Dean&Francis

ISSN 2959-6157

Research on trajectory planning based on optimal control modeling

Guanxi Zhao

Chongqing University,Chongqing,400044,China;
mezhao2003@163.com

Abstract:

With the advancement of autonomous driving technology, trajectory planning has emerged as one of the core technologies to achieve safe and efficient vehicle driving. An algorithm for single-vehicle trajectory planning based on the Optimal Control Problem (OCP) is proposed in this paper, which can not only plan a safe and efficient path for autonomous vehicles, but also realize the obstacle avoidance function. Firstly, the OCP is modeled for the cycle trajectory planning problem. Secondly, this paper transforms it into a Nonlinear Programming (NLP) with all discrete methods, and uses the interior point method to solve it. Then, the Y-type intersection is selected as the verification scene, and through a series of simulation experiments, the efficiency and safety of the algorithm in different driving environments are verified. The model and algorithm proposed in this paper are not only applicable to Y-type intersections, but also can be extended to other complex traffic roads, providing a new idea and method for trajectory planning in intelligent transportation systems.

Keywords: Optimal Control Problem; Obstacle Avoidance; Single Vehicle Trajectory Planning; Y-type Intersection;

1 Introduction

Autonomous driving technology has gradually become the forefront of modern transportation development. Autonomous driving is not only expected to improve traffic efficiency and reduce traffic accidents, but also significantly improve traffic congestion and energy consumption. At present, major automakers and technology companies have invested significant resources and research and development efforts to achieve a fully automated driving environment in the future. In the autonomous driving system, trajectory planning is a critical technology to achieve safe and efficient driving of vehicles. It involves calculating the trajectory of the vehicle from the origin position to the target position, ensuring that the vehicle can avoid obstacles, and enabling it to travel safely in a variety of traffic environments. Efficient trajectory planning algorithms can improve the reaction speed and decision-making ability of autonomous vehicles, so that they can better cope with complex traffic conditions.

In the increasingly complex intelligent transportation system, multi-vehicle cooperative trajectory planning will become particularly important. In multi-vehicle collaborative trajectory planning, multiple autonomous vehicles need to share information in real-time and work out driving strategies together to achieve the optimal state of the overall traffic system. Multi-vehicle collaborative trajectory planning not only further enhances traffic efficiency, but also reduces the incidence of traffic accidents. As the basis of multi-vehicle collaborative trajectory planning, single-vehicle automatic driving is regarded as an important breakthrough point to solve the problem of multi-vehicle collaborative trajectory planning.

In the context of multi-vehicle collaborative trajectory planning, a number of studies have proposed methods to deal with partial actuator performance loss and actuator bias fault, with the goal of achieving synchronization and tracking control in multi-agent systems. Chen and Song (2015) addressed this challenge by designing a robust fault-tolerant cooperative control method[1].

Traditional traffic lights are less efficient in handling high-volume traffic, so they need to be improved. Shi et al. (2016) proposed a real-time vehicle scheduling algorithm based on intelligent traffic flow to improve the efficiency of intersections[2]. In addition, intersections are a key factor in the number of collisions and traffic delays within urban areas[3]. Therefore, researchers are focusing on how to improve traffic efficiency and reduce traffic congestion by improving the management of intersections.

At present, various methods have been proposed to control the fixed trajectory planning of autonomous vehicles in complex environments. These methods include Cooperative Adaptive Cruise Control (CACC), which allows vehicles to maintain a small head distance between each other and safely navigate a convoy at a coordinated speed[4]. However, how to properly dispatch vehicles in a fleet and improve the stability of the system remains a challenge.

With scientific and industrial advances in the field of autonomous driving, vehicles are able to communicate with each other and interact with a variety of infrastructure[5]. The multi-agent management system, which comprises Vehicle Agents (VAs) and an Intersection Agent (IA), schedules bookings in parallel through time and space and informs the vehicles of the results of the booking so that individual vehicle agents can adjust their speed as efficiently as possible, thereby improving the traffic efficiency of the intersection[6]. In addition, to improve the efficiency of intersection management, a new method based on fleet management has been developed, which allows for the presence of only one fleet at a time within the conflict area, thus improving the efficiency of any intersection strategy[7]. In urban environments, a centralized intersection control strategy based on Mixed Integer Linear Programming (MILP) is proposed to minimize delays for highly automated vehicles approaching intersections[8]. The majority of traffic control research has concentrated on lane-based traffic systems. However, the advent of the CAV (Connected and Autonomous Vehicles) era opens up possibilities for autonomous driving of vehicles in laneless environments. The lane-free design provides greater flexibility and adaptability for autonomous driving, while reducing costs and improving the efficiency of road use[9]. In addition, a novel collaborative control method based on maximum plus potential field is proposed, which enables vehicles to achieve near expected high speeds in complex environments with high traffic volume.

