IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 18 December 2023, accepted 9 January 2024, date of publication 11 January 2024,
date of current version 18 January 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3352919

RESEARCH ARTICLE

On-Road Trajectory Planning of Connected and Automated Vehicles in Complex Traffic Settings: A Hierarchical Framework of Trajectory Refinement

FUZHOU ZHAO[1,2], LING HAN[3], MINGYANG CUI[2], HEYE HUANG[2], SHAN ZHONG[4], FEIFEI SU[5], AND LEI WANG[6]

[1]School of Automotive Engineering, Changshu Institute of Technology, Suzhou 215500, China
[2]The State Key Laboratory of Automotive Safety and Energy, Tsinghua University, Beijing 100084, China
[3]School of Mechanical and Electrical Engineering, Changchun University of Technology, Changchun 130012, China
[4]Changshu Institute of Technology, Suzhou 215500, China
[5]Baidu Technology (Beijing) Company Ltd., Beijing 100193, China
[6]Ziqing Intelligent Driving Technology (Beijing) Company Ltd., Beijing 100094, China

Corresponding author: Fuzhou Zhao (zhaofuzhou@cslg.edu.cn)

This work was supported by the Opening Fund of the State Key Laboratory of Automotive Safety and Energy under Grant KYF2213.

ABSTRACT This paper presents a hierarchical framework for on-road trajectory planning in complex traffic environments. Firstly, the processing of sparse coarse trajectories involves the utilization of DP (Dynamic Programming) generation and interpolation techniques. Then, for the waypoints with collision risk in the smoothed trajectory, the spiral search method is used to find some safe alternate waypoints. The alternate waypoints and the previous ones without collision risk form the amended trajectory. Concurrently, safety tunnels are constructed along the amended trajectory for the ego vehicle. Furthermore, with the constraint conditions of vehicle kinematics model and safety tunnels, nonlinear program (NLP) optimization is carried out for the amended trajectory of ego vehicle to obtain smooth and safe trajectories. For typical cases, simulation experiments demonstrate that the ego vehicle can ensure collision safety in dynamic traffic scenarios, while maintaining smooth vehicle velocity and small jitter of the front wheel angle. The proposed trajectory planning framework provides a novel decision-making method for connected and automated vehicles (CAVs).

INDEX TERMS Trajectory planning, nonlinear program optimization, connected and automated vehicles.

I. INTRODUCTION

The concept of automated vehicles has long captivated human beings for more convenient mobility and safer traffic experiences. At the very stroke of the DARPA Grand Challenge launched by the U.S. Department of Defense in 2004, the physical feasibility of automated vehicles reignited interest around the world [1]. Researchers hoped that automated vehicles could reduce traffic accidents and alleviate driver fatigue, and a lot of new technologies subsequently popped up on modules such as perception, localization, planning, and control. To further enhance safety, efficiency, convenience, and fuel consumption, the concept of connected and automated vehicles (CAVs) has been proposed, which leverages vehicle-to-everything (V2X) communication to improve trajectory planning [2]. Currently, both industry and academia continue to increase R&D investment and the related research is in full swing. The technology of CAVs is developing rapidly worldwide in recent years [3], [4], [5].

Methods for multi-vehicle coordination can effectively control traffic participants to improve traffic efficiency and ensure driving safety in various traffic scenarios with multi-lane roads. Existing research on multi-lane coordination of CAVs includes lane assignment at intersections, merges,

The associate editor coordinating the review of this manuscript and approving it for publication was Jianxiang Xi.

splits, and roundabouts, as well as multi-lane formation control with minimal following error and fluctuation [6], [7]. Coordinated control modes for CAVs encompass behaviors such as following, lane changing, and overtaking on roads with multiple lanes [8].

The ultimate goal of multi-vehicle coordinated control is to plan the trajectories of the relevant vehicles in a spatiotemporally efficient manner. Numerous models and strategies have been proposed to investigate CAV trajectory policies. However, trajectory planning on structured roads entails planning a feasible spatiotemporal channel while considering cooperative vehicles, pedestrians, road geographies, lanes, and traffic signals in complex driving environments.

For trajectory planning, sample-based methods randomly sample the state space to exhaustively explore possible models and filter out approximate optimal solutions [9]. The Rapidly-exploring Random Tree (RRT) is a classical trajectory planning algorithm based on incremental sampling [10]. However, the RRT algorithm is known for its meandering paths, inaccurate terminal states, and slow exploration, which renders it inefficient for automated vehicles despite various improvements [11], [12]. The RRT algorithm is more commonly used for trajectory planning in slow automated vehicles and indoor robots [13].

Search-based methods involve traversing a lattice state space to identify optimal waypoints, which typically include algorithms such as A*, Dijkstra, and DP. The A* algorithm, which uses a heuristic function to search for optimal paths in connected graphs, can obtain the node sequence connecting the initial and final nodes. However, the A* algorithm is characterized by drawbacks such as getting stuck in dead ends, high memory consumption, and no guarantee of trajectory planning quality [14], [15]. The Dijkstra algorithm, a heuristic search algorithm for finding the single-source shortest path, can be viewed as a simplified version of the A$*$ algorithm. However, it suffers from similar issues of low efficiency, occasional lack of optimal solutions, and high memory overhead [16], [17]. Although the DP algorithm is effective for solving multi-layer planning problems, it has some limitations. As the dimension of the variable increases, the total computation and storage requirements increase sharply, and it may even suffer from the curse of dimensionality [18], [19]. The parallel architecture of GPU hardware can be leveraged to accelerate trajectory planning for CAV formation using the DP algorithm [20].

The trajectories obtained through the search-based method are discrete states and do not consider the intermediate waypoints between the traversed points. Furthermore, enhancing decision-making efficiency and reducing storage overhead are additional considerations for sparse trajectory processing. In recent studies on the formation control of Connected and Autonomous Vehicles (CAVS) [21], traffic flow control at no-signal intersections [22], and traffic flow control at split or merge lanes [23], [24], coarse trajectory planning has been commonly employed for various purposes. When linearly connecting coarse trajectories, the vehicle displacement remains continuous but often lacks smoothness, resulting in discontinuous velocities for all trajectory states. This linear connection may exceed the tracking capabilities of the physical vehicle. Nonlinear interpolation serves as a refinement approach for coarse trajectories. Effective examples of nonlinear interpolation include the transfer regression model or B-spline curve interpolation for rough trajectories. By utilizing these methods, accurate prediction results can be obtained, significantly reducing the workload of the trajectory planner [25], [26]. In addition, Gaussian process regression is introduced to smooth the generated multi vehicle interaction trajectories [27], and generative adversarial learning frameworks are used to learn the potential distribution of waypoints [28].

However, using nonlinear interpolation to refine the coarse trajectory introduces another issue: the interpolated waypoints of the trajectory may exceed the safe range, leading to potential collision risks.

