Traffic & Transportation

Obstacle Avoidance Path Planning Strategy for Autonomous Vehicles Based on Genetic Algorithm

Xiaofeng WENG[1], Fei LIU[2], Sheng ZHOU[3], Jiacheng MAI[4], Shaoxiang FENG[5]

[1] 592630327@qq.com, Shanghai University of Engineering Science
[2] Corresponding author, liufei2021186@163.com, Shanghai University of Engineering Science
[3] 632539247@qq.com, Shanghai University of Engineering Science
[4] 872733836@qq.com, Shanghai University of Engineering Science
[5] 396176325@qq.com, Shanghai University of Engineering Science

Original Scientific Paper
Submitted: 2 Nov. 2023
Accepted: 27 Feb. 2024

Publisher:
Faculty of Transport and Traffic Sciences,
University of Zagreb

ABSTRACT

In order to enhance the driving ability of autonomous vehicles on structured roads and enable them to plan safe and comfortable paths, we propose an obstacle avoidance path strategy for autonomous vehicles based on genetic algorithm. The use of Frenet-Serret enhances the adaptability of the algorithm in complex environments. In order to improve the generation and optimisation of obstacle avoidance trajectory, we establish an anti-collision model. When the vehicle faces a potential collision with an obstacle, the genetic algorithm quickly iterates and selects the first nine genes to generate the rough solution and convex space of the path. Combined with convex space, the quadratic programming method will numerically optimise the generated rough solution to generate an accurate path that satisfies the constraints. In addition, in order to ensure the safety and comfort in the process of obstacle avoidance, based on the dynamic constraints of the vehicle, the speed planning is used to determine the speed curve. We simulate in various scenarios involving moving obstacles. The real-time simulation based on the HIL platform proves that the proposed path planning strategy is effective in various driving scenarios.

KEYWORDS

autonomous vehicle; genetic algorithm; anti-collision model; path planning; sequential quadratic programming (SQP).

1. INTRODUCTION

Currently, autonomous vehicle technology has garnered global attention from scholars. Research has demonstrated that driverless cars can significantly reduce the escalating number of traffic accidents [1–3]. With the aid of high-performance chips, autonomous vehicle technology can make faster decisions and path planning, resulting in an improved driving and traffic safety [4].

Autonomous vehicle systems comprise three modules: perception, planning and control [5]. The path planning module plays a crucial role in ensuring ride comfort, convenience and safety. Path planning is divided into two categories: global path planning and local path planning [6]. While global path planning designs the path from the starting point to the endpoint based on the map, it cannot update the driving environment in real-time. Therefore, local path planning is necessary as an auxiliary to ensure the safe arrival of autonomous vehicles at the target point [7].

Obstacle avoidance has always posed a significant challenge in the local path planning of driverless cars [8,9]. After identifying obstacles, driverless cars must make decisions and devise a suitable obstacle avoidance route to ensure their own safety. Currently, numerous obstacle avoidance planning algorithms have been tested on drones or robots, but there are still several unresolved issues when it comes to autonomous vehicles [10–12].

Most obstacle avoidance algorithms for autonomous vehicles can be categorised into two types. The first type is based on traditional obstacle avoidance path algorithms, such as the artificial potential field method (APF) and open space methods [13–15]. The second type relies on intelligent bionic algorithms, including the whale algorithm, genetic algorithm and particle swarm algorithm.

The artificial potential field (APF) method, a traditional algorithm, draws inspiration from mechanics and generates a repulsion potential field for obstacles and an attraction potential field for the target [16]. This enables the vehicle to avoid colliding with the obstacle boundary while driving towards the target [17]. By combining the potential field function with the spatial sampling method and employing the Bessel curve for path point smoothing, Zheng et al. [18] achieved the ability for the vehicle to navigate towards the target while effectively avoiding collision with the obstacle boundary. They successfully used the potential field evaluation function to determine the optimal path curve. Yang et al. [19] improved the target point by including a certain range of multi-target points, which altered the direction of gravity and resolved the local minimum problem of the path. However, the APF algorithm may also be prone to falling into local minimums and traps, making it difficult to plan a suitable path in complex scenarios.

The genetic algorithm is a heuristic optimisation algorithm that simulates the natural evolution process and is commonly used to solve search and optimisation problems. Its design inspiration comes from the genetic and natural selection mechanisms in biology. The genetic algorithm searches for the optimal solution by simulating genetic, crossover and mutation operations in the evolution process. Through repeated iterations, the genetic algorithm can gradually optimise individuals in the population and ultimately find an individual or solution set close to or reaching the optimal solution. Rui et al. [20] proposed a hybrid genetic algorithm that transforms the local search process into a mutation operator. This approach not only has a simple structure but also guarantees real-time performance. Liu J et al. [21] proposed a fast path planning method based on an improved genetic algorithm according to the principle of hierarchical path planning. Simulation results show that the optimisation time is significantly shortened. Chen et al. [22] developed an improved genetic algorithm for the path planning of intelligent four-wheeled robots. The algorithm has better convergence speed and optimisation probability than the traditional genetic algorithm. Hou et al. [23] proposed an enhanced ant colony algorithm with a communication mechanism for path planning. Through direct communication between individuals, the integration of historical paths is accelerated, and the path selection rules and heuristic functions are improved to enhance the convergence speed and search efficiency of the algorithm. Liu et al. [24] proposed a path planning method based on an improved grey wolf algorithm, which introduced interference factors and dynamic weights based on the lion optimisation algorithm to avoid the loss of diversity. However, the ability to jump out of local optima needs to be strengthened. Kumar et al. [25] proposed a path planning method combining the artificial bee colony and evolutionary programming algorithm. The artificial bee colony algorithm is used to perform a preliminary search according to the improved strategy, and then the evolutionary algorithm is used to refine the feasible path to reduce the search cost. Xu et al. [26] introduced the disaster strategy and dynamic mutation operator embedded in the A* algorithm into the genetic algorithm to reduce the premature phenomenon and improve the local search ability of the algorithm in the later stage. At the same time, the fitness function with multiple constraints improves the smoothness of the planned path. However, each disaster initialises the population, which reduces computational efficiency.

Genetic algorithm has excellent search performance, can quickly iterate and find the optimal solution [27]. Making full use of the suboptimal solution helps to obtain a safer and more reasonable planning trajectory. However, most of the research in this field mainly focuses on the application of mobile robots, and there is little research on obstacle avoidance path planning for autonomous vehicles. In addition, many related studies only consider two-lane, without considering the complex situation of obstacle avoidance in more lanes [28].

This paper proposes an obstacle avoidance strategy for driverless cars based on a genetic algorithm. The obstacle avoidance trajectory planning for autonomous vehicles is divided into two modules: path planning and speed planning. The path planning module receives input from perceived obstacle information, global path information and vehicle status information. It consists of convex space, genetic algorithm, numerical optimisation and speed modules. The first three modules work together to solve and optimise a path that satisfies the given constraints, while the speed module adds speed information to the path, resulting in a complete set of path data. Finally, this data is input to the control module and also used as input for the next cycle of the path planning module.

The main contributions of this paper are as follow:

1) Genetic algorithm is used to assist autonomous cars to make light decisions.
2) Genetic algorithm is integrated into three-lane obstacle avoidance trajectory planning.
3) A new security obstacle avoidance strategy is proposed.