Compared to traditional intersections, signal-free and lane-free intersections allow vehicles to move more flexibly, thereby increasing capacity and reducing congestion. In order to maintain the integrity of the original task to the maximum extent, a parallel computing framework is proposed for fault tolerant collaborative motion planning of multi-network autonomous vehicles at non-signal and non-lane intersections[10].

At present, the research to solve the problem of multi-vehicle collaborative trajectory planning mainly focuses on two directions: one is the fixed trajectory planning method. Avoid conflicts by pre-setting the trajectories of each vehicle, ensuring that vehicles can safely pass through the intersection without colliding. The advantage of the algorithm is that the calculation is small and the system behavior is stable and predictable. In fixed trajectory planning, the trajectory of each vehicle has been planned in advance, and the algorithm only needs to adjust the speed of each vehicle to avoid collision. Its disadvantage lies in the lack of flexibility and low space utilization efficiency, the fixed trajectory limits the mobility of the vehicle, and may cause large-scale traffic congestion when encountering emergencies or obstacles in a complex environment, which greatly reduces the traffic efficiency. The second is the free-space trajectory planning method. The trajectory planning method that regards the road as a free space allows the vehicle to adjust the path in real-time according to the environment, which can better cope with the dynamic and complex environment and ensure more efficient path selection. Its advantages lie in high flexibility and high efficiency. Its disadvantage is that it has high computational complexity, and it requires a lot of computing resources to plan the direction and speed, which faces stability and security challenges in the actual driving process.

Based on this, a trajectory planning algorithm for Y-shaped intersections based on computational optimal control problems is proposed by this paper, which aims to solve the optimal trajectory planning problem of a single vehicle in complex traffic scenarios. Through OCP modeling of trajectory planning problem, combined with all discrete method to transform it into Nonlinear Programming problem, and using Interior Point Method to solve it, a safe and efficient driving path with obstacle avoidance function can be effectively planned for autonomous vehicles. In order to verify the feasibility and operation effect of the algorithm, this paper chooses Y-type intersection as the verification scene, and verifies the efficiency and safety of the algorithm in different driving environments through a series of simulation experiments. This not only lays a solid technical foundation for multi-vehicle collaboration, but also can further promote the development of intelligent transportation systems. Single-vehicle trajectory planning will directly affect the effect of multi-vehicle collaboration, so it is very important to improve the accuracy and efficiency of single-vehicle trajectory planning to realize the global optimal problem of multi-vehicle collaboration.

2 Modeling of optimal control problem

Trajectory planning problem is essentially a process from a known starting point to a predetermined ending point, and requires the solution of an optimal trajectory satisfying certain conditions. In this process, the core task is to find a set of control strategies that enable the vehicle to move along this trajectory. Therefore, the theory of Optimal Control Problem is very suitable for this kind of modeling. The key of OCP is to optimize the control strategy, so that the system from the initial state to the final state, and at the same time to achieve the best performance index. Through OCP modeling, the trajectory planning problem can be transformed into an optimization problem, precisely defining the optimal path of the vehicle under given constraints. This not only ensures that the planned trajectory achieves the best results in terms of safety, smoothness and efficiency, but also allows flexibility to adapt to vehicle dynamics and complex traffic scenarios. Therefore, this paper chooses the OCP to model the trajectory planning problem of a bicycle in order to achieve the optimal trajectory planning in a complex environment.

2.1 Kinematic constraints

In a practical situation the car can be regarded as a system with an infinite number of degrees of freedom, but it is impractical to build such a system in an experiment. As long as the dynamic constraints of the vehicle can be reflected in the trajectory planning process, the controller can achieve the predetermined control objectives. Therefore, the bicycle kinematics model adopted in this paper is a hypothesis and simplification for the actual situation, as shown in Fig. 1.

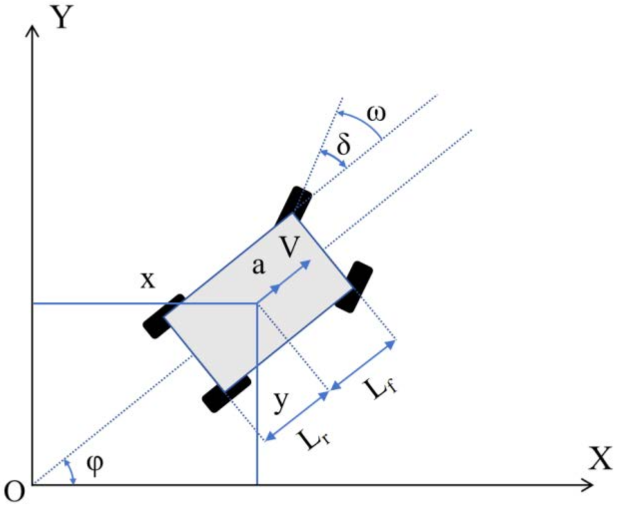

Fig. 1. Bicycle kinematics model of vehicle

The vehicle is regarded as a rigid body, and the influence of roll and pitch, aerodynamics and road roughness on vehicle change is ignored. XOY is the environment coordinate system, and the constraints of the bicycle kinematics model are:

$$\begin{aligned}
&\frac{d}{dt}x(t)=cos\varphi(t)*v(t),t\in[0,T]\\
&\frac{d}{dt}y(t)=sin\varphi(t)*v(t),t\in[0,T]\\
&\frac{d}{dt}v(t)=a(t),t\in[0,T]\\
&\frac{d}{dt}\delta(t)=\omega(t),t\in[0,T]\\
&\frac{d}{dt}\varphi(t)=tan\delta(t)/(L_r+L_f)*v(t),t\in[0,T]
\end{aligned}\tag{1}$$