Further iterations of numerical constrained optimization can be applied to the intermediate trajectory to enhance the quality of the trajectory [29]. A tracking planner can optimize parametric reference trajectories by utilizing lattice sampling to ensure the satisfaction of kinodynamic constraints [30]. Alternatively, a specific algorithm can be employed to generate multiple candidate paths, and subsequently, an objective function can be utilized to select the optimal trajectory. This method can also be considered as an indirect form of numerical optimization [31]. Another method in the same category involves the generation and evaluation of clothoid tentacles in the reference frame using various criteria [32]. These optimization methods either optimize rough trajectories within a large search space or evaluate a considerable number of candidate trajectories. However, challenges related to computational load and node accuracy remain. Iterative trajectory optimization methods can be employed to progressively handle complex constraints, gradually reducing the constraint conditions. Nevertheless, it is important to note that this iterative approach may be time-consuming [33].

To address the aforementioned issues, we propose a hierarchical scheme for trajectory planning of Connected and Autonomous Vehicles (CAVS) on structured roads. The proposed scheme consists of multiple layers that work hierarchically to refine the trajectory. In this scheme, the edge cloud module collects relevant information, such as vehicle states, road geometries, and traffic signs, either through road testing equipment or V2X communication. This information is then transmitted to the trajectory planning module of the ego vehicle. The trajectory planning module, based on the current traffic state, determines the ego-vehicle trajectory. It operates with three layers: the coarse trajectory layer, the smoothed trajectory layer, and the tunnel optimization layer. Each higher layer serves as a refinement of the lower layer, leading to a gradual reduction in the search space and computational load. Finally, the refined trajectories are sent to the

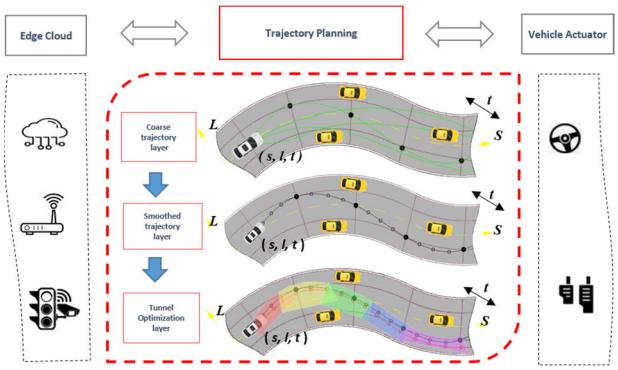

FIGURE 1. Hierarchical diagram of trajectory planning for ego vehicle (HD, zoom support).

vehicle actuator, enabling the execution of specific tasks such as steering, acceleration, and braking. A hierarchical diagram representing the trajectory planning process is depicted in Fig. 1.

In this paper, the authors initially focus on the coarse trajectory planning with random traffic settings, then construct safe regions of tunnel constraints for the smoothed trajectories, and finally conduct constraint optimization for the amended trajectories to obtain the feasible trajectories for the ego vehicle. Specifically, the contributions in this paper are as follows:

1) In the Frenet frame, the DP (Dynamic Programming) generation and interpolation techniques are used to process sparse coarse trajectories. This approach guarantees that the generated waypoints are physically feasible within the trajectory-smoothed layer.

2) Based on the smoothed trajectory, the spiral search method is utilized to seek the amended trajectory nearby the waypoints with latent collision risk, and the safe tunnels for the front and rear axles of the ego vehicle are dynamically constructed.

3) Building upon the amended trajectory and safe tunnels, the NLP (Nonlinear Programming) optimization is employed to further refine the trajectory of the ego vehicle. The optimization process takes into account various constraints and objectives to generate a trajectory that is both safe and smooth.

The remainder of this paper is organized as follows. Section II demonstrates coarse trajectory planning. Section III presents the problem of trajectory smoothing in line with vehicle kinematics. Section IV reveals the problem of constrained optimization of trajectories within the safety tunnels. Simulation results are shown in Section V, and Section VI concludes our efforts described throughout the paper.

II. COARSE TRAJECTORY PLANNING

To effectively model the traffic scene, a comprehensive understanding of the relevant information within the sampling horizon is imperative. This includes factors such as road geometry, road boundaries, lane lines, the status of cooperative vehicles, the intended movements of cooperative vehicles, the maximum steering angle of the ego vehicle, the maneuverability of the ego vehicle, traffic regulations, etc. Certain data can be seamlessly provided to the trajectory planner through V2X from the edge cloud. V2X communication is typically implemented using wireless communication technologies such as Dedicated Short-Range Communications (DSRC) or Cellular Vehicle-to-Everything (C-V2X) technology. These technologies allow vehicles to transmit and receive data, over short to medium ranges, enabling seamless communication between vehicles and their surroundings. V2X facilitates cooperative maneuvers between vehicles, such as merging, lane changing, and platooning, by sharing position, velocity, and trajectory data, leading to safer and smoother trajectory planning. Conversely, the remaining data can be acquired through the detection sensors of the ego vehicle.

To streamline the representation of the road grid with a continuous heading angle, we utilize the Frenet frame (s, l) to describe any given waypoint P. Here, the s coordinate axis aligns with the road reference line, while the l coordinate axis is perpendicular to it.

When traversing the optimal trajectory, it is essential to choose the shortest path within the given time interval and verify whether the current path intersects with a moving cooperative vehicle in the Frenet frame. Simultaneously, the absolute coordinates of point P will be utilized. Assuming that the projection of point P onto the road reference line corresponds to point p_1, the coordinates of p_1 in the Frenet frame are denoted as $(s, 0)$. Additionally, in the Cartesian frame, this point can be expressed as $(x(s), y(s))$, where $x(s)$ and $y(s)$ represent the horizontal and vertical coordinates in the Cartesian frame, respectively. The tangent direction angle of point p_1 at the road reference line is represented by Equation (1).

$$\theta = \arctan\left(\frac{y(s)}{x(s)}\right) \tag{1}$$

In the Cartesian frame, the point P can be represented as Equation (2) - (3).

$$x = x(s) + l\ cos(\theta + \frac{\pi}{2}) \tag{2}$$

$$y = y(s) + l\ sin(\theta + \frac{\pi}{2}) \tag{3}$$

Assuming that the ego vehicle undergoes K steps of node traversal in total and travels $N_s(i)$ nodes in each step in the s direction of the sampling horizon, where the distance between adjacent nodes in the s direction is Δs, and the travel time of each step is ΔT. The velocity of the vehicle in the s direction can be determined by Equation (4).

$$V_s(s) = \frac{N_s(i)\,\Delta s}{\Delta T},\ N_s(i) \in \{0, 1, 2, \ldots, p\},\ s \in \{1, 2, \ldots, K\} \tag{4}$$

The maximum length of each step, ΔL, can be got as Equation (5).

$$\Delta L = p\Delta s \tag{5}$$

In Equation (4), if $V_s(s)$ is equal to 0, it indicates that the ego vehicle is obstructed by a stationary surrounding vehicle, and the ego vehicle must either be in a parked state or unable to proceed in the s direction. This represents the lower limit of velocity in the s travel direction. On the other hand, if $V_s(s)$ is equal to $\frac{p\Delta s}{\Delta T}$, it signifies that the velocity of the ego vehicle has reached the upper limit in the s direction, suggesting a relatively smooth traffic situation ahead.

