\frac{\dot{y}_1 + \dot{y}_2}{(x_2 - x_1)^4} - \frac{\ddot{y}_1 - \ddot{y}_2}{2(x_2 - x_1)^3}.
\end{aligned} \tag{4}$$

We assume that the maximum lateral displacement value of the designed planning trajectory is s, that is, $s = y_2 - y_1$. Moreover, we assume that the distance between the robot and the previous robot at the initial moment of executing the

obstacle avoidance control is d, that is, $d = x\{2\}\text{-}x\{1\}$. Therefore, $c_3 = 10s/d^3, c_4 = 15s/d^4, c_5 = 6s/d^5$.

When it is substituted into formula (5), the reference path expression for lateral obstacle avoidance is obtained as

$$y = 10\frac{s}{d^3}x^3 + 15\frac{s}{d^4}x^4 + 6\frac{s}{d^5}x^5. \tag{5}$$

The lane-changing trajectory generated by using the quintic polynomial route planning algorithm and the constraints that the lateral obstacle avoidance path should satisfy are reliable and safe. Moreover, it satisfies various constraints of the path and robot dynamics, is suitable for complex road conditions, and can be directly generated by the set boundary conditions.

During the entire lateral obstacle avoidance process of the robot, the critical condition for successful obstacle avoidance between the robot and the preceding robot is that the lateral displacement of the self-robot is greater than the width h of the front robot when the head of the self-robot moves to the tail position of the front robot.

The path expression by using quintic polynomial planning is as follows:

$$y = 10\frac{s}{d^3}x^3 + 15\frac{s}{d^4}x^4 + 6\frac{s}{d^5}x^5. \tag{6}$$

The above formula is rewritten into the desired lane-changing path function expression with time as the parameter variable:

$$y(t) = \frac{s}{t_f^5}\left(10t_f^5 + 15t_f t^4 + 6t^5\right). \tag{7}$$

Among them, s is the lateral displacement of the robot to complete the entire lane-changing process, and t_f is the time taken for the entire lane-changing process. Therefore, the critical conditions for the robot to achieve lateral obstacle avoidance are as follows:

$$h = \frac{s}{t_f^5}\left(10t_f^5 + 15t_f t^4 + 6t^5\right). \tag{8}$$

In terms of the longitudinal displacement during the lane-changing process, the obstacle avoidance conditions that the self-robot and the front robot need to meet are as follows:

$$x_m(t) \geq x_l(t) + I + h\sin(\theta(t)), t \in \left[0, t_g\right]. \tag{9}$$

In the formula, x_l is the longitudinal displacement of the front robot, x_m is the longitudinal displacement of the self-robot, h is the width of the front robot, I is the length dimension of the robot, and t_g is the time from the robot head to the front robot tail.

The above formula is organized into the following:

$$x_t(t) = x_m(t) - x_l(t) - I - h\sin(\theta(t)), t \in \left[0, t_g\right]. \tag{10}$$

The critical condition of the minimum safe distance model of the robot's lateral obstacle avoidance is that the head of the self-robot is connected with the tail of the preceding robot. If it is assumed that the safe distance without collision when the two robots change lanes is b, we can get the following:

$$b = \min \int_0^t \int_0^\sigma (a_l(t) - a_m(t))\mathrm{dtd}\sigma + (v_l(0) - v_m(0)t), t \in \left[0, t_g\right]. \tag{11}$$

In the formula, a_l is the acceleration of the front robot, and a_m is the acceleration of the self-robot. When the robot performs lateral obstacle avoidance at high speed, the heading angle of the robot is usually relatively small, and Bakhsh et al. believe that the heading angle is usually less than 5° in this process. Therefore, in the process of changing lanes of the robot, it is considered that the longitudinal speed of the robot remains unchanged, which is equal to the original robot speed.

Considering the length of the robot and the static safety distance s_0 during the robot's driving process, according to the above formula, the expression for the minimum longitudinal safety distance is as follows:

$$b = (v_m - v_l)t_g + h\cos\theta + s_0. \tag{12}$$

Among them, the static safety distance s_0 is 2 m.