Where x and y are the lateral displacement and longitudinal displacement of the vehicle respectively; v is the speed of the car, a is the acceleration of the car; δ is the direction angle of the car, ω is the yaw angle speed, φ is the front wheel angle; Respectively, L_r and L_f is the distance from the car's center of mass to the front and rear axes respectively, and all variables are functions of time t. In addition, according to the actual situation, the inequality constraints related to the maximum and minimum values of some variables are applied in this paper. Position constraints, velocity constraints, acceleration constraints, front wheel angle constraints, orientation angle constraints, and yaw angle velocity constraints are shown as follows:

$$\begin{aligned}
&x_{min}\leqslant x_t\leqslant x_{max},y_{min}\leqslant y_t\leqslant y_{max}\\
&v_{min}\leqslant v_t\leqslant v_{max}\\
&a_{min}\leqslant a_t\leqslant a_{max}\\
&\delta_{min}\leqslant \delta_t\leqslant \delta_{max}\\
&\varphi_{min}\leqslant \varphi_t\leqslant \varphi_{max}\\
&\omega_{min}\leqslant \omega_t\leqslant \omega_{max}
\end{aligned}\tag{2}$$

The maximum and minimum values of each constraint are set in different specific simulation scenario.

2.2 Two-point Constraints

When considering dynamic constraints and control requirements, clear constraints on the starting point and ending point can ensure that the initial and end conditions of trajectory planning are consistent with the actual application scenarios, which is the basis for ensuring the feasibility of the planned path.

$$x_{t=0}=x_{start},y_{t=0}=y_{start}\tag{3}$$

$$x_{t=N}=x_{goal},y_{t=N}=y_{goal}\tag{4}$$

Where $x_{t=0}$ and $y_{t=0}$ are the initial state, x_{start} and y_{start} are the starting point condition set; $x_{t=N}$ and $y_{t=N}$ are the final state, and x_{goal}, y_{goal} are the set end condition. In the fourth chapter of this paper, multiple different points with the same displacement will be set for solving and comparing.

2.3 Obstacle detection constraint

In trajectory planning, obstacle detection is the key link to ensure vehicle safety. In order to avoid collisions, obstacles must be detected and avoided in real-time during trajectory planning. Let the whole free space environment be M, the occupied space of the vehicle be the set Γ_1, and the position P of the vehicle is in the state $P \in \Gamma_1$ at all times during the trajectory planning process; The space occupied by the obstacle is Γ_2. In order to ensure safe driving, the occupied space Γ_1 of the vehicle at any time cannot be intersected with the occupied space of the obstacle, that is:

$$\Gamma_1 \cap \Gamma_2 = \varnothing, \forall t \in [t_0, t_f] \tag{5}$$

where t_0 is the initial moment, t_f is the final moment. The establishment of the above formula means that the vehicle must always remain in the free space M during the entire movement, while avoiding overlap with the obstacle space Γ_2.

2.4 Objective function

In the trajectory planning problem, the objective function is used to guide the path optimization in the control algorithm, and its design will directly affect the final characteristics of the path. Generally, the objective function is to optimize the motion performance of the vehicle. In this paper, the smoothness of the driving trajectory and the driving comfort and safety are improved by minimizing the acceleration and steering angle.

$$Cost = \int_{t_0}^{t_f} a(t)^2 + \int_{t_0}^{t_f} \delta(t)^2 \tag{6}$$

Where cost is the name of the objective function and $a(t)$ is the acceleration a at each moment from the initial time t_0 to the final time t_f. The integral of the square of this item measures the acceleration on the entire trajectory. Optimizing this item is helpful to reduce the sharp acceleration or deceleration of the vehicle in automatic driving and improve the comfort and safety during driving. $\delta(t)$ is the front wheel angle δ at each moment from the initial time t_0 to the final time t_f The integral of the square of this term measures the change of steering angle along the entire path. Optimizing this term is helpful to reduce sharp turns and improve the smoothness and safety of the trajectory.