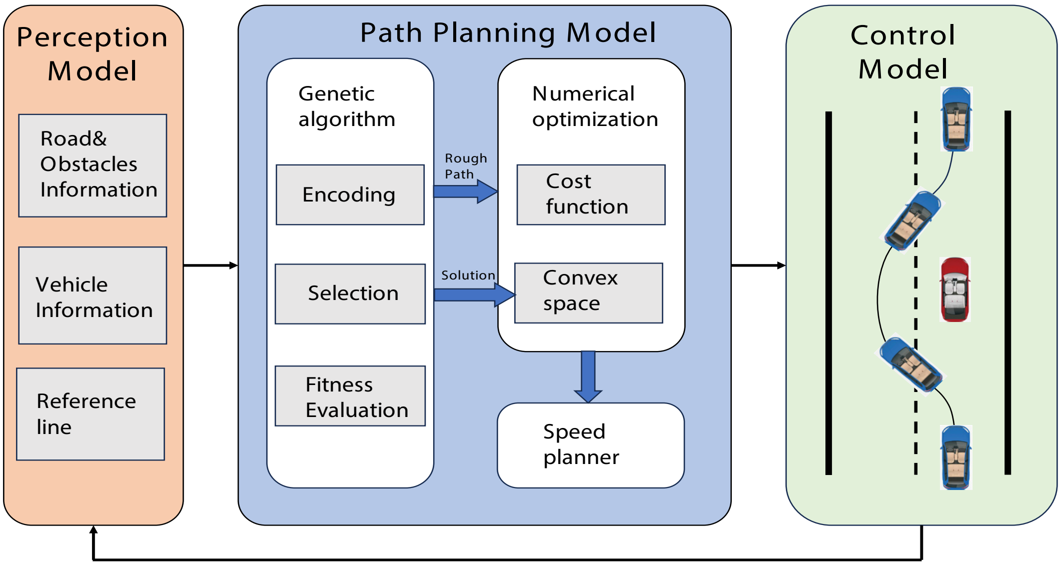

Figure 1 – Structure diagram of the obstacle avoidance strategy

2. MODELLING

2.1 Frenet-Serret

Compared with the traditional coordinate system, the Frenet-Serret can more intuitively represent and understand the position and direction of the vehicle on the curved road. Using Frenet-Serret for path planning will be simpler, more efficient and easier to implement.

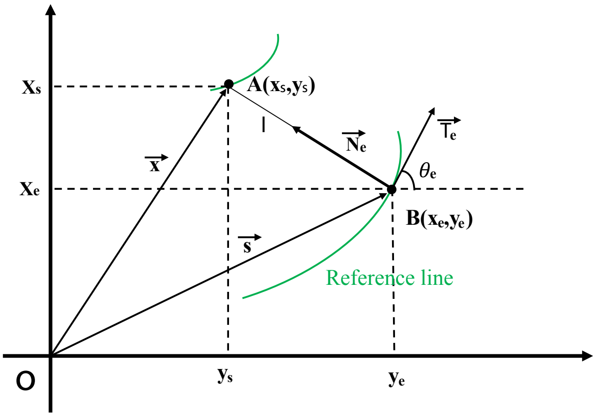

Figure 2 – Frenet-Serret

Suppose that the vehicle is in A (x_s,y_s), Point B (x_e,y_e) is the projection point of the vehicle on the reference line. θ_e is its tangent vector angle.

$$\begin{cases} d_x = x_s - x_e \\ d_y = y_s - y_e \\ \iota = sign((y_s - y_e)\cos(\theta_e) - (x_s - x_e)\sin(\theta_e))\sqrt{(x_s - x_e)^2 + (y_s - y_e)^2} \end{cases} \tag{1}$$

The vehicle heading angle and path curvature can be written in the following form**:**

$$\begin{cases} \vec{x} = \vec{s} + \iota\overrightarrow{N_e} \\ x = x_e - \iota sin\,\theta_e \\ y = y_e + l\,cos\,\theta_e \\ \theta_e = tan^{-1}\left(\frac{y'_e}{x'_e}\right) \\ k_e = \frac{x'_e y''_e - x''_e y'_e}{(x_e'^2 + y_e'^2)^{\frac{3}{2}}} \end{cases} \tag{2}$$

2.2 Anti-collision model

The anti-collision model is established to ensure that the driverless vehicle has sufficient response distance to deal with emergencies. It can ensure that the driverless car has sufficient ride comfort and safety when performing obstacle avoidance.

Longitudinal anti-collision distance

Based on the Frenet-Serret, the deceleration ability of the autonomous vehicle and the obstacle information, the longitudinal anti-collision distance is established in this paper.

$$S = S_{br} + S_m \tag{3}$$

where S_{br} is the linear braking reserve distance of the autonomous vehicle. S_m is the safe driving distance that needs to be maintained between the front and rear vehicles. S_{br} and S_m can be expressed as follows:

$$\begin{cases} S_{br} = \frac{v_{AV^2}}{2a_{D\_max}} + v_{AV}t_1 + v_r t_2 + d_{D\_min} \\ S_m = \frac{3v_{AV}}{\kappa(f+g)} + 0.5l \\ v_r = |v_{AV} - v_{fc}| \end{cases} \tag{4}$$

where t_1 sets aside running time for the algorithm, which takes 0.1 seconds here. t_2 represents the time delay of the brake, $t_2 = 0.4$s. represents the speed of the vehicle. a_{D_max} represents the maximum braking deceleration. The minimum safe stopping distance $d_{D\_min} = 3.5$.

S_m represents the safety distance between the front and rear vehicles. According to *Equation 4*, it shows a correlation with road adhesion coefficient and vehicle speed. *k* represents the safety factor, *f* represents the adhesion coefficient of the ground and the coefficient constant $g = 0.3$. It is concluded that the safety distance S_m between the front and rear vehicles will increase in muddy roads, while when the road is dry, it will decrease. v_r represents the relative speed of the two vehicles, which is the absolute value of the speed difference between the two vehicles.

Lateral anti-collision distance

The lateral anti-collision distance of the driverless car is related to whether the driverless car will obliquely collide or rub with the obstacle vehicle during obstacle avoidance. Assuming that the length of all vehicles in this paper is *l* and the width is *w*, in order to reduce the amount of calculation for the subsequent algorithm, the lateral safety distance is taken as a fixed value.

$$w_0 = \frac{3}{2}w \tag{5}$$

Collision warning area

The longitudinal anti-collision distance and the lateral anti-collision distance are combined into a collision warning area. Once a driverless car enters this area, it means that the possibility of collision is greatly increased.

The anti-collision model of the moving obstacle and the collision warning area are shown in the yellow dotted line and the red dotted line respectively. The long axis will change according to the driving direction of the obstacle vehicle. Its expression can be written as follows:

$$\begin{cases} \dfrac{x^2}{(a-b)x+ab}+\dfrac{y^2}{c^2}=1 \\ d_1=\sqrt{b^2-c^2} \\ d_2=\sqrt{a^2-c^2} \end{cases} \tag{6}$$

where a and b are the major axis of the left and right semi-ellipses, respectively, and c is the short half-axis. d_1, d_2 represent the half focal length of the left and right ellipses, respectively. The expressions for the major and minor axes of the collision warning area are as follows:

$$\begin{cases} a=\dfrac{S_{br}+S_m}{2} \\ b=l+0.8|v_r| \\ c=1.5l \end{cases} \tag{7}$$