2.3. Establishment of Longitudinal Obstacle Avoidance Safety Distance Model. The specific braking process of the robot's longitudinal obstacle avoidance is as follows: First, after the driver finds the obstacle ahead, he realizes that the obstacle avoidance operation should be performed. Then, the driver moves his foot from any position to the brake pedal area, depresses the brake pedal, and keeps it until the final speed of the robot is 0. The braking process is represented in Figure 5 and includes the following stages:

(1) The stage of discovery of danger (t_1'): it refers to the time required for the driver to realize the obstacle avoidance operation and stop the acceleration operation after discovering the obstacle ahead

(2) Leg movement stage (t_1''): it refers to the movement time required by the driver to move the foot from any position to the brake pedal area

(3) Braking gap elimination stage (t_2): when the brake pedal is stepped on, the actual robot itself does not start braking because there is a gap in each component of the braking system; it takes a certain amount of time to cancel at this time

(4) Growth stage of deceleration (t_3): it refers to the duration that the deceleration of the robot increases from 0 to the maximum deceleration

(5) Hold stage of maximum deceleration (t_4): at this time, the braking force and deceleration of the robot reach the peak value, and the robot maintains the maximum braking deceleration until the robot stops completely

(6) The release stage g of the braking force (t_5): it refers to the time required for the driver to release the brake pedal until the brake pressure drops to zero after the driver successfully completes the obstacle avoidance operation

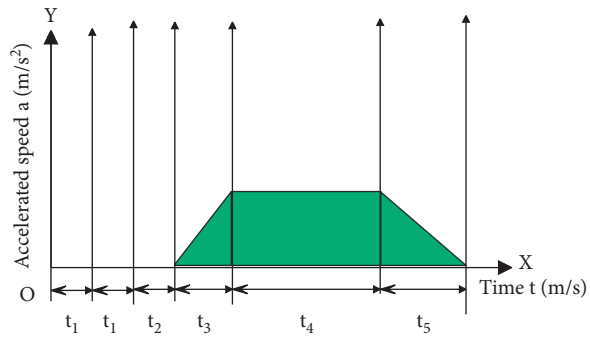

Figure 5: Variation of deceleration during braking.

The following is an analysis of the driving distance of the robot in each braking phase:

We assume that the initial speed of the robot is v_0 (km/h), and the maximum deceleration is a_1. During the driver's reaction time, foot-moving time, and braking coordination time period, the braking force is zero, and the robot runs at a constant speed, so

$$s_1 = v_0(t_1 + t_2). \tag{13}$$

In the braking force growth stage, the braking deceleration increases linearly, the robot performs variable deceleration motion, and the deceleration reaches the maximum value a_1 when the robot finally reaches a stable state, so the acceleration is as follows:

$$a = \frac{dv}{dt} = kt = \frac{a_1}{t_3}t. \tag{14}$$

The initial speed is v_0, and the speed of the robot at time t is $v = v_0 - \int a_1/t_3 t dt = v_0 - a_1/2t_3 t^2$, so

$$s_2 = \int_0^{t_3} v\mathrm{d}t = v_0 t_3 - \frac{a_1}{6}t_3^2. \tag{15}$$

The velocity at time t_3 is $v_3 = v_0 - a_1/2t_3$.

The deceleration in the braking force holding stage is the maximum value, and the robot performs a uniform deceleration movement, so

$$s_3 = \frac{v_3^2}{2a_1} = \frac{v_0^2}{2a_1} - \frac{v_0 t_3}{2} + \frac{a_1 t_3^2}{8}. \tag{16}$$

In summary, the total braking distance during the braking process is as follows:

$$s = s_1 + s_2 + s_3 = \left(t_1 + t_2 + \frac{t_3}{2}\right)v_0 + \frac{v_0^2}{2a_1} - \frac{a_1}{24}t_3^2. \tag{17}$$

When ignoring higher-order small quantities, we have the following:

$$s = \left(t_1 + t_2 + \frac{t_3}{2}\right)v_0 + \frac{v_0^2}{2a_1} = (T_1 + T_2)v_0 + \frac{v_0^2}{2a_1}. \tag{18}$$

In the above formula, $T_1 = t_1 = t_1' + t_1''$ is the danger detection stage and leg movement stage, which are combined into the driver operation delay stage, and $T_2 = t_2 + t_3/2$ is the braking gap elimination stage and the deceleration growth stage, which are combined into the braking delay stage.