In the Frenet frame, the coordinates of the left boundary point L on the road surface can be represented as $(s, l_l(s))$, while the coordinates of the right boundary point R can be represented as $(s, l_r(s))$. Here, $l_l(s)$ and $l_r(s)$ are functions of s, respectively.

Taking into account the body width of the ego vehicle, the actual upper and lower limits of the road surface in l direction can be expressed as Equation (6) - (7).

$$l_{lc} = l_l(\mathrm{s}) - \frac{D_v + D_r}{2} \tag{6}$$

$$l_{rc} = l_r(\mathrm{s}) + \frac{D_v + D_r}{2} \tag{7}$$

where D_v and D_r are the width and its allowance for the ego vehicles, respectively.

Assuming that in the l direction, the ego vehicle travels $N_l(j)$ nodes in each step, the distance between the adjacent nodes in the l direction is Δl, and the travel time of each step is ΔT. Then the velocity of the vehicle in the l direction is expressed by Equation (8).

$$V_l(s) = \frac{N_l(j)\,\Delta l}{\Delta T}, N_l(i) \in \{0, 1, 2, \ldots, q\} \tag{8}$$

where q is the maximum number of nodes that the ego vehicle travels in the l direction.

The position at the initial instant of the ego vehicle, denoted as s_0 and l_0, is known for each sampling horizon. During each step of the trajectory, the ego vehicle traverses $N_s(i)$ nodes in the s direction and $N_l(j)$ nodes in the l direction. The final position of the $N_l(j)$ nodes in the l direction can be determined by Equation (9).

$$l_b = l_{rc} + (l_{lc} - l_{rc}) * \frac{N_l(j)}{q}, N_l(j) \in \{0, 1, 2, \ldots, q\} \tag{9}$$

It is crucial to evaluate the total cost value of every potential trajectory, which comprises the initial step to the final step.

The cost value is determined by the cost function, which should be designed based on the specific characteristics of driving, such as safety, traffic efficiency, energy consumption, ride comfort, and other relevant factors. The total cost function can be expressed by Equation (10).

$$Cost = \sum_{i=1}^{k} \left(\gamma(i) + w_s \frac{1}{V_s(i)} + w_l V_l(i) + w_\theta \theta(i) + w_m mean(l(i)) \right) \tag{10}$$

The first term in the accumulated cost function stands for the collision safety penalty, which is expressed by Equation (11).

$$\gamma(i) = \begin{cases} \infty & Collision\ occurrence \\ 0 & otherwise \end{cases} \tag{11}$$

The second term stands for the traffic efficiency penalty, which is weighted by w_s.

The third term stands for the lane-change penalty, which is weighted by w_l.

The fourth item stands for the lane-change penalty supplement, which can especially suppress the side slip of the ego vehicle, and w_θ weights this item.

The fifth item stands for the centerline penalty for deviating from the lane centerline, and w_m weights this item.

Thus, the optimal node trajectory for the ego vehicle can be obtained by the DP algorithm. To improve computational efficiency and avoid dimensionality curses, the grid of the road model should not be meshed densely for node traversal. The coarse trajectory can be presented by points p_i with Frenet coordinates (s_i, l_i, t_i), $i \in \{0, 1, 2, \ldots, k\}$ at equal time interval ΔT, where k stands for the maximum value of receding horizon.

To make the coarse trajectory easy to be followed by the ego vehicle, it is necessary to refine the coarse trajectory. It is a simple effort to interpolate n small-distance points p_{ij} ($j = 0, 1, 2, \ldots, n$) with equal intervals between two adjacent points p_i, thus the detailed trajectory can be obtained. The time interval Δt between adjacent small-distance points is fixed, so the ego vehicle travels at a constant velocity between the interpolated point series p_{ij}.

$$V_{sj} = (S_{i(j+1)} - S_{ij})/\Delta t \tag{12}$$

$$V_{lj} = (l_{i(j+1)} - l_{ij})/\Delta t \tag{13}$$

$$\Delta t = \Delta T/n \tag{14}$$

The distances between the adjacent large-distance points p_i obtained by the DP algorithm are often different, which can lead to sudden changes in the velocity of V_{sj} and V_{lj}, which are expressed by Equation (12) - (14). Thus, it will cause both a_{si} and a_{li} to be infinite at point p_i. Due to the inertia of the rotational and translational parts of the ego vehicle, it is impossible to physically follow such a coarse trajectory with sudden state change.

Fig. 2 shows the detailed trajectory obtained by equal interval interpolation of the coarse trajectory in the sampling horizon on the sock-shaped track. The detailed trajectory shown in Fig. 2 has several obvious inflections visible to our naked eye at the footprints marked by hollow rectangles, which are not feasible for the ego vehicle to follow. Further improvement is needed to ensure the trajectory velocity is continuous, and the trajectory state of the ego vehicle can be controllable.

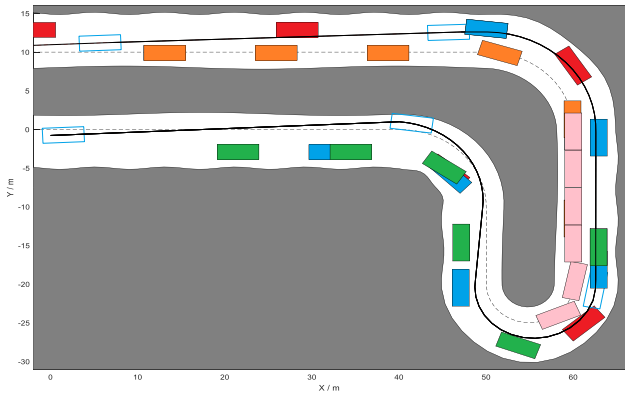

FIGURE 2. Footprint of the detailed trajectory with DP algorithm (Solid line: Detailed trajectory; Dashed line: Road reference line of left lane; Hollow rectangle: Ego vehicle; Solid rectangle: Cooperative vehicle) (SVG, Zoom Support).

III. SMOOTHING COARSE TRAJECTORY

In a traffic setting, it is not feasible to directly interpolate the coarse trajectory of the DP algorithm with equal intervals to obtain the vehicle waypoints. Although the detailed trajectory obtained in this way is continuous in position, the velocity is often discontinuous and unsmooth, which does not conform to the kinematics requirement of the vehicle entity.

To address the issue that detailed trajectories are infeasible for the ego vehicle to follow, we propose other interpolation methods to smooth the coarse trajectory such as quintic polynomial interpolation, and Bezier curve interpolation. These interpolation schemes can ensure the planned trajectory to be smooth, that is, the corresponding displacement and velocity can be continuous. We plan to improve the quality of the coarse trajectory from the perspectives of smoothness, collision safety, and efficiency.