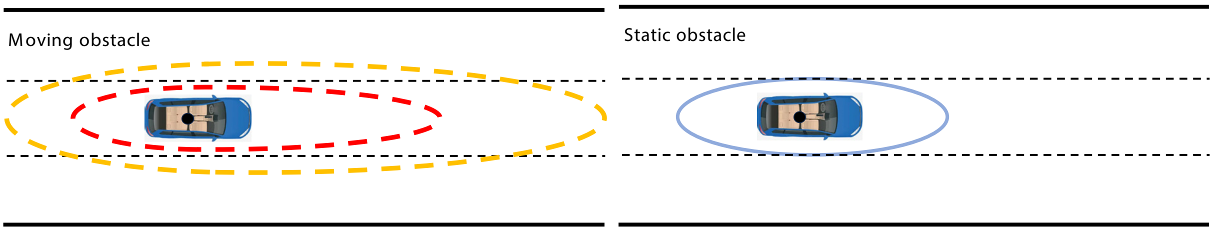

Figure 3 – Moving and static obstacle collision warning area

3. OBSTACLE AVOIDANCE STRATEGY BASED ON GENETIC ALGORITHM

3.1 Planning path points

Path planning based on genetic algorithm needs to generate path points on the road. The number and arrangement of path points will greatly affect the computational efficiency of the algorithm and the quality of the generated path. Through a large number of experiments, it is concluded that the effect of nine rows and six columns is better. Therefore, this paper chooses to create nine rows and six columns of path points for the vehicle in front of the vehicle. With the aim of covering the entire three-lane structured road, the distance between each line of path points is 0.5 meters. In order to ensure sufficient planning distance, the distance between each column is dynamically adjusted and will change with the speed of the vehicle. When the vehicle speed $v_e \leq 10$ m/s, the distance of S_p=40m is always planned forward; when $v_e > 10$ m/s, the representation of the longitudinal distance for forward planning is as follows:

$$S_p = v_e \cdot t_0 \tag{8}$$

Path planning requires sufficient distance to avoid obstacles. However, planning too far away is meaningless and will increase the amount of calculation. So this paper chooses t_0=4*s*.

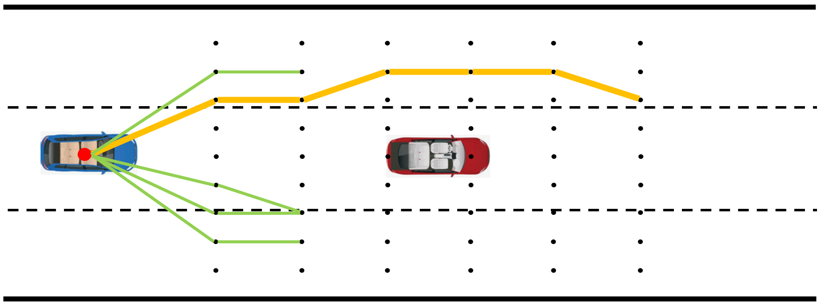

Figure 4 – Creating path points

3.2 Genetic algorithm

Encoding

In this paper, each chromosome O_p is composed of multiple genes o_i and each gene is composed of a_i,b_i,c_i,d_i,e_i,f_i $(1 \leq i \leq 9)$. Where a_i,b_i,c_i,d_i,e_i,f_i are randomly selected points in each column. Therefore, a gene can be regarded as a path.

Path encoding through path points a_i,b_i,c_i,d_i,e_i,f_i $(1 \leq i \leq 9)$ can quickly and clearly represent many paths. The specific expression for chromosome O_p is as follows:

$$O_p = [o_1 o_2 o_3 o_4 o_5 \dots o_{20}] = \begin{bmatrix} a_{1,p} & a_{2,p} & \cdots & a_{20,p} \\ b_{1,p} & & \cdots & \\ c_{1,p} & & \cdots & \\ d_{1,p} & & \cdots & \\ e_{1,p} & & & \\ f_{1,p} & & & f_{20,p} \end{bmatrix}, \quad p = 1,2 \dots N_{pop} \tag{9}$$

Initialisation

The generation of the population is related to whether the optimal solution can be quickly solved and whether it will fall into the local optimal solution. Randomly generated individuals will increase the scope of the search and make it easier to find the optimal solution. Due to the fact that excellent individual genes will be retained in the next generation to ensure continuous progress. Therefore, 70% of the individuals in the population will take randomly generated results, 20% of the individuals are obtained through the mutation of the best chromosome and 10% are inherited from the best chromosome of the previous generation. This can ensure that the overall adaptability of each iteration is increased, and 10% inheritance can also avoid the algorithm falling into the local optimal solution.

Selection

In this paper, two steps are taken to screen chromosomes. Each chromosome has its corresponding fitness function value, which is arranged from low to high and the top 10% of the excellent chromosomes are inherited and mutated. Mutation is to replace the three path nodes of the chromosome with an adjacent node. These three points adopt the principle of roulette, and the closer to the obstacle, the easier it is to be selected as the mutation point. The mutation method is not equal probability random, but there is a 60% probability of mutation into a point away from the obstacle and a 40% probability of mutation into a point near the obstacle.

$$\begin{cases} p_i = ke^{-\mu d_i} \\ P = \dfrac{p_i}{\sum\limits_{i=1}^{6} p_i} \\ d_1 = \sqrt{(x_i - x_{obs}) + (y_i - y_{obs})} \end{cases} \tag{10}$$

where d_i is the distance between the ith point on the gene and the obstacle, p is the probability of being selected according to the distance. k is the normalised coefficient, μ is the distance coefficient.

Mutation

The fitness function of obstacle avoidance needs to consider factors such as comfort, safety and path length [29,30]. Therefore, this function consists of multiple parts. In terms of comfort, the lateral displacement and lateral acceleration of the path are limited to ensure that there is no large lateral acceleration. The security of the path will be ensured by the obstacle avoidance model. Once the path planning enters the obstacle avoidance warning area, the value of the cost function will change dramatically.

Cost function can be written as follows:

$$F_{cost} = \omega_1 S_{lat} + \omega_2 C_{acc} + \omega_3 C_{col} + \omega_4 C_{bk} \tag{11}$$

where S_{lat} is the difference between the maximum lateral displacement path point and the road centerline, which represents the lateral displacement cost of the obstacle avoidance path of the autonomous vehicle. The cost of the lateral acceleration represented by C_{acc} is calculated by the slope of the two points. C_{col} represents the collision cost. If the driverless car enters the prohibited driving area, this cost function will be triggered, expressed as 1, otherwise 0. Generally speaking, the shorter the distance of obstacle avoidance path planning, the better. Therefore, C_{bk} is used to represent the longitudinal distance between the starting point and the end point of obstacle avoidance, i.e. the distance cost. ω_1, ω_2, ω_3, ω_4 are the cost coefficients of these costs, where ω_3 is set to be particularly large to prevent collisions.

Fitness function:

$$J = \frac{1}{1 + F_{cost}} \tag{12}$$

Solving feasible convex space

In this paper, nine genes after genetic algorithm iteration are selected as the basis for opening convex space. This can avoid ignoring safety due to the cost value, and will not let the autonomous car pursue the optimal fitness function too much. The number of iterations will greatly affect the real-time performance of the algorithm. After a large number of experiments, 60 iterations are selected to balance the quality of the rough solution and the amount of calculation. The points on the rough path are expressed as l_{i_center}. Based on the points on the rough solution of the path, extend l_i to both sides to form a passable convex space. The constraint condition of of convex space is as follows:

$$\begin{cases} l_{min} \leq l_i \leq l_{max} \\ \frac{1}{4}w \leq l_i \leq \frac{5}{4}w \end{cases} \tag{13}$$

l_{max}, l_{min} represents the upper and lower bounds of horizontal coordinates. Based on the location of obstacles and the anti-collision distance model, l_{max}, l_{min} can be expressed as follows:

$$\begin{cases} l_{max} = l_{obs} \dfrac{\sqrt{l^2 + w^2 \cdot \sin\left(\tan^{-1}\frac{w}{l} + \Delta\theta\right)}}{2} - \Delta s \quad l_i < l_{obs} \\ l_{min} = l_{obs} \dfrac{\sqrt{l^2 + w^2 \cdot \sin\left(\tan^{-1}\frac{w}{l} + \Delta\theta\right)}}{2} + \Delta s \quad l_i \geq l_{obs} \end{cases} \tag{14}$$

where Δs is the reserved distance for the convex space and the obstacle. $\Delta\theta$ is the angular difference between the orientation of the vehicle and that of the obstacle.