From the aforementioned analysis of the braking process of the robot and Figure 6, it can be obtained that the critical warning distance of the robot is set as follows:

$$d_w = (T_1 + T_2)v_0 + \frac{v_0^2}{2a_1} + d_0. \tag{19}$$

Therefore, the minimum safe distance for braking is the difference between the critical warning distance and the distance traveled by the driver's operation delay stage, namely,

$$d_b = T_2 v_0 + \frac{v_0^2}{2a_1} + d_0. \tag{20}$$

Among them, d_0 is the minimum robot distance to be maintained after both robots stop.

When the actual distance between the two robots reaches d_w, the system judges that the robot has entered a potentially dangerous state and sends a warning signal to the driver to remind the driver that a dangerous situation is about to occur and corresponding measures need to be taken. When the distance between the two robots keeps approaching until it reaches d_b, the self-robot still does not brake, the robot enters an emergency state, and the system automatically controls the robot to brake to prevent the two robots from colliding.

The distance of the robot can maintain a small linear relationship with the driving speed of the self-robot. The safety distance model established by this characteristic is as follows:

$$s_w = v_r T_h + s_0. \tag{21}$$

Among them, S_w is the safety distance, T_h is the time distance of the robot head, V_r is the speed of the following robot, and S_0 is the distance between the two robots when the robot stops.

Although the algorithm of the safe distance between the robot head is relatively simple, it still has certain drawbacks. The focus of this algorithm is biased towards road usage. When the robot is running, the speed difference between the front and rear robots is large, and the safety distance model established by this algorithm is more dangerous, and the distance between the two robots will be too close to ensure the safety of movement.

Through the classification of the robot's travel speed in the motion path, the new safety distance expression is obtained as follows:

$$d_s = \begin{cases} T_2 v_0 + \dfrac{v_r^2}{2a_1} + d_0, & \dfrac{v_r > 60\mathrm{km}}{h}, \\ v_r T_h + s_0, & \dfrac{v_r \le 60\mathrm{km}}{h}. \end{cases} \tag{22}$$

The critical warning distance is expressed as follows:

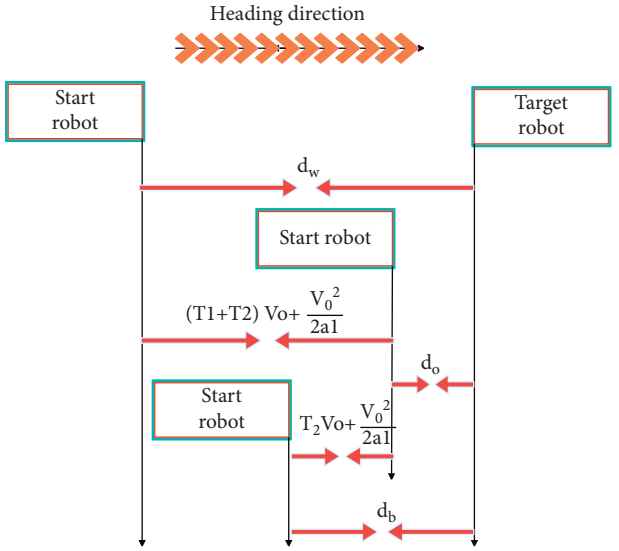

Figure 6: Schematic diagram of the distance between two robots.

$$d_w = \begin{cases} (T_1 + T_2)v_0 + \dfrac{v_r^2}{2a_1} + d_0, & \dfrac{v_r > 60\text{km}}{h}, \\ v_r T_h + s_0 + 25, & \dfrac{v_r \le 60\text{km}}{h}. \end{cases} \quad (23)$$

In the formula, d_s is the safe robot distance and v_r is the robot speed. Under low-speed conditions, the 25 m before reaching the critical braking distance is regarded as the critical warning distance, and T_1 is the driver's response delay time, which is generally 0.3–1.2 s, and 1.0 s is taken in this paper. T_2 is the brake delay time, 0.6 is taken in this paper, a_1 is the maximum braking deceleration, 8m/s^2 is taken in this paper, T_h is the time distance of the robot head, generally 1.2–2.0 s, and 1.2 s in this paper, and d_0, s_0 is the robot distance after the two robots stop at high speed and low speed, generally 2–5 m, and 3 m in this paper.