A. QUINTIC POLYNOMIAL INTERPOLATION

High-order polynomials can easily fit many curves of different shapes passing through two fixed points, so that large-distance points p_i ($i = 0,1,2, \ldots, k$) can be connected by higher-order polynomial curves, and these points are expressed by the coordinates (s_i l_i) in the Frenet frame. When any two adjacent points p_i are fitted to a curve, the initial conditions such as position, velocity, and acceleration of both points can just determine the coefficients of a quintic polynomial. Therefore, the fitted curve is given by a quintic polynomial as Equation (15) - (16), where a_i is the coefficient of any polynomial term.

$$s(t) = a_5t^5 + a_4t^4 + a_3t^3 + a_2t^2 + a_1t + a_0 \quad (15)$$

$$l(t) = b_5t^5 + b_4t^4 + b_3t^3 + b_2t^2 + b_1t + b_0 \quad (16)$$

The position coordinates of the key points p_i in the Frenet frame have been given by the DP algorithm, and the corresponding velocity can be got in the following presentation. When the ego vehicle is overtaking or avoiding obstacle vehicles around on-road traffic flow, the decoupled trajectory $s(t)$, which is a monotonically increasing function with time t in the s component, can only move forward and cannot go backward. Nevertheless, the decoupled trajectory $l(t)$ will reciprocate across the reference line of the lane to avoid collision with the cooperative vehicles in the overtaking behavior. Thus, the displacement of the ego vehicle in the l component is no longer a monotonic function to time t.

For the initial curve segment p_0p_1 in the sampling horizon, the first and last velocities along the s-axis can be expressed by Equation (17) - (18), respectively.

$$V_{ss} = (s(1) - s(0))/\Delta T \quad (17)$$

$$V_{es} = (s(2) - s(0))/(\Delta T * 2) \quad (18)$$

While for the final curve segment $p_{k-1}p_k$, the corresponding velocities can be given by Equation (19) - (20), respectively.

$$V_{ss} = (s(k) - s(k-2))/(\Delta T * 2) \quad (19)$$

$$V_{es} = (s(k) - s(k-1))/\Delta T \quad (20)$$

For the middle curve segment p_ip_{i+1}, these are given by Equation (21) - (22), respectively.

$$V_{ss} = (s(i+1) - s(i-1))/(\Delta T * 2) \quad (21)$$

$$V_{es} = (s(i+2) - s(i))/(\Delta T * 2) \quad (22)$$

Since the ego vehicle can change lanes continuously during the overtaking maneuver, the coarse trajectory may reciprocate in the l component. There can exist many extreme points of the decoupled trajectory $l(t)$. It is not appropriate to take the average velocity of the adjacent nodes for the extreme point, which can cause the vehicle to rush out of the road boundary at the extreme point.

For the final curve segment $p_{k-1}p_k$ along the l- axis, the first and last velocities can be given by Equation (23) - (24), respectively.

$$V_{sl} = (l(k) - l(k-1))/\Delta T \quad (23)$$

$$V_{el} = V_{sl} \quad (24)$$

For the curve segment p_ip_{i+1} other than final one along the l-axis, its first and last velocities are given by Equation (25) - (26), respectively.

$$V_{sl} = (l(i+1) - l(i))/\Delta T \quad (25)$$

$$V_{el} = (l(i+2) - l(i+1))/\Delta T \quad (26)$$

If p_ip_{i+1} is not the first curve segment and $[l(i+1)-l(i)][l(i)-l(i-1)] < 0$, the velocity of the first node is updated by Equation (27).

$$V_{sl} = 0 \quad (27)$$

The acceleration values of the adjacent first point and last point of the curve segment p_ip_{i+1} in the Frenet frame can be expressed as Equation (28).

$$a_{ss} = a_{es} = C_s, a_{sl} = a_{el} = C_l \quad (28)$$

where C_s and C_l are all constants.

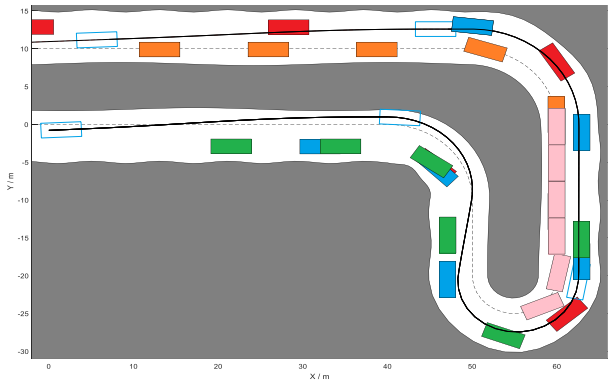

FIGURE 3. Smoothed trajectory with quintic polynomial interpolation (Solid line: Smoothed trajectory; Dashed line: Road reference line; Hollow rectangle: Ego vehicle; Solid rectangle: Cooperative vehicle) (SVG, Zoom Support).

Given the initial states of the first point and last point of the curve segment, the coefficients of the quintic polynomial curve can all be determined. By substituting the time t into the Equation (15) - (16), the trajectory coordinate (s, l, t) of the ego vehicle can be planned between the two key points traversed by the DP algorithm. In addition to the planned trajectory, the heading angle needs to be provided so that the ego vehicle can follow the trajectory smoothly. The heading angle $\theta_1(t)$ of the road reference line can be added to the trajectory coordinate, and the feasible trajectory coordinate $P(s, l, \theta_1, t)$ can be obtained to navigate the ego vehicle.

However, the tangent angle $\theta_2(t)$ of the vehicle trajectory with the quintic polynomial interpolation does not match the heading angle $\theta_1(t)$ in the Cartesian frame. To reduce the crab-walking effect and improve the trajectory applicability for the ego vehicle, Equation (29) can be used to compromise $\theta_1(t)$ and $\theta_2(t)$ to make the vehicle trajectory smoother.

$$\theta(t) = k_1\theta_1(t) + k_2\theta_2(t) \tag{29}$$

where k_1 and k_2 are the weight coefficients respectively.

Ultimately, we can get the trajectory coordinate $P(s, l, \theta, t)$ to navigate the ego vehicle. Fig. 3 shows the simulation process for the smoothed trajectories with the quintic polynomial interpolation. It can be seen that the trajectory becomes very smooth at the key points with hollow rectangle, especially for the 2nd or 3rd footprint.

B. BEZIER CURVE INTERPOLATION

The coarse trajectory coordinate (s, l, t) obtained by the DP algorithm can be decomposed into sub-coordinate (s, t). For $n+1$ corresponding control points $p_0(s_0, t_0)$, $p_1(s_1, t_1)$, …, $p_n(s_n, t_n)$, we connect these control points to form a control polyline. The polyline can be approximated to a smooth Bezier curve.

The waypoints of Bezier curve along the s-axis is determined by Equation (30), where C_n^i is a combination formula about n and i, and ξ is any bounded real variable.

$$s(i) = \sum_{i=0}^{n} C_n^i(1-\xi)^{n-i}\xi^i p_i,\ \xi\in[0,\ 1] \tag{30}$$

TABLE 1. Deviation rate of s value with Bezier curve interpolation.