3.3 Optimisation path based on quadratic programming

The rough solution of the path needs to be optimised. The quadratic optimisation is simple and efficient, and can search for the global optimal solution. Moreover, quadratic optimisation can use inequality to represent constraints. Combined with the feasible convex space solved above, this paper chooses sequential quadratic programming as the optimisation method. In the optimisation, we will pay attention to two aspects of safety and comfort. The expression function of the path is as follows:

$$l = a_0 + a_1 s_i + a_2 s_i^2 + a_3 s_i^3 + a_4 s_i^4 + a_5 s_i^5 \tag{15}$$

In this paper, the quintic polynomial is selected as the representation of the path to ensure that the planned path has continuous curvature. In order to simplify the calculation and increase the real-time performance of the algorithm, the path planning will be generated in the Frenet-Serret and then converted to the Cartesian coordinate system [31]. The motion path point is defined as T_n^i=(s_i,l_i). In order to meet the safety and occupant comfort of the solution path, the designed cost function consists of the following items:

$$J = r_0 J_{lat} + r_1 J_{l\_a} + r_2 J_{l\_b} + r_3 J_{col} + r_4 J_{lin} \tag{16}$$

The cost function consists of five parts. J_{lat} is the cost incurred due to lateral displacement. J_{l_a} is the cost associated with the rate of change of lateral displacement. J_{l_b} is the lateral second derivative cost. J_{col}

is the collision warning cost. And J_{lin} is the path change cost. The lateral displacement cost is expressed as follows:

$$J_{lat} = \sum_{i=0}^{n-1} l_i^2 \tag{17}$$

where n is the overall count of planned path points. The establishment of the lateral displacement cost is to prevent excessive lateral displacement of the driverless car, so that the planning path can complete the target with a smaller lateral movement. However, the limitation of lateral displacement will increase the collision risk of autonomous vehicles, so the collision warning cost is established to maintain a reasonable distance between autonomous vehicles and obstacles. Combined with the above collision warning model, the collision warning cost function can be expressed as follows:

$$J_{col} = \sum_{i=0}^{n-1} e^{\frac{\Delta d_i^2}{\sigma^2}} \tag{18}$$

$$\begin{cases} G_1(s_i,l_i) = \sqrt{(s_i - (s_{obs} - d_1))^2 + (l_i - l_{obs})^2} + \sqrt{(s_i - (s_{obs} + d_1))^2 + (l_i - l_{obs})^2} \\ G_2(s_i,l_i) = \sqrt{(s_i - (s_{obs} - d_2))^2 + (l_i - l_{obs})^2} + \sqrt{(s_i - (s_{obs} + d_2))^2 + (l_i - l_{obs})^2} \end{cases} \tag{19}$$

where Δd_i represents the distance between the i^{th} path point and the obstacle. σ is a parameter for smoothing the control cost function. Combined with the anti-collision model and collision warning model, the long axes a and b of the left and right semi-ellipses in the model are selected as the judgment conditions for the function to set zero. When G_1 $(s_i,l_i) < 2a$ or G_2 $(s_i,l_i) < 2b$,it means that the vehicle enters the range of the obstacle avoidance model. At this time, the collision warning cost will generate the value of the corresponding point, limiting the autonomous vehicle to continue to approach the obstacle. Conversely, in other cases, $J_{col} = 0$. This piecewise cost function helps to improve the security of the path.

The purpose of the collision avoidance warning cost is to prevent the collision between the driverless car and the obstacle. However, this can also lead to a larger lateral acceleration of the driverless car, which is fatal for comfort. Therefore, in order to limit the excessive curvature of the path, the lateral displacement change rate cost is added. The choice is made to employ the second derivative of the lateral displacement to characterise the rate of change of the lateral displacement. The expression is as follows:

$$\begin{cases} J_{l\_a} = \sum_{i=0}^{n-1} l'^2_i \\ J_{l\_b} = \sum_{i=0}^{n-1} l''^2_i \end{cases} \tag{20}$$

The lateral displacement changing rate and the second derivative of the lateral displacement of the vehicle can be expressed as follow:

$$l_i' = \frac{l_{i+1} - l_i}{\Delta s} \tag{21}$$

$$l_i'' = \frac{l_{i+1} - 2l_i - l_{i-1}}{\Delta s^2} \tag{22}$$

The path change cost J_{lin} is used to limit the optimised path. The rough solution path obtained above has determined a better obstacle avoidance direction and path. The optimised path should be based on the direction of the original path (rough solution) and there should not be too many conflicts.

$$J_{lin} = \sum_{i=0}^{n-1} (l_i - l_{i\_center})^2 \tag{23}$$

where l_{i_center} is the path point of the rough solution. The difference between the corresponding path points represents the degree of deviation between the optimised path and the coarse solution path.

Thus, by utilising the quintic polynomial of the path to solve for the polynomial coefficients, the aforementioned cost function can be transformed into an expression involving s_i.

This objective function hopes that the path has smaller curvature and better security. Simultaneously, be-sides satisfying environmental constraints, the path must adhere to the kinematic constraints of the vehicle. To address this, the sequential quadratic programming algorithm is employed for problem resolution. The solution problem can be expressed as follows:

$$min{:}J(a_0,a_1,\ldots,a_5)=r_0\sum_{i=0}^{n-1}l_i^2+r_1\sum_{i=0}^{n-1}l'^2_i+r_2\sum_{i=0}^{n-1}l''^2_i+r_3\sum_{i=0}^{n-1}e^{\frac{-\Delta d_i^2}{\sigma^2}}+\sum_{i=0}^{n-1}(l_i-l_{i\_center})^2 \tag{24}$$

$$\begin{cases} l_{i\_min}\le l_i\le l_{i\_max} \\ -\frac{1}{4}w\le l_i\le\frac{7}{4}w \end{cases} \tag{25}$$

4. VELOCITY PLANNING

Speed planning adopts the interpolation method. The time of the discrete points on the path is obtained, and then the time corresponding to the other points on the path is obtained by the difference. The velocity curve can then also be obtained.