In single-vehicle trajectory planning, OCP can precisely define the optimal trajectory of a vehicle under given constraints, and it is an ideal method to solve these problems. The optimal control problem after transformation is shown as follows:

$$minimizecost = \int_{t_0}^{t_f} a(t)^2 + \int_{t_0}^{t_f} \delta(t)^2$$

$$subjectto: \frac{d}{dt} x(t) = cos\varphi(t) * v(t), t \in [0,T]$$

$$\frac{d}{dt} y(t) = sin\varphi(t) * v(t), t \in [0,T]$$

$$\frac{d}{dt} v(t) = a(t), t \in [0,T] \tag{7}$$

$$\frac{d}{dt} \delta(t) = \omega(t), t \in [0,T]$$

$$\frac{d}{dt} \varphi(t) = tan\delta(t) / (L_r + L_f) * v(t), t \in [0,T]$$

$$x_{min} \leqslant x_t \leqslant x_{max}, y_{min} \leqslant y_t \leqslant y_{max}$$

$$v_{min} \leqslant v_t \leqslant v_{max}$$

$$a_{min} \leqslant a_t \leqslant a_{max}$$

$$\delta_{min} \leqslant \delta_t \leqslant \delta_{max}$$

$$\varphi_{min} \leqslant \varphi_t \leqslant \varphi_{max}$$

$$\omega_{min} \leqslant \omega_t \leqslant \omega_{max}$$

In the process of transforming trajectory planning into optimization problem through OCP modeling, various constraints such as bicycle kinematics constraints are required to simplify the vehicle kinematics model reasonably. Starting and ending constraints determine the initial and end conditions of trajectory planning, enabling vehicles to accurately complete trajectory planning in complex environments. Obstacle detection constraints serve as computational constraints on the distance between vehicles and obstacles, avoiding collision risks, and laying a foundation for safe, stable and efficient trajectory planning.

3 Solving of OCP

3.1 OCP solving methods and classification

The core goal of solving OCP is to find a set of control variables that can drive the system from the initial state to the final state optimally. The solving methods of OCP are mainly divided into indirect method and direct method. The optimal control problem could be transformed into a two-point Boundary Value Problem (BVP) by the indirect method by deriving the necessary conditions for Pontryagin's Maximum Principle. For small-scale problems or situations with clear analytical solutions, indirect methods can provide accurate and efficient solutions. However, its disadvantage is that for nonlinear or high-dimensional problems, solving two-point boundary value problems is more complicated, and it is difficult to apply in large-scale complex systems. In contrast, the direct rule turns the optimal control problem into a Nonlinear Programming (NLP) problem by discretizing this problem, which is then solved by parameter optimization. The direct method can deal with all kinds of complex objective functions and constraints flexibly, and is especially suitable for large-scale, nonlinear and multi-constraint optimization problems. According to the different discretization methods, direct method is further divided into partial discretization method and total discretization method. The partial discretization method only discretizes the control variables, while the state variables remain as continuous variables, and the control strategy is solved through the optimization process. The total discretization law discretized both the control variables and the state variables at the same time, and completely transformed the continuous time OCP problem into a finite dimension NLP. Due to the complex situation of trajectory planning problem, it needs to deal with a large number of state variables and control variables, and the total discrete method can effectively deal with this large-scale problem.

3.2 Principle of direct method

In the direct method, the OCP first transforms the continuous time domain control problem into a finite step optimization problem by discretization process. The specific discretization process is: the time interval $[t_0, t_f]$ is discretized into N discrete points $t_0, t_1, ..., t_N$, set the time step to Δt. The displacement variables x(t) and y(t) are discretized to x_t and y_t at each time point, the velocity variable v(t) is discretized to v_t, the acceleration variable a(t) is discretized to a_t, the orientation angle variable δ(t) is discretized to δ_t, and the yaw angle velocity variable $\omega(t)$ is discretized to ω_t. The front wheel angle $\varphi(t)$ is discretized into φ_t, where $t \in t_0, t_1, ..., t_N$, denotes the time node after discretization.

The bicycle kinematic constraints are discretized accordingly.

$$\begin{aligned}
x_{t+1} - x_t &= v_t * cos\varphi_t * \Delta t, \forall t \in t_0, t_1, ..., t_{N-1} \\
y_{t+1} - y_t &= v_t * sin\varphi_t * \Delta t, \forall t \in t_0, t_1, ..., t_{N-1} \\
v_{t+1} - v_t &= a_t * \Delta t, \forall t \in t_0, t_1, ..., t_{N-1} \\
\delta_{t+1} - \delta_t &= \omega(t) * \Delta t, \forall t \in t_0, t_1, ..., t_{N-1} \\
\varphi_{t+1} - \varphi_t &= v_t * tan\delta_t / (L_r + L_f) * \Delta t, \forall t \in t_0, t_1, ..., t_{N-1}
\end{aligned} \tag{8}$$

The two-point constraints on the starting and ending points of each variable in the above formula remain unchanged.

The objective function in the original OCP is an integral form, and after discretization, the objective function is transformed from an integral form about the variables a and δ into a weighted sum of the variables at discrete time points, as shown in the following formula:

$$Cost = \sum_{t=t_0}^{t_N} a_t^2 + \sum_{t=t_0}^{t_N} \delta_t^2 \tag{9}$$

3.3 Modeling of road vehicle obstacle constraints

In order to simplify the calculation and ensure absolute safety, the vehicle and the obstacle can be approximated as two circles. This approximation method is often used in trajectory planning and obstacle avoidance algorithms.