3. Design of Obstacle Avoidance Strategy

The robot's active obstacle avoidance system should be able to monitor the information through the robot's perception layer, including real-time motion factors such as the robot's distance from the previous robot, the robot's speed relative to the previous robot, and the driving status of the robot in different motion paths. In this way, the safety situation of the robot in its current state is judged, and the selection of vertical obstacle avoidance or horizontal obstacle avoidance is performed. When the driver faces an uncontrollable danger, the active obstacle avoidance system should choose a lower-risk obstacle avoidance strategy for the robot. This risk is not only a risk to the self-robot but also a risk to the entire moving environment. Therefore, when designing the obstacle avoidance strategy, this paper mainly considers motion safety.

Common robots drive normally. During the braking process, only the relative speed and relative displacement relationship between the self-robot and the front robot need to be considered, and it has no effect on robots in other motion paths. Therefore, in the process of active obstacle avoidance, priority should be given to braking obstacle avoidance, which is the safest decision for the entire sports driving process. However, when obstacle avoidance cannot be performed by braking, for example, if the robot in front encounters an emergency danger and suddenly brakes suddenly, the relative speed of the self-robot and the front robot is too large, or the relative displacement is too small. The self-robot still cannot successfully brake and avoid obstacles at the maximum deceleration that can be achieved. At this time, lateral lane change should be considered to avoid obstacles. However, when changing lanes to avoid obstacles, the robot's information on the target movement path of the lane-changing target should also be considered.

During the process of lateral obstacle avoidance, the movement hazards that occur are mainly divided into the following categories: first, when the robot is changing lanes to avoid obstacles, the speed of the self-robot is too fast, or the robot in front of the current moving path is too slow due to emergency braking, resulting in the distance between the self-robot and the robot in front of the current moving path. If it is too close, it will collide with the robot in front of the current movement path when changing lanes to avoid obstacles; second, when the robot is changing lanes to avoid obstacles, the speed of the robot behind the current movement path is too fast. When changing lanes, the rear robot will directly hit the self-robot, but due to this situation, the self-robot cannot avoid obstacles through effective behavior, is an uncontrollable rear-end collision, and will not be considered in this article; third, when the robot moves to the left to avoid obstacles because the robot on the left path is generally faster, when changing lanes to avoid obstacles, the rear robot on the left path may be too fast, causing the robot to change. If the road is not timely, it causes more sports accidents; fourth, when the robot changes lanes to the right to avoid obstacles, the speed of the robot on the right is too low, and the robot in front of the right motion path may be rear-ended. There are often obstacles in the right motion path, so when changing lanes to avoid obstacles, the risk of exercise is also greatly increased.

When the robot is changing lanes and the road motion conforms to the above four conditions, lateral obstacle avoidance may cause more dangerous and uncontrollable motion accidents, which do not conform to the obstacle avoidance strategy based on motion safety criteria in this paper. Therefore, choosing vertical obstacle avoidance is a safer movement strategy at this time. Through the analysis of the safety obstacle avoidance model and obstacle avoidance strategy, it can be seen that the decision-making mechanism of the active obstacle avoidance system in this paper is shown in Figure 7.