The lateral acceleration of the velocity curve must be limited because the friction between the tire and the road is not infinite, and the centrifugal force of the vehicle is provided by the tire. At the same time, excessive lateral acceleration will affect comfort, so the lateral acceleration is limited to obtain a better velocity curve. The restriction expression is as follows:

$$\begin{cases} m\frac{v_i^2}{r}=mv_i^2k_i\le\mu mg \\ m\frac{v_i^2}{r}=mv_i^2k_i\le ma_{y\_m} \end{cases} \tag{26}$$

Among them, k_i and v_i denote the curvature and velocity of the i^{th} path point, respectively. μ represents the coefficient of friction of the road, while a_{y_m} represents the maximum lateral acceleration permissible by the vehicle while ensuring comfort. Speed also needs to be limited to ensure passenger comfort. The speed limit expression is as follows:

$$\begin{cases} v_i\le\sqrt{\frac{\mu g}{k_i}} \\ v_i\le\sqrt{\frac{a_{y\_m}}{k_i}} \end{cases} \tag{27}$$

In reality, driving on the road often encounters road speed limit signs. Therefore, it also needs to be taken into account. Assuming that the maximum speed of the road is v_{r_m}, the maximum speed v_{max} can be expressed as follows:

$$v_{max}=min\left(v_{r\_m},\sqrt{\frac{a_{y\_m}}{k_i}},\sqrt{\frac{\mu g}{k_i}}\right) \tag{28}$$

The discrete point velocity and acceleration can be represented by the corresponding time t_i.

$$\begin{cases} v_i=\frac{\Delta s}{\Delta t_i} \\ a_i=\frac{\Delta v_i}{\Delta t_i} \\ \Delta t_i=t_i-t_{i-1} \\ \Delta v_i=v_i-v_{i-1} \end{cases} \tag{29}$$

The distance, velocity and acceleration of adjacent discrete points satisfy the following relationship:

$$\begin{cases} v_i=v_{i-1}+a_{i-1}\Delta t_i \\ s_i=s_{i-1}+v_{i-1}\Delta t_i+0.5a_{i-1}\Delta t_i^2 \end{cases} \tag{30}$$

In speed planning, jerk, excessive acceleration and speed (without obstacles) close to road constraints are three points to be considered. Therefore, the above three are used as optimisation objectives to construct the cost function of speed planning. The function can be expressed as follows:

$$J_{speed}=r_{acc}C_{s\_acc}+r_{jerk}C_{s\_jerk}+r_{ref}C_{s\_ref} \tag{31}$$

Among them, r_{acc}, r_{jerk}, r_{ref} represent the weight coefficients of each cost respectively. Speed planning can be expressed in following forms:

$$min{:}J(t_i) = r_{acc}\sum_{i=1}^{N_i} a_i^2 \cdot +r_{jerk}\sum_{i=1}^{N_i-1}\left(\frac{a_{i+1}-a_i}{t_{i+1}-t_i}\right)^2 + v_{ref}\sum_{i=1}^{N_i}(v_i - v_{ref})^2 \tag{32}$$

$$\begin{cases} 0 \le v_i \le v_{max} \\ a_{min} \le a_i \le a_{max} \\ v_{i+1} = v_i + a_i \Delta t_{i+1} \\ s_{i+1} = s_i + v_i(f_{i+1} - t_i) + 0.5 a_i \Delta t_i^2 \end{cases} \tag{33}$$

where v_{ref} represents the road reference speed, a_{max}, a_{min} represent the maximum forward and backward acceleration. By transforming the speed planning into the above form to solve the optimisation problem, a more comfortable speed curve can be easily obtained.

5. SIMULATION RESULTS AND ANALYSIS

In this paper, the semi-physical simulation platform (HIL) is selected as the simulation equipment to verify the above planning strategy in real time. In order to verify the effectiveness of the strategy, three high-probability scenarios were selected, including straight and curved roads. The real-time simulation in this chapter uses a three-lane structured road. Self-driving cars need to determine the direction of lane change on their own. This helps to verify the security of our proposed obstacle avoidance strategy. In the figure, the black implementation represents the road boundary, and the black dotted line represents the lane boundary. The real-time simulation on the HIL platform will prove the effectiveness of the strategy on autonomous vehicles.

At the same time, in order to show the advantages of the strategy, it is compared with the artificial potential field method (APF) [16] and the dynamic programming algorithm (DPM) [32]. *Table 1* shows the parameters of the simulated vehicle.

Table 1 – Simulation of vehicle parameters

Unit	Name	Value
Kg	Vehicle weight	1620
mm	Vehicle width	1916
mm	Vehicle wheelbase	2984
s	100 kilometres acceleration time	8.1
Kg.m^2	Moment of inertia around Z axis	3870

In the following multi-scene simulation, the generation interval of trajectory planning is 100 ms, while the trajectory tracking interval of autonomous vehicles is 10 ms. For Scenario 1 and Scenario 3, since the cost of avoiding obstacles from both directions is the same, it is assumed that obstacle avoidance will be performed from the left side of the self-driving car (the lane above the figure). Scenario 2 represents a complex three-lane scene, and the algorithm in this paper will choose a safer left side to avoid obstacles. Therefore, other algorithms are also set to avoid obstacles from the left to ensure the rationality of the comparison.

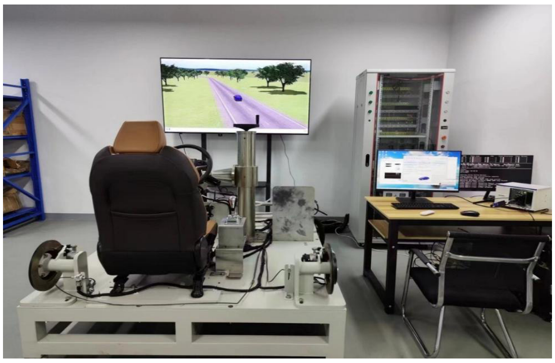

Figure 5 – Hardware-in-the-loop simulation platform (HIL)