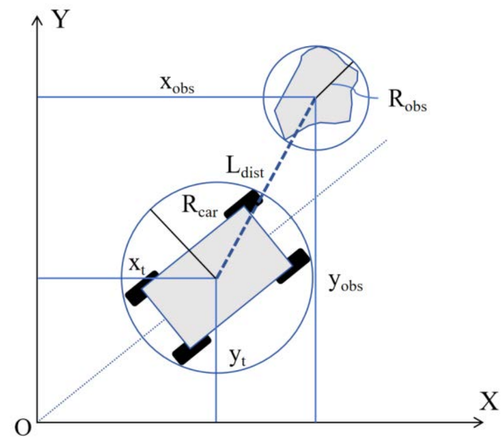

Fig. 2.Obstacle detection constraints

The constraints are as follows:

$$L_{dist} = \sqrt{(x_t - x_{obs})^2 + (y_t - y_{obs})^2}, t \in [0, N] \tag{10}$$

$$L_{dist} > R_{car} + R_{obs} \tag{11}$$

Where, x_t and y_t are the horizontal and vertical coordinate positions of the car under each time step, x_{obs} and y_{obs} are the horizontal and vertical coordinate positions of the obstacle respectively, L_{dist} is the geometric distance from the center of the vehicle to the center of the obstacle circle, R_{car} and R_{obs} are the radii of the vehicle and the obstacle respectively. After this constraint is established, L_{dist} can always be greater than the sum of the two radii to avoid collision.

3.4 Interior point method

After discretization, the OCP is transformed from a control optimization problem in continuous time domain to a NLP problem in discrete time domain.

$$\begin{aligned}
&minimizeCost = \sum_{t=t_0}^{t_N} a_t^2 + \sum_{t=t_0}^{t_N} \delta_t^2 \\
&subjectto: x_{t+1} - x_t = v_t * cos\varphi_t * \Delta t, \forall t \in t_0, t_1, ..., t_{N-1} \\
&y_{t+1} - y_t = v_t * sin\varphi_t * \Delta t, \forall t \in t_0, t_1, ..., t_{N-1} \\
&v_{t+1} - v_t = a_t * \Delta t, \forall t \in t_0, t_1, ..., t_{N-1}
\end{aligned} \tag{12}$$

$$\delta_{t+1} - \delta_t = \omega(t) * \Delta t, \forall t \in t_0, t_1, ..., t_{N-1}$$
$$\varphi_{t+1} - \varphi_t = v_t * tan\delta_t / (L_r + L_f) * \Delta t, \forall t \in t_0, t_1, ..., t_{N-1}$$
$$L_{dist} = \sqrt{(x_t - x_{obs})^2 + (y_t - y_{obs})^2}, \ t \in [0, N]$$
$$x_{min} \leqslant x_t \leqslant x_{max}, y_{min} \leqslant y_t \leqslant y_{max}$$
$$v_{min} \leqslant v_t \leqslant v_{max}$$
$$a_{min} \leqslant a_t \leqslant a_{max}$$
$$\delta_{min} \leqslant \delta_t \leqslant \delta_{max}$$
$$\varphi_{min} \leqslant \varphi_t \leqslant \varphi_{max}$$
$$\omega_{min} \leqslant \omega_t \leqslant \omega_{max}$$
$$L_{dist} > R_{car} + R_{obs}$$

The Internal Point Method is employed to solve the NLP problem. Interior Point Method (Interior Point Method) is an optimization algorithm that suits for solving NLP problems, especially for large scale problems with inequality constraints. Unlike the traditional Simplex Method, the interior point method which searches for the optimal solution along the boundary of the feasible region, the Interior Point Method iteratively approaches the optimal solution from within the feasible domain, hence the name "interior point method". This method is renowned for its high efficiency in dealing with large-scale, nonlinear and multi-constraint problems, making it a popular choice for solving NLP problems obtained after discretization of OCP.

4 Scenario simulation and problem solving

4.1 Scene modeling and parameters

4.1.1 Y-type intersection, length, Angle

Y-intersection is a common scene in traffic planning, it is usually composed of a main road and two branch roads, the three roads meet at the intersection. In order to carry out accurate trajectory planning and optimal control of vehicle movement in such complex scenes, the geometric characteristics of Y-intersection are accurately modeled in this paper. The length L of each road is set to 10 meters, the width W is set to 6 meters, the angle θ between the roads is 120°, and the position of the central intersection is set to the origin (0, 0).