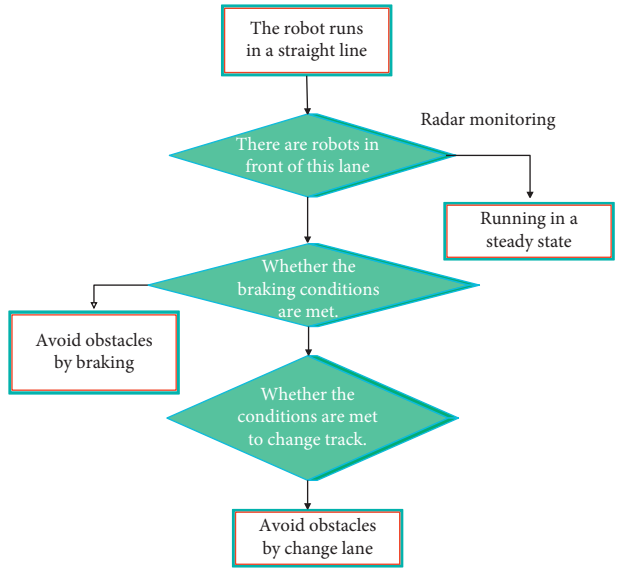

Figure 7: Flowchart of the decision-making mechanism.

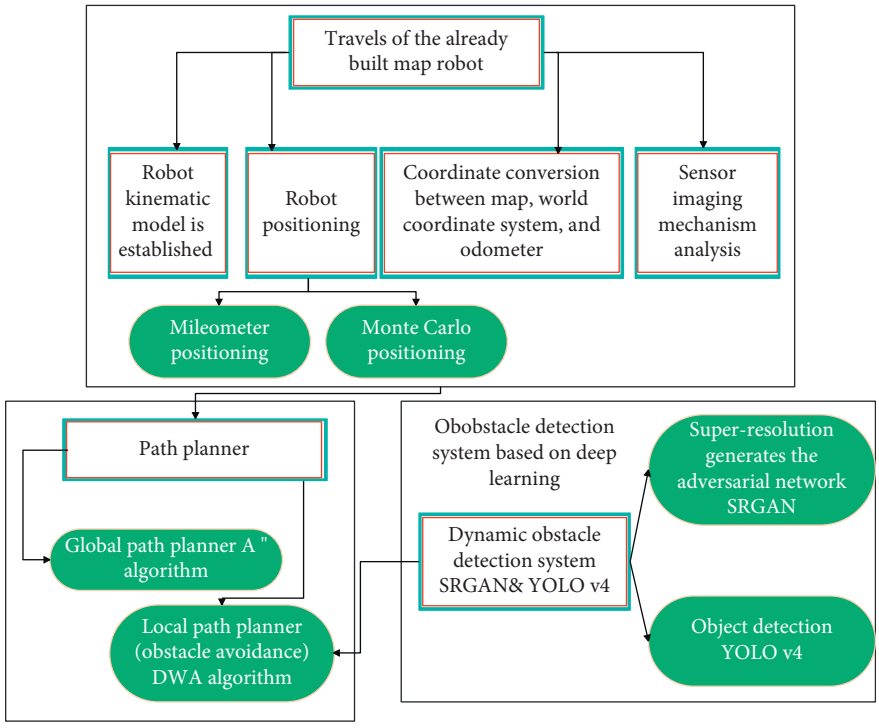

Figure 8: Overall architecture diagram of the robot route planning and obstacle avoidance system based on deep learning.

4. Motion Route Planning and Obstacle Avoidance Method of Mobile Robot Based on Deep Learning

Figure 8 shows the overall architecture of the deep learning-based route planning and obstacle avoidance system constructed in this paper.

It constructs a robot route planning and obstacle avoidance system based on deep learning, and then the motion route planning and obstacle avoidance effect of the proposed model are verified. The results shown in Tables 1 and 2 and Figures 9 and 10 are obtained.

From the above research, it can be seen that the robot based on deep learning proposed in this paper has a good

Table 1: The route planning effect of the robot based on deep learning.

Number	Route plan
1	68.545
2	70.975
3	75.667
4	78.082
5	78.423
6	76.504
7	72.683
8	75.178
9	70.075
10	78.499
11	71.253
12	71.334
13	74.205
14	77.906
15	68.653
16	69.438
17	71.715
18	76.613
19	73.530
20	75.326
21	69.810
22	76.105
23	74.340
24	71.330
25	67.749
26	77.695
27	74.919
28	73.891
29	68.869
30	78.396

Table 2: Obstacle avoidance effects of robots based on deep learning.