5.1 Obstacle avoidance test of straight static obstacle

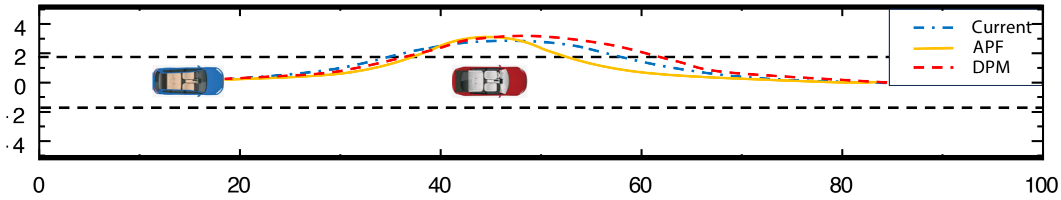

Figure 6 – Straight obstacle avoidance path comparison diagram

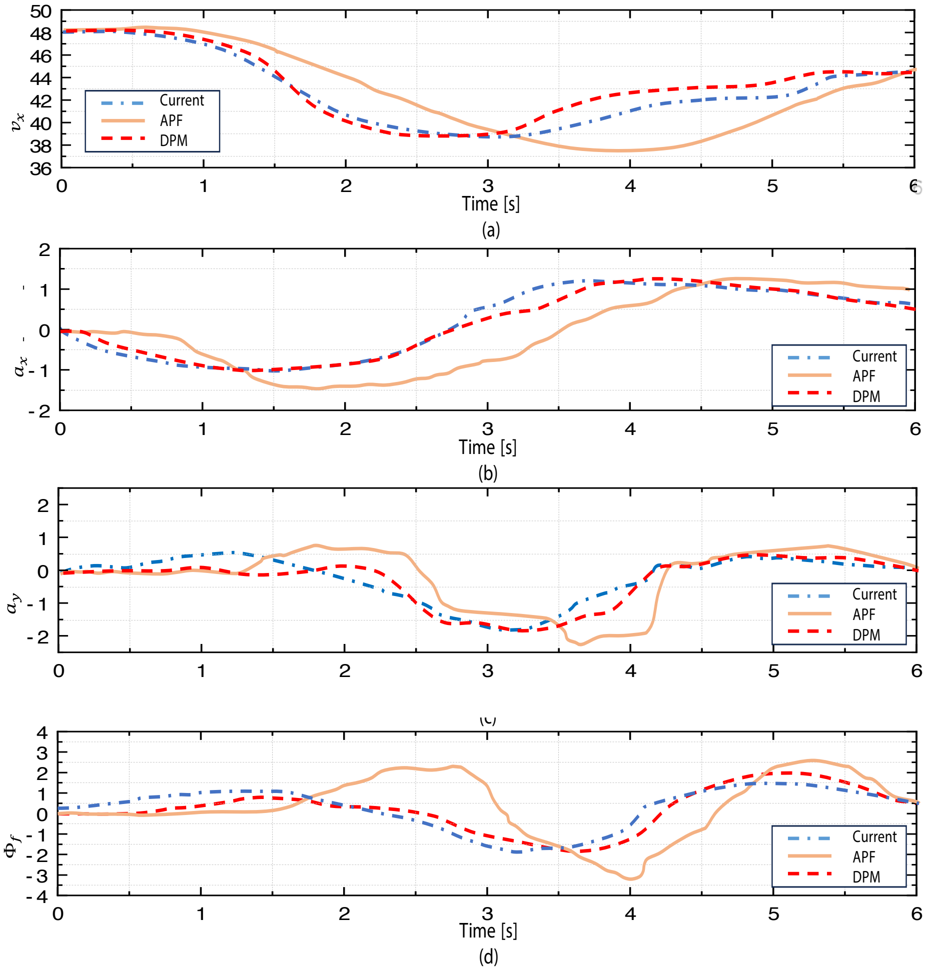

Figure 7 – Straight obstacle avoidance path comparison diagram
(a) Longitudinal velocity v_x (b) Longitudinal acceleration a_x (c) Lateral acceleration a_y (d) Front wheel angle Φ_f

The static obstacle avoidance test on the straight road uses a three-lane structural road and static obstacles to verify the most basic obstacle avoidance. The planning trajectories of these three algorithms are shown in *Figure 6*. Obviously, all algorithms can successfully avoid obstacles and ensure safe driving of autonomous vehicles. However, the algorithm proposed in this paper is the first algorithm to start obstacle avoidance, so there is enough distance to plan a more comfortable trajectory. It can be seen from *Figure 7* that although APF can successfully avoid it, the comfort of its planned trajectory is relatively poor. As shown in *Figure 7a–7b*, the speed of the autonomous vehicle during obstacle avoidance is maintained between 39 and 48km/h, and the longitudinal acceleration is also between −1.7 and 1.3. Compared with the DPM algorithm, the algorithm proposed in this paper has less change in lateral and longitudinal acceleration. At the same time, the front wheel steering angle of the algorithm proposed in this paper is small, which means that the method proposed in this paper has certain advantages in comfort.

5.2 Obstacle avoidance test of curved obstacles

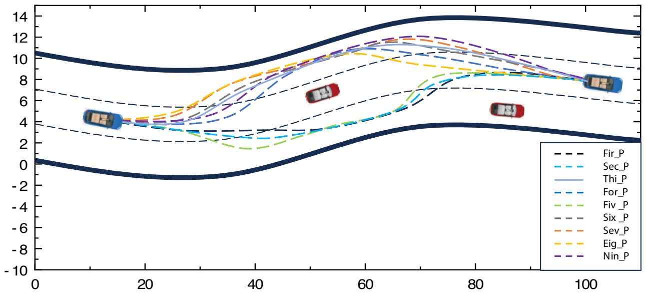

Figure 8 – Obstacle avoidance path generation in curve

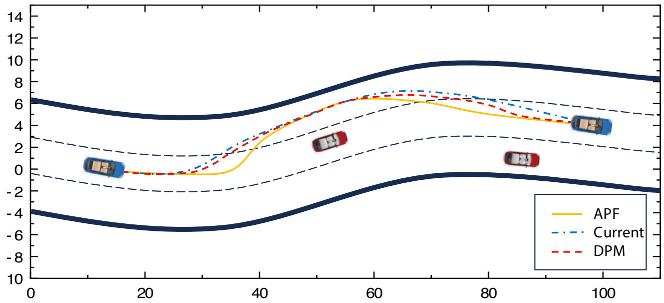

Figure 9 – Curve obstacle avoidance path comparison

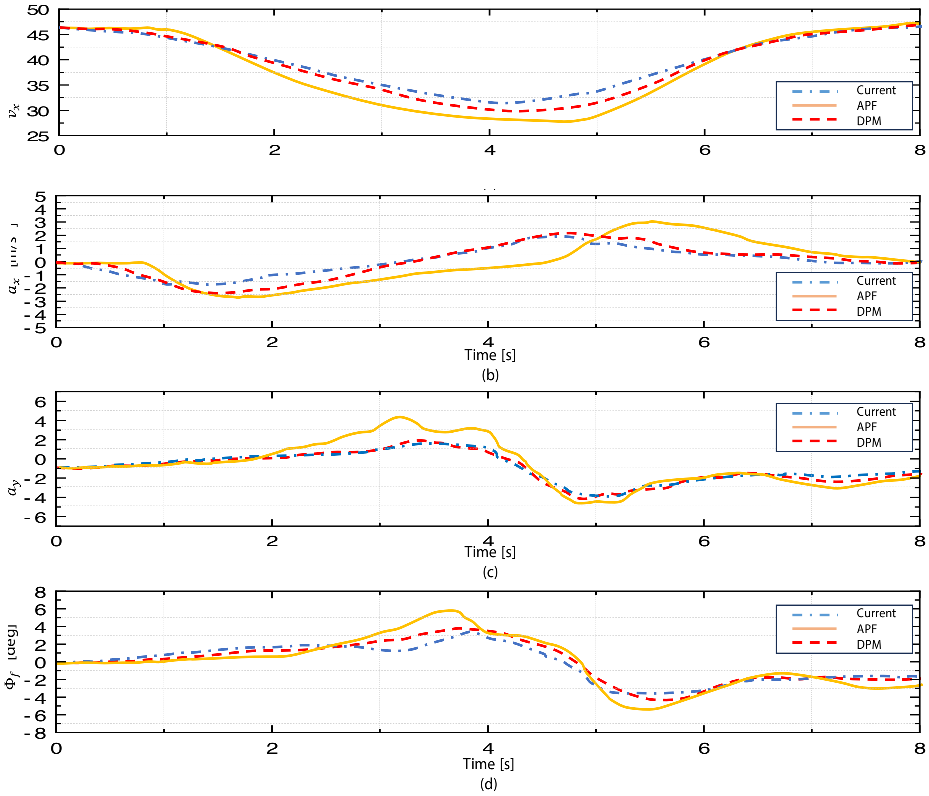

Figure 10 – Curve obstacle avoidance simulation diagram
(a) Longitudinal velocity v_x (b) Longitudinal acceleration a_x (c) Lateral acceleration a_y (d) Front wheel angle Φ_f

Scenario 2 aims to simulate more complex scenes in real life. In Scenario 2, the autonomous vehicle needs to complete the test of avoiding multiple obstacles in the curve. *Figure 8* shows that the algorithm proposed in this paper needs to select a safe path from nine paths after iteration. The black dotted line represents the path with the best adaptability, which means that after judging the cost function, it gets the highest evaluation. However, the algorithm recognises that most path selections bypass obstacles on the left side, so it adopts a safer path, i.e. a third path. This is to not to give the algorithm too much insurance for the pursuit of adaptability and cost, only to make the path planned by the algorithm more secure and reliable, and refuse to bring too much stimulation and discomfort to passengers. *Figure 9* shows the obstacle avoidance trajectories planned by the APF, the DPM and the algorithm proposed in this paper. It can be seen that the algorithm in this paper begins to perform obstacle avoidance action at a distance of 34 meters from the obstacle, while the DPM and the APF begin planning later. It can be seen from *Figure 10* that the proposed algorithm is significantly better than the APF algorithm in terms of longitudinal velocity, longitudinal acceleration, lateral acceleration and front wheel angle. Compared to the DPM, the new algorithm proposed in this paper starts obstacle avoidance earlier, so it is better than the DPM in longitudinal velocity and longitudinal acceleration. This proves that the trajectory planned by the proposed algorithm has better comfort and safety.