Time t (s)	0	2	4	6	8	10
s Value of coarse trajectory (m)	0	40	73.33	100	140	180
s Value of Bezier curve (m)	0	37.48	71.35	104.81	140.82	180
Deviation rate (%)	0	-6.3%	-2.7%	4.8%	0.6%	0

TABLE 2. Deviation rate of l value with Bezier curve interpolation.

Time t (s)	0	2	4	6	8	10
l Value of coarse trajectory (m)	0.8	-0.26	1.18	2.60	-0.45	0.36
l Value of Bezier curve (m)	0.8	0.52	0.97	1.07	0.53	0.36
Deviation rate (%)	0	-300%	-17.8%	58.8%	217.8%	0

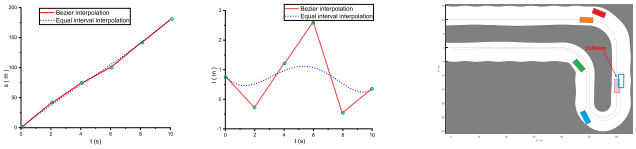

FIGURE 4. Interpolation of coarse trajectories with Bezier curve: (a) Bezier curve with (s, t) (Left); (b) Bezier curve with (l, t) (Middle); (c) Collision caused by large l deviation (Right). (SVG, Zoom Support).

Compared with a coarse trajectory case planned by the DP algorithm, the deviation rate of the s coordinate of the key point p_i for the Bezier curve interpolation is shown in Tab. 1.

Similarly, the l component of Bezier curve interpolation is determined by Equation (31), where C_n^i is a combination formula about n and i, and ξ is any bounded real variable.

$$l(i) = \sum_{i=0}^{n} C_n^i(1-\xi)^{n-i}\xi^i q_i,\ \xi\in[0,\ 1] \tag{31}$$

The deviation rate of the l coordinate of the key point p_i is shown in Tab. 2.

Fig. 4 shows the interpolation of coarse trajectories with Bezier curves in 2D coordinate systems, respectively.

Fig. 4a and Fig. 4b are consistent with Tab. 1 and Tab. 2 in the coordinate value of key points. It can be seen that the Bezier curves can only pass through the first and final key points and be tangent to the head and tail of the polyline segment formed by all key points. For strictly monotonic s-polyline, the Bezier curve can follow the polyline segment better, and the fitting errors are relatively small. While for non-monotonic l-polyline, the deviation rate of Bezier curve fitting trajectory at the key points are relatively large. As deviating from the key points of the coarse trajectory, it can collide with the cooperative vehicles. Fig. 4c shows that the ego vehicle follows the smoothed trajectory interpolated by Bezier curve, and collides with the red cooperative vehicle for large l deviation from key points.

From the perspective of deviation rate, the trajectories generated by equal interval interpolation and quintic polynomial interpolation strictly pass through the key points, and there exists no deviation from key points. However, Bezier curve interpolation generates significant deviation near the key points, shown in Fig. 4a and 4b.

Analysis from collision safety, the trajectories generated by equal interval interpolation and quintic polynomial interpolation do not collide with the surrounding vehicles in a dynamic environment, as shown in Fig. 2 and Fig. 3. While the trajectory of Bezier curve interpolation collides in a dynamic environment, as shown in Fig. 4c.

TABLE 3. Execution time comparison with different Schemes.

CASE NO.	1	2	3	4	5	6
Scheme A (ms)	1.033	1.028	1.021	1.016	1.011	1.023
Scheme B (ms)	8.437	8.536	8.410	8.363	8.368	8.284
Scheme C (ms)	7.348	7.370	7.261	7.254	7.208	7.189

Note: Scheme A - equal interval interpolation; Scheme B - quintic polynomial interpolation; Scheme C- Bezier curve interpolation.

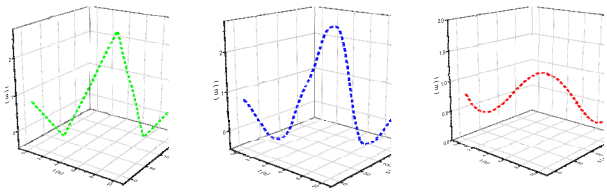

FIGURE 5. 3D Trajectory comparison of different interpolation schemes: a) Equal interval interpolation (Left); b) Quintic polynomial interpolation (Middle); c)Bezier curve interpolation (Right). (HD, Zoom Support).

We compare the execution time of Matlab code of equal interval interpolation, quintic polynomial interpolation, and Bezier curve interpolation for random 6 times respectively, as shown in Tab. 3.

The trajectory planning of the ego vehicle on the road traffic scenarios needs to ensure real-time performance of computation. It can be seen from Tab. 3 that the program execution time of the equal interval interpolation, Scheme A, is shortest of all schemes, and the time of the other schemes is basically longer than an order of time magnitude. The computation time of quintic polynomial interpolation to complete the task is slightly longer than that of Bezier interpolation. The program is performed in MATLAB 2020a and run on a computer with a CPU of Intel i9-12900KF and a memory of 32.0 GB. When using high-performance hardware and a programming language with strong real-time, the computing speed will be greatly improved.

Fig. 5 compares the 3D spatiotemporal trajectory curves of three interpolation schemes for coarse trajectories. The trajectory inflection of equal interval interpolation shown in Fig. 5a are obvious. Although the trajectory of Bezier curve interpolation shown in Fig. 5c is relatively smooth, the shape distortion is more prominent. The trajectory of quintic polynomial interpolation shown in Fig. 5b can well balance smoothness and shape fidelity.

From the above discussion, we weighed the deviation rate of key points, collision safety, planning efficiency, and smoothness, and therefore chose the trajectory of quintic polynomial interpolation as the smoothed trajectory for ego vehicle, and further discuss will be conducted thereinafter.

IV. NUMERICAL OPTIMIZATION OF SMOOTHED TRAJECTORY

According to our preliminary analysis, the smoothed trajectory created using quintic polynomial interpolation demonstrates favorable collision safety. However, it is important to note that apart from the key points of the coarse trajectory, the interpolation points of other waypoints may still pose a risk of colliding with the road boundary or cooperative vehicles, especially in extreme settings. To ensure the safety of all waypoints along the smoothed trajectory, further optimization is necessary.

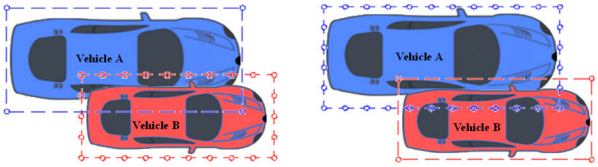

FIGURE 6. Outline overlap principle to check collision of vehicles: a) Vehicle B enters into the outline of vehicle A (Left); b) Vehicle A enters into the outline of vehicle B (Right). (HD, Zoom Support).

A. BUILDING COLLISION-FREE TUNNEL

Both the ego vehicle and the cooperative vehicles can be simplified as rectangles for collision detection, and the coordinate positions of these rectangles can be reliably obtained through onboard sensors or edge clouds. A collision is considered to occur when the rectangle of the ego vehicle overlaps that of cooperative vehicles.