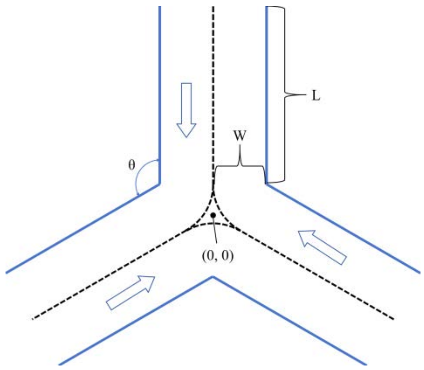

Fig. 3. Scene diagram of Y-type intersection

4.1.2 Commuter planning diagram

In the traditional traffic planning, traffic light control is a common way of intersection management. However, the traditional traffic light control methods often lead to inefficiency during peak hours, increasing the waiting time for vehicles, which in turn causes traffic congestion. In order to improve commuting efficiency, especially in complex traffic environments, more advanced scheduling and planning methods must be considered. In addition, in the actual traffic scenario, the vehicle not only needs to obey the traffic lights, but also needs to deal with various obstacles. In this situation, the fixed trajectory planning method relying solely on adjusting speed cannot effectively avoid these obstacles. Instead, it needs to be regarded as free space trajectory planning and combined with obstacle avoidance strategies for Optimal Control Problem solving, so as to ensure that vehicles can still safely and efficiently pass through intersections under the condition of obstacles.

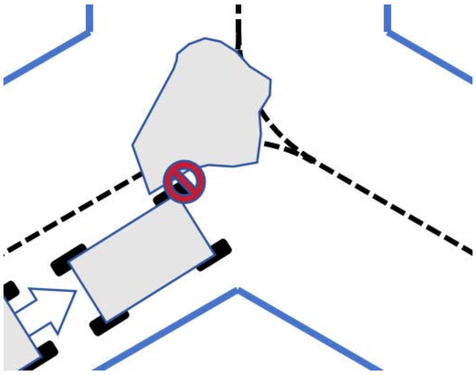

Fig. 4. diagram of obstacles encountered by the fixed trajectory planning method

4.2 Modeling and solving of OCP problem for Y-type intersection

4.2.1 Computer Configuration Parameters

This paper mainly simulates the trajectory planning problem of a bicycle passing through a Y-intersection. In order

to maximize the computational efficiency in the simulation solution process, the following configurations were adopted for the computer in this paper: (1) CPU: 13th Gen Intel(R) core(TM) i9-13900HX; (2) Memory: 32GB DDR5 RAM; (3) Operating system: Windows 11 64-bit. The computer uses the programming language Python and the CasADi toolkit for modeling and solving.
The above configuration can provide sufficient computational resources for solving Nonlinear Programming problems, and ensure that complex numerical computation tasks can be processed quickly and stably.

4.2.2 Parameter List

Tab. 1. Parameter List

Parameter Name	Symbol	Value/Range	Unit
Vehicle lateral position	x	$(-\infty, \infty)$	m
Vehicle longitudinal position	y	$(-\infty, \infty)$	m
Vehicle speed	v	(-20, 20)	m/s
Vehicle acceleration	a	(-5, 5)	m/s^2
Vehicle heading angle	δ	(-1, 1)	rad
Vehicle yaw rate	ω	$(-\frac{\pi}{9}, \frac{\pi}{9})$	rad
Vehicle front wheel steering angle	φ	(-π, π)	rad
Distance from vehicle's center of mass to front axle	L_f	1.5	m
Distance from vehicle's center of mass to rear axle	L_r	1.5	m
Equivalent circular radius of vehicle	R_{car}	1.5	m
Equivalent circular radius of obstacle	R_{obs}	1	m
Intersection length	L	20	m
Intersection width	W	6	m
Angle between intersection arms	θ	120	°
Time step	dt	0.01	s
Number of time steps	N	100	/

4.3 Result Analysis

4.3.1 Solution for normal passage without obstacles

This paper first simulates the situation of vehicles passing different trajectories in the Y-intersection under normal conditions. So as to calculate the optimal trajectory of the single vehicle from different starting points to the ending points, the initial solution needs to be set first. In this paper, the linear interpolation method is employed to generate the initial solution, which is universal and can adapt to different starting points and ending points. Through linear interpolation, a straight line from the starting point to the ending point is generated as the initial path, and on this basis, the optimization is carried out to ensure that the final path is smoother and more reasonable under the dynamic constraints.

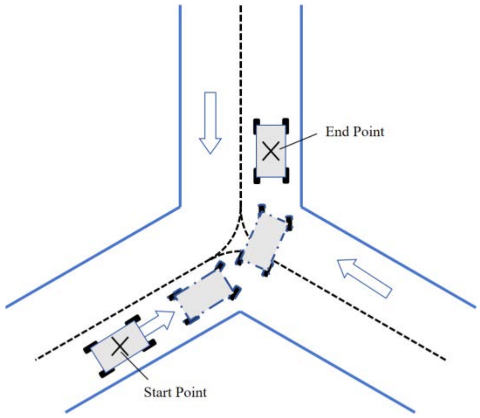

Fig. 5. Schematic drawing of simulation without obstacles

The simulation does not set the concept of a lane, thus allowing the vehicle to freely choose a path within the space to improve space utilization. In this paper, trajectory planning and calculation are carried out for the obstruction-free environment at first, and the simulation trajectory results are shown in Fig. 6. below.