5.3 Dynamic obstacle detour

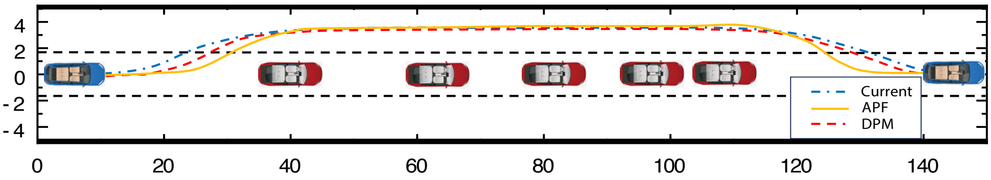

Figure 11 – Comparison of dynamic obstacle avoidance path planning

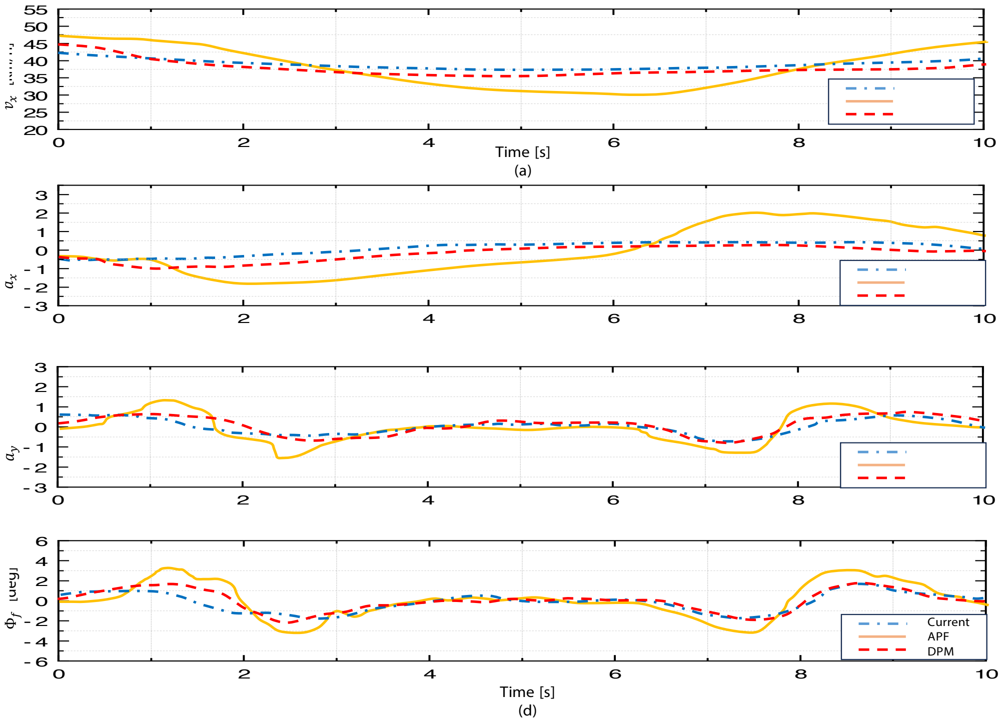

Figure 12 – Dynamic obstacle simulation diagram

(a) Longitudinal velocity v_x (b) Longitudinal acceleration a_x (c) Lateral acceleration a_y (d) Front wheel angle Φ_f

Scenario 3 tests the situation when the self-driving bike suddenly begins to slow down when it encounters the vehicle in front. Autonomous vehicles need to perform dynamic trajectory planning to bypass the decelerating obstacle vehicles. Moving obstacle vehicles will increase the time and distance required for planning. *Figure 11* shows the trajectories planned by the three algorithms. From the diagram, it can be seen that the algorithm proposed in this paper begins to deviate from the lane centre line at a distance of 32 meters from the obstacle, while the APF and the DPM begin to perform obstacle avoidance later. The late lane change time means that a larger steering is needed. This is not only detrimental to safety, but also causes greater lateral acceleration. As shown in *Figures 12b and 12d*, it can be seen that the proposed algorithm has the smallest change in the front wheel angle and lateral acceleration. In the whole obstacle avoidance process, the velocity difference between the DPM and the APF is 10km/h and 15km/h respectively, which exceeds the new algorithm proposed in this paper. Furthermore, the new algorithm is superior to the other two in the front wheel angle. Therefore, the trajectory planned by the new algorithm has better comfort and fluency than the APF and the DPM.

6. CONCLUSION

This paper proposes an obstacle avoidance trajectory planning strategy for autonomous vehicles based on genetic algorithm. The new strategy uses the genetic algorithm to quickly iterate to obtain a rough path and to complete the soft decision of the lane change direction. Based on the rough path, the convex space is opened up and the optimisation boundary is limited to reduce the amount of calculation and increase the real-time performance. The quadratic optimisation based on convex space will optimise the coarse solution path and generate a path that satisfies the constraint conditions. Finally, speed planning will give speed information to the path and generate a complete, comfortable and safe obstacle avoidance trajectory. Then, the real-time simulation verification of the new strategy is carried out on the HIL platform. Different from the two-lane road, the three-lane structured road is selected as the simulation experiment section. Scenarios 1 and 3 show the ability of the new strategy in obstacle avoidance trajectory planning on straight roads. Due to the existence of soft decision, and the fact that the cost of lane changing direction is consistent, it is assumed that the lane changing goes from the left side. Scenario 2 selects a three-lane curved road section with multiple obstacles as a simulation scene to simulate a complex scene in reality. Real-time simulation proves that the proposed new strategy will pay more attention to security rather than the cost. On the basis of ensuring sufficient safety, the planned trajectory has certain advantages over the APF and the DPM.

Future work will focus on the application of the new strategy in more complex scenarios, and it is planned to deploy it to real vehicles in experiments for wider verification and use.

ACKNOWLEDGMENTS

Funding: The authors acknowledge the support provided by the Natural Science Foundation of Hebei Province of China (No. E2016402066) and the High-Level Talent Project of Hebei Province of China: Integrated Research and Simulation Realisation of Vehicle Ride Comfort and Handling Stability (Grant No. B2017003026).

REFERENCES

[1] Paden B, et al. A survey of motion planning and control techniques for self-driving urban vehicles. *IEEE Transactions on Intelligent Vehicles*. 2016;1(1):33-55. DOI: 10.1109/TIV.2016.2578706.

[2] Vagale A, et al. Path planning and collision avoidance for autonomous surface vehicles I: A review. *Journal of Marine Science and Technology*. 2021;26(4):1292-1306. DOI: 10.1007/s00773-020-00787-6.

[3] Jeon G. A literature review of the path planning problem for unmanned aerial vehicle. *Journal of the Military Operations Research Society of Korea.* 2021;37(2):115-128.