Fig. 6 depicts a schematic diagram of checking a collision between two vehicles. If any discrete point of one vehicle enters the bounding contour of another one, a collision is sure to occur. Fig. 6a shows the collision judgment principle of vehicle B entering the outline of vehicle A. To prevent negligence caused by the complete inclusion, it is necessary to simultaneously judge whether any discrete point of vehicle A enters the boundary outline of vehicle B. Fig. 6b shows the judgment schematic of vehicle A entering the boundary outline of vehicle B.

When driving on curvy roads, the trajectory of the front and rear axle centers does not coincide. To prevent collision accidents, it is necessary to determine safe driving zones for the front and rear axles separately.

To find a safety zone of point P for the center of the rear axle of the ego vehicle in Fig. 7a, the vehicle contour can be expanded around the ego vehicle. The maximum extension vector is defined by the vector d_{lim} as in Equation (32) - (34).

$$d_{lim} = [d_w, d_L, d_w, d_L]^T \tag{32}$$

$$d_w = w_L - w_B \tag{33}$$

$$d_L = 2 * (w_L - w_B) \tag{34}$$

where the vector d_{lim} is composed of the maximum extension distance from the left edge, rear one, right one, and front one of the ego vehicle, respectively. w_L stands for the lane width, and w_B stands for the vehicle width.

The vehicle contour is expanded in four different directions within a constant step. It is necessary to judge whether the expanded vehicle contour collides with the cooperative vehicles or the road boundaries for each expansion. If there is a collision in a certain direction or the expansion distance

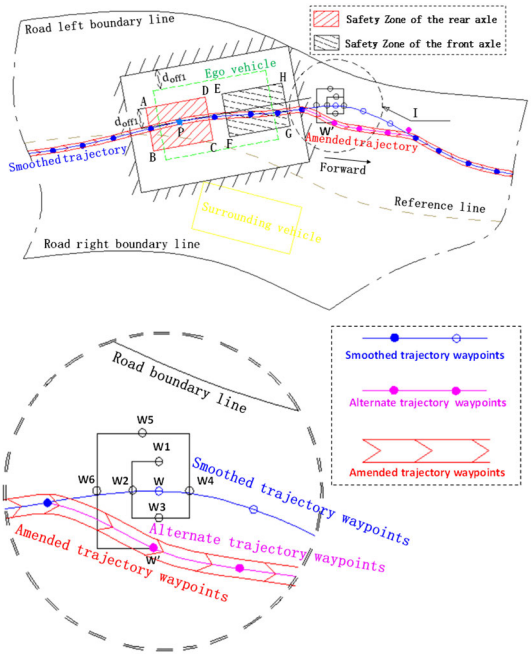

FIGURE 7. Schematic of the safety zone for the front and rear axles: a) Overall view (Top); b) Partially enlarged view (Bottom). (HD, Zoom Support).

exceeds the component of d_{lim}, the expansion along this direction is terminated, while the expansion in other directions continues. The extension vector d_{off} can be determined outside the four different directions in Equation (35).

$$d_{off} = [d_{off1}, d_{off2}, d_{off3}, d_{off4}]^T \tag{35}$$

where the vector d_{off} is composed of the actual extension distance from the left edge, the rear one, the right one, and the front one of the ego vehicle, respectively.

With the offsets of d_{off1},d_{off2}, d_{off3} and d_{off4} respectively around the point P, the rectangle ABCD can be obtained in Fig. 7a. The rectangular ABCD is the safety zone of the center of the rear axle of the ego vehicle. Similarly, the rectangular EFGH is the safety area of the front axle in Fig. 7a.

If any component of vector d_{off} is equal to 0, it means that the ego vehicle collides with the cooperative vehicles or the road boundaries at the current waypoint. To avoid the over-complicated graph near point P in Fig. 7a, we examine another waypoint W (x, y) in the smoothed trajectory. Suppose that there exists a case where a certain component of the vector d_{off} of waypoint W is equal to 0. It is necessary to find a new collision-free waypoint around the current waypoint W (x, y) with a certain step size. The method is to traverse a new point in the vicinity of the point W along the spiral point series $W_1, W_2, W_3, W_4, W_5, \ldots, W_i, \ldots$until there exist four non-zero margins in all the directions of some point W_i. The spiral point series is gradually away from the point W. When the Point W_i is found, it is the alternate trajectory waypoint W' for the original smoothed trajectory waypoint W in Fig.7b.

The waypoints of the alternate trajectory (Pink solid dot) and the waypoints of the smoothed trajectory without collision risk (Blue solid dot) together constitute the amended trajectory waypoints. Fig. 7b is a partially enlarged view of Fig. 7a, and Fig. 7b details the process of searching for an alternate trajectory waypoint with a spiral line.

$$\left.\begin{aligned} x_{rt1} &= x - d_{off2}cos\theta - d_{off1}sin\theta \\ y_{rt1} &= y + d_{off1}cos\theta - d_{off2}sin\theta \\ x_{rt2} &= x + d_{off4}cos\theta - d_{off1}sin\theta \\ y_{rt2} &= y + d_{off1}cos\theta + d_{off4}sin\theta \\ x_{rt3} &= x + d_{off4}cos\theta + d_{off3}sin\theta \\ y_{rt3} &= y - d_{off3}cos\theta + d_{off4}sin\theta \\ x_{rt4} &= x - d_{off2}cos\theta + d_{off3}sin\theta \\ y_{rt4} &= y - d_{off3}cos\theta - d_{off2}sin\theta \end{aligned}\right\} \tag{36}$$

The coordinates, $A(x_{rt1}, y_{rt1})$, $B(x_{rt2}, y_{rt2})$, $C(x_{rt3}, y_{rt3})$ and $D(x_{rt4}, y_{rt4})$, of the four vertices of the rectangle ABCD, which marks the safety rectangular zone around the center of the rear axle of the ego vehicle, can be expressed as Equation (36), where the coordinate (x, y) is the center of the rear axle of the ego vehicle, the vector d_{off} is defined by Equation (35) and θ is the heading angle of ego vehicle of smoothed trajectory.

The coordinates, $E(x_{ft1}, y_{ft1})$, $F(x_{ft2}, y_{ft2})$, $G(x_{ft3}, y_{ft3})$, and $H(x_{ft4}, y_{ft4})$, of the four vertices of the rectangle EFGH, which marks the safety rectangular zone around the center of the front axle of the ego vehicle, can be expressed as Equation (37).

$$\left.\begin{aligned} x_{ft1} &= x_{rt1} + L_{rf}\cos\theta,\ y_{ft1} = y_{rt1} + L_{rf}\sin\theta \\ x_{ft2} &= x_{rt2} + L_{rf}\cos\theta,\ y_{ft2} = y_{rt2} + L_{rf}\sin\theta \\ x_{ft3} &= x_{rt3} + L_{rf}\cos\theta,\ y_{ft3} = y_{rt3} + L_{rf}\sin\theta \\ x_{ft4} &= x_{rt4} + L_{rf}\cos\theta,\ y_{ft4} = y_{rt4} + L_{rf}\sin\theta \end{aligned}\right\} \tag{37}$$

where L_{rf} is the wheelbase of the ego vehicle and θ is the heading angle of the ego vehicle of smoothed trajectory.