Number	Avoidance
1	78.754
2	72.134
3	78.065
4	72.351
5	79.500
6	73.767
7	71.953
8	73.466
9	72.781
10	78.214
11	76.874
12	81.140
13	77.948
14	78.821
15	77.924
16	72.228
17	76.499
18	80.341
19	75.530
20	76.700
21	78.129
22	72.041
23	79.942
24	74.381
25	71.999
26	73.878
27	77.475
28	80.792
29	80.005
30	81.837

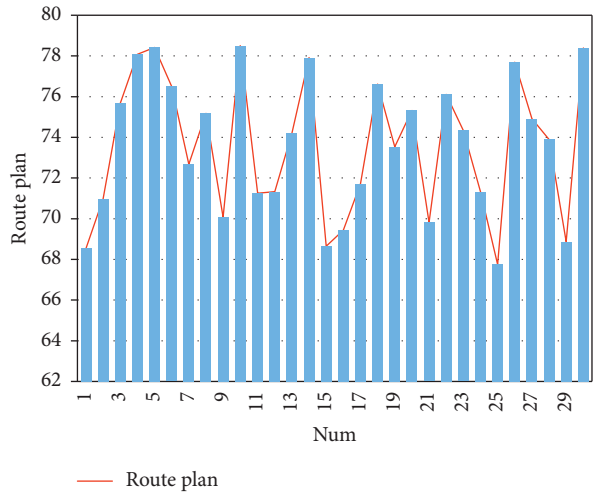

Figure 9: Statistical chart of the route planning effect of the robot based on deep learning.

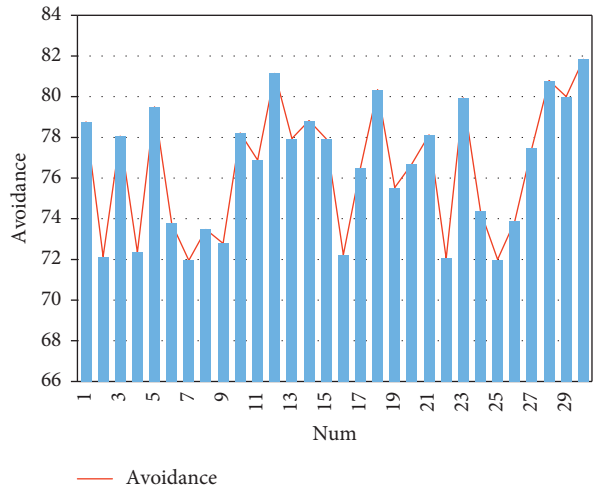

Figure 10: Statistical diagram of the obstacle avoidance effect of the robot based on deep learning.

motion path planning effect and obstacle avoidance effect and can effectively improve the autonomous motion effect of the robot.

5. Conclusion

Path planning is the core problem of mobile robot navigation research. Its goal is to search for an optimal or suboptimal collision-free path from a specified starting point to a target point in an obstacle environment through a certain planning algorithm. At present, the research on the global path planning algorithm with known static environment information has been relatively mature. However, in actual implementation, the environmental information obtained by the robot is often incomplete, and there are dynamic changes. Therefore, how to perform path planning in a dynamic environment is still a difficult problem. This paper combines the deep learning theory to analyze the process of mobile robot motion path planning and obstacle avoidance and builds an intelligent model through simulation research. The simulation experiment study shows that the robot based on deep learning proposed in this paper has a good motion path planning effect and obstacle avoidance effect and can effectively improve the autonomous motion effect of the robot.

Data Availability

The labeled dataset used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was supported by the Henan Provincial Department of Education (Key Technology Research of Intelligent Mobile Handling Robot Based on Visual Perception, No. 21A520007).

References

[1] Y. Chen, K. Feng, J. Lu, and Z. Hu, "Machine vision on the positioning accuracy evaluation of poultry viscera in the automatic evisceration robot system," *International Journal of Food Properties*, vol. 24, no. 1, pp. 933–943, 2021.

[2] A. I. Martyshkin, "Motion planning algorithm for a mobile robot with a smart machine vision system," *Nexo Revista Científica*, vol. 33, no. 02, pp. 651–671, 2020.

[3] A. Chaudhury, C. Ward, A. Talasaz et al., "Machine vision system for 3D plant phenotyping," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 16, no. 6, pp. 2009–2022, 2018.