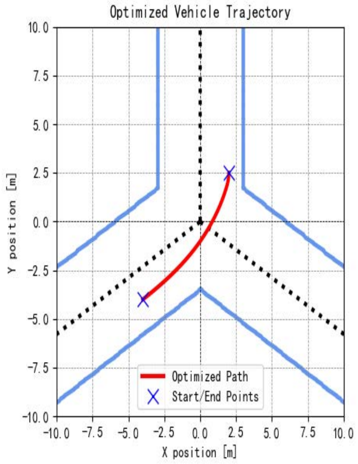

Fig. 6. Diagram of imulation trajectory planning without obstacles

4.3.2 Solution with Obstacles

In the presence of an obstacle, the vehicle's path needs to bypass the obstacle to avoid a collision. The initial solution is no longer simply set as a straight path from the starting point to the ending point, but the obstacle avoidance trajectory planning method is introduced to determine the boundary of the obstacle by treating the obstacle as a circular region. By selecting the appropriate obstacle avoidance points at the boundary of obstacles, these obstacle avoidance points are inserted between the starting point and the ending point, and a polygonal path is generated by using linear interpolation method. This polygonal path can ensure that the vehicle can bypass obstacles while approaching the optimal path as much as possible. Thus to achieve effective obstacle avoidance effect.

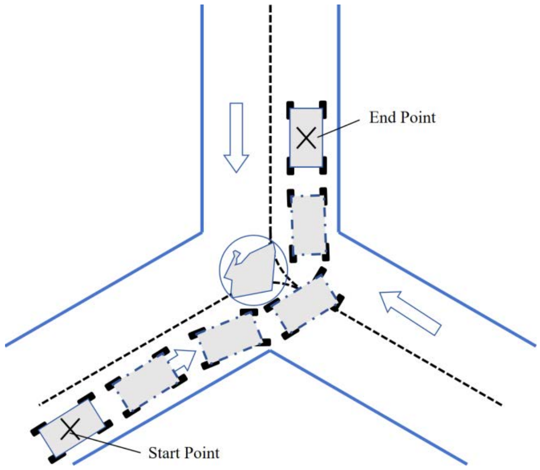

Fig. 7. Schematic drawing of obstacle avoidance simulation

In the process of simulation solution, this paper sets multiple starting points and corresponding ending points of vehicles in different directions, and ensures that the linear distance between the starting point and the ending point is equal. After 50 times of calculations, part of the trajectory obtained by simulation is shown in the figure below.