Thus, the rear safety zone of the center of the rear axle, illustrated by the red rectangle ABCD in Fig. 7a, and the front safety zone of the front axle, illustrated by the black rectangle EFGH in Fig. 7a, are both obtained by Equation (36) and (37). When the waypoint corresponding to the center of the rear axle update forward along the amended trajectory, the rear safety zone and the front safety zone all update. The envelope of numerous rectangles of the rear safety zone are superimposed to form the collision-free tunnel of the rear axle center with the red hollow rectangles in Fig. 8. Likewise, that of the front safety zone are superimposed together to form the collision-free tunnel of the front axle center with the black hollow rectangles in the same figure. Fig. 8 also shows the footprints of key points of the coarse trajectory for five cooperative vehicles.

B. TRAJECTORY CONSTRAINT OPTIMIZATION

We have built two unobstructed safety tunnels for the ego vehicle, constraining the center of the rear axle and that of the front axle respectively, along the amended trajectory. These collision-free tunnels can serve as position constraints

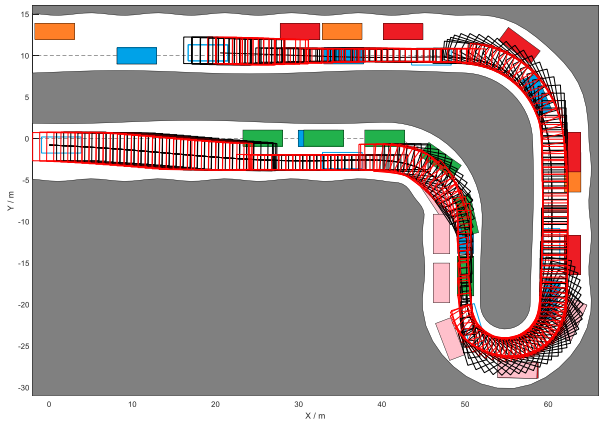

FIGURE 8. Collision-free tunnels of the center of the rear and the front axle (Red Rectangle: Tunnel of the rear axle; Black Rectangle: Tunnel of the front axle) (SVG, Zoom Support).

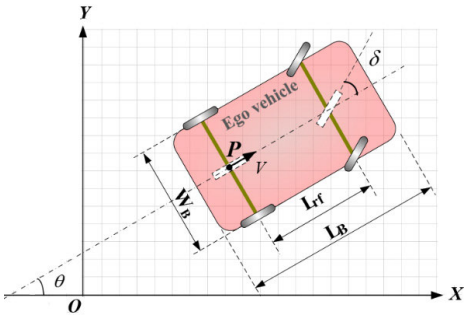

FIGURE 9. Vehicle kinematics model of the ego vehicle.

of the ego vehicle for subsequent numerical optimization of trajectory. The replanned trajectories are always safe from the cooperative vehicle and road boundary as long as the axle centers of the ego vehicle are within the corresponding safety tunnels. Since there are infinitely feasible trajectories for the ego vehicle to travel, we need to add other constraints and construct an effective cost function to ultimately find the optimal trajectory for the ego vehicle.

In addition to the tunnel safety constraint, the motion of the ego vehicle must also meet the constraint of the vehicle kinematics model, so that the vehicle entity can run smoothly. We build a 2-DOF vehicle kinematics model of the ego vehicle, as shown in Fig. 9.

The state vector can be defined as Equation (38)

$$\xi(t) = \left[x, y, v, \theta, \delta, x_f, y_f\right]^T \tag{38}$$

where (x, y) and (x_f, y_f) respectively stand for the positions of the rear and front axle center of the ego vehicle, v stands for the velocity of the rear axle of the ego vehicle, θ stands for the heading angle of the ego vehicle, and δ stands for the steering angle of the front wheel of the ego vehicle.

Assuming that the axle loads of the front axle and the rear one of the ego vehicle are equal, the velocity difference between the center of the front axle and the rear axle is negligible, and the slip angle of the vehicle is small. The differential equation (39) - (43) of vehicle motion can be obtained from the vehicle kinematics model, and each state of the ego vehicle must satisfy the constraint of these equations [34].

$$\dot{x} = v\cos\left(\theta + \frac{\delta}{2}\right) \tag{39}$$

$$\dot{y} = v\sin\left(\theta + \frac{\delta}{2}\right) \tag{40}$$

$$\dot{\theta} = \frac{v\cos\left(\frac{\delta}{2}\right)\tan\delta}{L_{rf}} \tag{41}$$

$$x_f = x + L_{rf}cos\theta \tag{42}$$

$$y_f = y + L_{rf}sin\theta \tag{43}$$

The control vector can be defined as Equation (44)

$$u(t) = [v, \delta]^T \tag{44}$$

The trajectory optimization presented subsequently is a constrained nonlinear program (NLP) which consists of a cost function and several constraint conditions. We can model the constrained problem as Equation (45) and use the interior point method to obtain a feasible trajectory for the ego vehicle.

$$\begin{aligned} &\textbf{\textit{Minimize}}\ f\left(\xi\left(t\right), u\left(t\right)\right), t \in [0, t_f] \\ &s.t.\ c_i\left(\xi\left(t\right), u\left(t\right)\right) \le 0, i = 1, 2, \ldots, k \\ &h_i\left(\xi\left(t\right), u\left(t\right)\right) = 0, j = 1, 2, \ldots, l \end{aligned} \tag{45}$$

where $f\left(\xi\left(t\right), u\left(t\right)\right)$ and $c_i\left(\xi\left(t\right), u\left(t\right)\right)$ are both convex and second-order differentiable, and t_f represents the maximum time of the receding horizon.

The cost function f(ξ(t), u(t)) is defined as Equation (46)

$$\begin{aligned} f(\xi(t), u(t)) = &\int_0^{t_f} w_\xi \left\|\xi - \xi_{ref}\right\|^2 dt + \int_0^{t_f} w_u \|\dot{u}\|^2 dt \\ &+ \int_0^{t_f} w_\theta \left\|\theta - \theta_{\text{traj}}\right\|^2 dt \end{aligned} \tag{46}$$

where w_ξ, w_u, and w_θ are all weighting vectors for each error term, and $\xi_{ref} = [x_{ref}, y_{ref}, \theta_{ref}, x_{fref}, y_{fref}]^T$ is the amended trajectory state of the vehicle kinematics model, and $\theta_{traj}\left(t\right) = [\theta_{amen}, \theta_{fren}]^T$ is the nominal heading angle. θ_{amen} and θ_{fren} respectively represent the heading angle of the amended trajectory of the ego vehicle and that of the road reference line in the Frenet frame.