[4] D. Wang, H. Song, and D. He, "Research advance on vision system of apple picking robot," *Transactions of the Chinese Society of Agricultural Engineering*, vol. 33, no. 10, pp. 59–69, 2017.

[5] L. Cheng, B. Song, Y. Dai, H. Wu, and Y. Chen, "Mobile robot indoor dual Kalman filter localisation based on inertial measurement and stereo vision," *CAAI Transactions on Intelligence Technology*, vol. 2, no. 4, pp. 173–181, 2019.

[6] V. N. Ganesh, S. G. Acharya, S. Bhat, and S. V. Yashas, "Machine vision robot with real time sensing," *Journal of Advancements in Robotics*, vol. 1, no. 3, pp. 30–34, 2020.

[7] H. H. Chu and Z. Y. Wang, "A study on welding quality inspection system for shell-tube heat exchanger based on machine vision," *International Journal of Precision Engineering and Manufacturing*, vol. 18, no. 6, pp. 825–834, 2017.

[8] V. Villani, F. Pini, F. Leali, C. Secchi, and C. Fantuzzi, "Survey on human-robot interaction for robot programming in industrial applications," *IFAC-PapersOnLine*, vol. 51, no. 11, pp. 66–71, 2018.

[9] S. Huang, K. Shinya, N. Bergström, Y. Yamakawa, T. Yamazaki, and M. Ishikawa, "Dynamic compensation robot with a new high-speed vision system for flexible manufacturing," *International Journal of Advanced Manufacturing Technology*, vol. 95, no. 9-12, pp. 4523–4533, 2018.

[10] H. Zhang, M. Li, S. Ma, H. Jiang, and H. Wang, "Recent advances on robot visual servo control methods," *Recent Patents on Mechanical Engineering*, vol. 14, no. 3, pp. 298–312, 2021.

[11] S. Smys and G. Ranganathan, "Robot assisted sensing, control and manufacture in automobile industry," *Journal of ISMAC*, vol. 01, no. 03, pp. 180–187, 2019.

[12] S. Papanastasiou, N. Kousi, P. Karagiannis et al., "Towards seamless human robot collaboration: integrating multimodal interaction," *International Journal of Advanced Manufacturing Technology*, vol. 105, no. 9, pp. 3881–3897, 2019.

[13] J. Li, J. Yin, and L. Deng, "A robot vision navigation method using deep learning in edge computing environment," *EURASIP Journal on Applied Signal Processing*, vol. 2021, no. 1, pp. 1–20, 2021.

[14] Y. Onishi, T. Yoshida, H. Kurita, T. Fukao, H. Arihara, and A. Iwai, "An automated fruit harvesting robot by using deep learning," *Robomech Journal*, vol. 6, no. 1, pp. 1–8, 2019.

[15] Y. Xiong, Y. Ge, L. Grimstad, and P. J. From, "An autonomous strawberry-harvesting robot: design, development, integration, and field evaluation," *Journal of Field Robotics*, vol. 37, no. 2, pp. 202–224, 2020.

[16] Y. Cho, J. Jeong, and A. Kim, "Model-assisted multiband fusion for single image enhancement and applications to robot vision," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 2822–2829, 2018.

[17] A. Tabb and K. M. A. Yousef, "Solving the robot-world hand-eye (s) calibration problem with iterative methods," *Machine Vision and Applications*, vol. 28, no. 5, pp. 569–590, 2017.

[18] M. M. Al-Isawi and J. Z. Sasiadek, "Guidance and control of a robot capturing an uncooperative space target," *Journal of Intelligent and Robotic Systems*, vol. 93, no. 3, pp. 713–721, 2019.

[19] S. Ali, Y. Jonmohamadi, Y. Takeda, J. Roberts, R. Crawford, and A. K. Pandey, "Supervised scene illumination control in stereo arthroscopes for robot assisted minimally invasive surgery," *IEEE Sensors Journal*, vol. 21, no. 10, pp. 11577–11587, 2020.

[20] X. Xu, "Research on teaching reform of mechanical and electronic specialty based on robot education," *Open Access Library Journal*, vol. 8, no. 8, pp. 1–9, 2021.