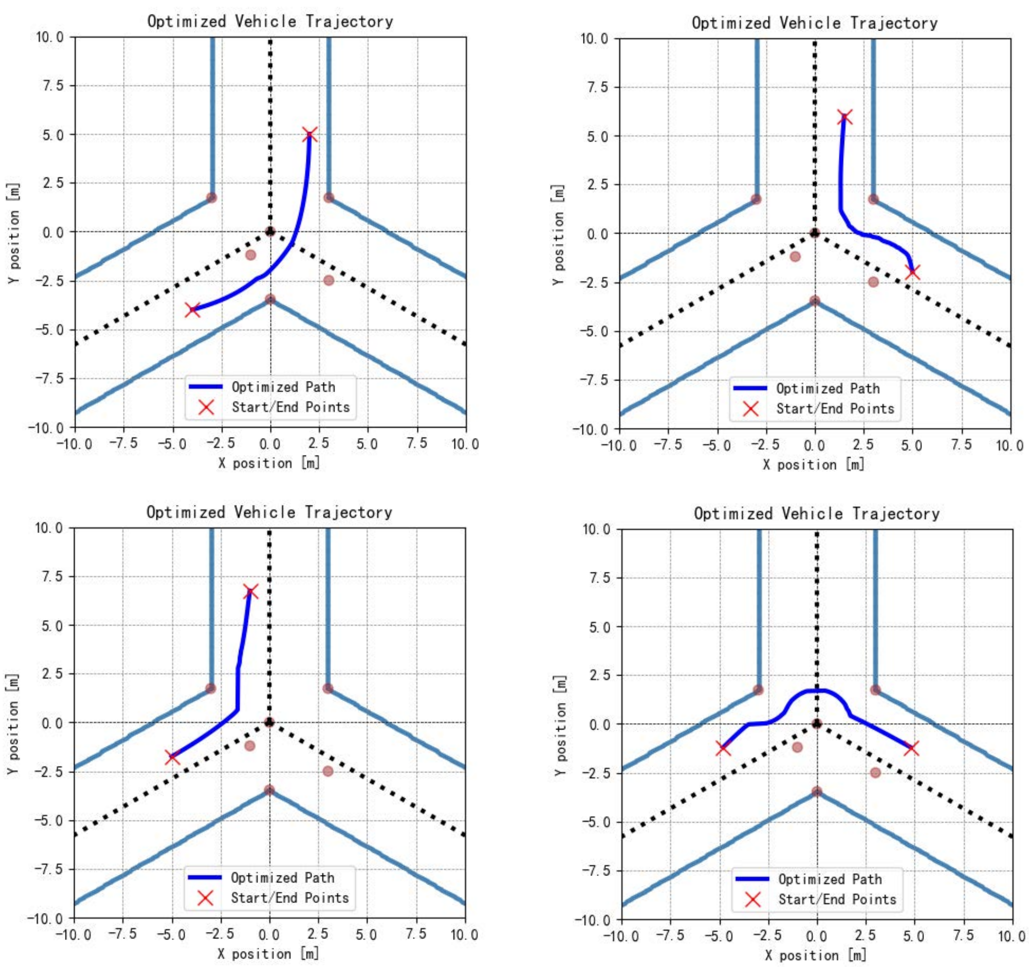

Fig. 8. Diagram of obstacle avoidance simulation trajectory planning

In addition, this paper also summarizes the maximum, minimum and average values of the simulation running time in the table. The statistics can help us to better understand the performance of the vehicle through different trajectories in the absence of obstacles.

Tab. 2. Simulation run time statistics table

Maximum Run Time (s)	Minimum Run Time (s)	Average Run Time (s)
2.572	1.869	2.034

5 Conclusion

This paper mainly studies the trajectory planning problem of bicycles passing Y-intersection, and proposes a trajectory planning algorithm based on computational optimal control modeling and obstacle avoidance function. Firstly, the Optimal Control Problem (OCP) is modeled, and then the Nonlinear Programming (NLP) problem is discretized by all discretization method in the direct method, and the problem is solved by interior point method. Through a series of simulation experiments, we verify the effectiveness of the algorithm under normal conditions and with obstacles. The optimized trajectory can satisfy the kinematic constraints of the vehicle, and at the same time, the vehicle can pass the intersection safely and smoothly with high operating efficiency. The maximum running time calculated by the trajectory planning experiment in the case of obstacles is 2.572 seconds. The minimum running time is 1.869 seconds; The average run time is 2.034 seconds. The experimental results show that the algorithm can quickly calculate and generate the trajectory planning scheme that meets the actual requirements, effectively avoid the inefficient blockage problem caused by the traffic light control in the traditional traffic, and solve the problem that the fixed trajectory planning encounters the lack of flexibility and low space utilization efficiency. In addition, the model and solution method proposed in this paper are not only applicable to Y-type intersections, but also can be extended to other complex traffic roads, such as intersections, etc., providing a new idea and method for trajectory planning in intelligent transportation systems. Through the reasonable OCP modeling and optimization algorithm, more efficient and safe vehicle traffic can be realized, which provides theoretical support for future traffic management and planning.

Reference

[1] Chen, G., and Song, Y.D. (2015). Robust fault-tolerant cooperative control of multi-agent systems: A constructive design method. Journal of the Franklin Institute, 352(10), 4045–4066

[2] M. Shi, H. Jiang and S. Li, “An intelligent traffic-flow-based real-time vehicles scheduling algorithm at intersection”, Proc. ICARCV, vol. 2016, pp. 1-5

[3] Y. Rahmati and A. Talebpour, “Towards a collaborative connected automated driving environment: A game theory based decision framework for unprotected left turn maneuvers”, Proc. IEEE IV Symp., pp. 1316-1321, Jun. 2017

[4] Dey, K. C., Yan, L., Wang, X., Wang, Y., Shen, H., Chowdhury, M., Yu, L., Qiu, C., & Soundararaj, V. (2016). A Review of Communication, Driver Characteristics, and Controls Aspects of Cooperative Adaptive Cruise Control (CACC). IEEE Transactions on Intelligent Transportation Systems, 17(2), 491–509

[5] Namazi, E., Li, J., & Lu, C. (2019). Intelligent Intersection Management Systems Considering Autonomous Vehicles: A Systematic Literature Review. IEEE Access, 7, 91946–91965

[6] Atakan Yasin Yesilyurt et al., “A Reservation Method for Multi-agent System Intersection Management with Energy Consumption Considerations”, IFAC PapersOnLine, 54-2 (2021), 246–251

[7] Masoud Bashiri and Cody H. Fleming, “A Platoon-Based Intersection Management System for Autonomous Vehicles”, 2017 IEEE Intelligent Vehicles Symposium (IV), June 11-14, 2017, Redondo Beach, CA, USA

[8] Eduardo Rauh Müller et al., “Intersection control for automated vehicles with MILP”, IFAC-PapersOnLine, 49-3 (2016), 037–042

[9] Dimitrios Troullinos, Georgios Chalkiadakis, Ioannis Papamichail, Markos Papageorgiou, “Collaborative Multiagent Decision Making for Lane-Free Autonomous Driving”, AAMAS 2021, May 3-7, 2021, Online

[10] Bai Li et al., “Fault-Tolerant Cooperative Motion Planning of Connected and Automated Vehicles at a Signal-Free and Lane-Free Intersection”, IFAC PapersOnLine, 51-24 (2018), 60–67