[4] Chen ZZ, Xiao JZ, Wang GF. An effective path planning of intelligent mobile robot using improved genetic algorithm. *Wireless Communications & Mobile Computing*. 2022. DOI: 10.1155/2022/9590367.

[5] Maw AA, et al. IADA*: Improved anytime path planning and replanning algorithm for autonomous vehicle. *Journal of Intelligent and Robotic Systems*. 2020;100(3-4):1391-1405. DOI: 10.1177/09544070221088364.

[6] Yao TT, et al. Review of path planning for autonomous underwater vehicles. *The 2019 International Conference On Robotics, Intelligent Control And Artificial Intelligence*. 2019;482-487. DOI: 10.1145/3366194.3366280.

[7] Qin H, et al. Review of autonomous path planning algorithms for mobile robots. *Drones*. 2023;7(3). DOI: 10.3390/drones7030211.

[8] Lee SH, Chung CC. Autonomous-driving vehicle control with composite velocity profile planning. *IEEE Transactions on Control Systems Technology.* 2021;29(5):2079-2091. DOI: 10.1109/TCST.2020.3029713.

[9] Wang M, et al. Velocity planning method base on fuzzy neural network for autonomous vehicle. *IEEE Access.* 2021;9:19111-19126. DOI: 10.1109/ACCESS.2021.3054124.

[10] Wang M, et al. Improved genetic algorithms based on data-driven operators for path planning of unmanned surface vehicle. *International Journal of Robotics & Automation.* 2019;34(6):713-722. DOI: 10.2316/J.2019.206-0315.

[11] Maw AA, et al. IADA*: Improved anytime path planning and replanning algorithm for autonomous vehicle. *Journal of Intelligent and Robotic Systems*. 2020;100(3-4):1391-1405. DOI: 10.1177/09544070221088364.

[12] Gugan G, Haque A. Path planning for autonomous drones: Challenges and future directions. *DRONES.* 2023;7(3):169. DOI: 10.3390/drones7030169.

[13] Sun LY, et al. APF-bug-based intelligent path planning for autonomous vehicle with high precision in complex environment. *International Journal of Robotics & Automation.* 2023;38(4):277-283. DOI: 10.2316/J.2023.206-0741.

[14] Zhang YJ, et al. Optimal vehicle path planning using quadratic optimization for Baidu Apollo open platform. *2020 IEEE Intelligent Vehicles Symposium (IV)*. 2021;18(5). DOI: 10.1177/17298814211042730.

[15] Wu HX, et al. Research on vehicle obstacle avoidance path planning based on APF-PSO. *Proceedings of the Institution of Mechanical Engineers Part D-Journal of Automobile Engineering.* 2022;237(6):1391-1405. DOI: 10.1177/09544070221088364.

[16] Ji J, et al. Path planning and tracking for vehicle collision avoidance based on model predictive control with multiconstraints. *IEEE Transactions on Vehicular Technology*. 2017;66(2):952-964. DOI: 10.1109/TVT.2016.2555853.

[17] Duan YJ, et al. Active obstacle avoidance method of autonomous vehicle based on improved artificial potential field. *International Journal of Advanced Robotic Systems.* 2022;19(4). DOI: 10.1177/17298806221115984.

[18] Zheng L, et al. Bezier curve-based trajectory planning for autonomous vehicles with collision avoidance. *IET Intelligent Transport Systems.* 2020;14(13):1882-1893. DOI: 10.1049/iet-its.2020.0355.

[19] Yang WL, et al. Improved artificial potential field and dynamic window method for amphibious robot fish path planning. *Applied Sciences-Basel.* 2021;11(5). DOI: 10.3390/app11052114.

[20] Borges Lopes R, et al. A simple and effective evolutionary algorithm for the capacitated location–routing problem. *Computers & Operations Research*. 70(2016):155–162.

[21] Liu J, Lu Cao, Sun H. Improved genetic algorithm for fast path planning of USV. *MIPPR 2015: Remote Sensing Image Processing, Geographic Information Systems, and Other Applications*. 2015; 9815:981529. DOI: 10.1117/12.2210736.

[22] Chen ZZ, Xiao JZ, Wang GF. An effective path planning of intelligent mobile robot using improved genetic algorithm, *Wireless Communications and Mobile Computing*, 2022;8. DOI: 10.1155/2022/9590367.

[23] Hou WB, et al. Enhanced ant colony algorithm with communication mechanism for mobile robot path planning. *Robotics and Autonomous Systems*, 2022;148:103949. DOI: 10.1016/j.robot.2021.103949.

[24] Liu JY, Wei XX, Huang HJ. An improved grey wolf optimization algorithm and its application in path planning. *IEEE Access*, 2021;9:121944 –121956. DOI: 10.1109/ACCESS.2021.3108973.

[25] Kumar S, Sikander A. Optimum mobile robot path planning using improved artificial bee colony algorithm and evolutionary programming. *Arabian Journal for Science and Engineering*, 2022;47(3):3519-3539. DOI: 10.1007/s13369-021-06326-8.

[26] Xu Xing, et al. Global path planning for mobile robots based on improved genetic algorithm. *Computer Integrated Manufacturing System*, 2022;28(6):1659-1672.

[27] Orozco-Rosas U, et al. Mobile robot path planning using membrane evolutionary artificial potential field. *Applied Soft Computing.* 2019;77:235-251. DOI: 10.1016/j.asoc.2019.01.036.

[28] Zhu WW, et al. Local motion planning and tracking on autonomous driving vehicles. *IEEE Chinese Automation Congress (CAC 2020).* 2020;7679-7684. DOI: 10.1109/CAC51589.2020.9327148.

[29] Su SC, et al. Collaborative motion planning based on the improved ant colony algorithm for multiple autonomous vehicles. *IEEE Transactions on Intelligent Transportation Systems.* 2023. DOI: 10.1109/TITS.2023.3250756.

[30] Shang EK, et al. An improved A-Star based path planning algorithm for autonomous land vehicles. *International Journal of Advanced Robotic Systems.* 2020;17(5). DOI: 10.1177/1729881420962263.

[31] Zhang J, et al. Autonomous land vehicle path planning algorithm based on improved heuristic function of A-Star.

International Journal of Advanced Robotic Systems. 2021;18(5). DOI: 10.1177/17298814211042730.

[32] Zhu S, Aksun-Guvenc B. Trajectory planning of autonomous vehicles based on parameterized control optimization in dynamic on-road environments. *Journal of Intelligent & Robotic Systems.* 100(2020):1055-1067. DOI: 10.1007/s10846-020-01215-y.

翁晓风 刘飞 周盛 买嘉诚 冯少祥

基于遗传算法的自动驾驶汽车避障路径规划策略

摘要:

为了增强自动驾驶汽车在结构化道路上的驾驶能力，使其能够规划安全舒适的路径，我们提出了一种基于遗传算法的自动驾驶汽车避障路径策略。Frenet-Serret的使用增强了算法在复杂环境中的适应性。为了改进避障轨迹的生成和优化，我们建立了一个防撞模型。当车辆面临与障碍物的潜在碰撞时，遗传算法快速迭代并选择前九个基因来生成路径的粗略解和凸空间。结合凸空间，二次规划方法将对生成的粗解进行数值优化，以生成满足约束的精确路径。此外，为了确保避障过程中的安全性和舒适性，基于车辆的动态约束，采用速度规划来确定速度曲线。我们在涉及移动障碍物的各种场景中进行模拟。基于HIL平台的实时仿真证明，所提出的路径规划策略在各种驾驶场景下都是有效的。

关键字:

自动驾驶汽车；遗传算法；防撞模型；路径规划；序列二次规划