The first term of the polynomial in Equation (46) represents the weighted cumulative error between the current vehicle state and the amended trajectory state of kinematics model. The second term represents the fluctuation error of the current control vector defined by Equation (44). And the third term represents the cumulative error of the current heading vector deviating from the nominal vector. The error component of $\|\theta - \theta_{amen}\|$ guides the ego vehicle to drive along the direction of the amended trajectory, and the component of $\left\|\theta - \theta_{fren}\right\|$ guides the ego vehicle to drive along the s-axis direction.

TABLE 4. Simulation scenarios with two cases.

PARAM	S_0(m)		L_0(m)		S_e(m)		v (m/s)	
Case	A	B	A	B	A	B	A	B
Veh1 (RD)	46.9	97.8	2.87	2.86	197.6	225.9	15.1	12.8
Veh2 (BU)	30.6	90.0	2.93	2.85	115.9	152.9	8.5	6.3
Veh3 (GN)	20.1	66.1	2.90	-0.05	84.8	125.3	6.5	5.9
Veh4 (OG)	88.4	74.7	-0.10	2.90	152.6	142.4	6.4	6.8
Veh5 (PK)	75.5	23.4	-0.02	2.92	99.7	28.1	2.4	0.5

Note: The initial position of the ego vehicle is (0, 0.8), and the receding horizon is a total of 200 sampling horizons.

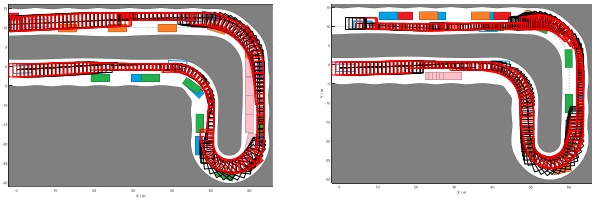

FIGURE 10. Safety tunnels of the trajectories of the ego vehicle: a) Case A (Left); b) Case B (Right). (SVG, Zoom Support).

The first constraint inequality with Equation (45) is the position constraint of the center of the rear axle and that of the front axle within the corresponding tunnel safety zones, respectively. The safety zones enclosed by the rectangle of the front and rear axles of the ego vehicle are both convex ones, which satisfies the condition of the optimal solution.

The second constraint equality with Equation (45) stands for the differential equations of vehicle kinematics with (39) - (43) and the initial settings of the state profiles.

Based on the sampling time of the road grid, the cost function represented by Equation (46) and the corresponding constraint conditions can all be discretized, and then the NLP problem can be solved numerically.

V. TRAJECTORY SIMULATION AND ANALYSIS

Trajectory planning with dynamic traffic flow is simulated in a sock-like track, which includes different segments, such as straight line, right-turn, U-turn, and left-turn. The initial position, terminal position, velocity, and other parameters of five cooperative vehicles can be randomly generated in the simulation scenarios.

To compare the trajectory planning result, Tab. 4 gives the initial position and velocity of the ego vehicle with Case A and Case B. For the parameters in Tab. 4, S_0 stands for the initial position along the s-axis, L_0 stands for that along the l-axis, S_e stands for the final position along the s-axis, and v stands for the velocity of the cooperative vehicle in the receding horizon. Since the ego vehicle has many maneuvers, such as overtaking, ACC, or lane-changing, its velocity changes all the time in the receding horizon.

Fig. 10 shows the safety tunnels of amended trajectories for both Case A and Case B with the scenarios profiled by Tab. 4. The subsequent NLP optimization of the trajectories of the ego vehicle would be constrained within the tunnels.

Fig. 11 shows the footprints of all vehicles in the respective simulation scenarios of the two cases corresponding to Tab. 4,

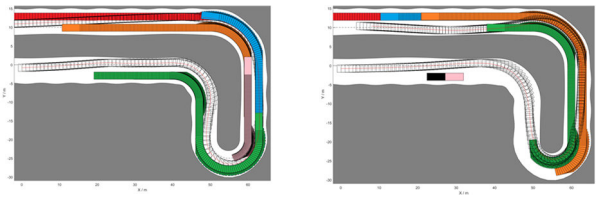

FIGURE 11. Footprints of all vehicles during the receding horizon: a) Case A (Left); b) Case B (Right). (HD, Zoom Support).

where the hollow rectangles represent the ego vehicle and the solid ones represent the cooperative vehicles. It can be seen that the ego vehicle constantly changes lanes to avoid collision and overtakes the cooperative vehicles, and indeed the NLP optimization algorithm can plan feasible trajectories in both cases.

In both cases depicted in the receding horizon of Fig. 11, the long footprint of any vehicle indicates that the vehicle's velocity is high, while the short one indicates that the velocity is low. In Fig. 11a, the red vehicle travels the longest distance, corresponding to the track position from 46.9m to 197.6m along the s-axis in Tab. 4, and the average velocity is 15.1m/s. In the same case, the pink vehicle travels the shortest distance, which corresponds to the track position from 75.5m to 99.7m along the s-axis in Tab. 4, and the average velocity is 2.4m/s. In Fig. 11b, the red vehicle travels the longest distance, corresponding to the track position from 97.8m to 225.9m along the s-axis in Tab. 4, and the average velocity is 12.8m/s. In the same case, the pink vehicle travels the shortest distance, which corresponds to the track position from 23.4m to 28.1m along the s-axis in Tab. 4, and the average velocity is 0.5m/s. In the two cases shown in Tab. 4, it happens that the red cooperative vehicles are both the fastest and the pink ones are both the slowest.

For Case A with Tab. 4, the velocities of the five cooperative vehicles are high on the whole, and the initial s-coordinate values of green and orange vehicles with relatively close velocities are quite different, thus the traffic flow formed by the cooperative vehicles is relatively smooth. It can be seen from Fig. 11a that the trajectory of the ego vehicle planned by the NLP optimization algorithm is relatively gentle, and the trajectory curvature of the lane-changing is relatively small as the ego vehicle travels along the track. While for Case B, the velocities of cooperative vehicles are generally lower than those in Case A. In particular, the mean velocity of the pink vehicles in Case B is low to 0.5m/s, which is almost equivalent to the stationary obstacle vehicle on the road. And the initial s-coordinates of the blue, green and orange vehicles are relatively close at 90.0m, 66.1m and 74.7m, respectively. The velocities of these vehicles are also close, and they occupy the left and right lanes from the initial waypoints of the receding horizon. These data indicate that the traffic flow composed of cooperative vehicles in Case B is relatively congested. It can be seen from Fig. 11b that an emergency lane-change with the trajectory of the ego vehicle,

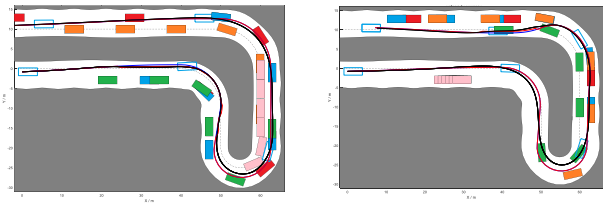

FIGURE 12. Trajectories and partial footprints of all vehicles: a) Case A (Left); b) Case B (Right). (Dashed line: Road reference line; Blue line: Smoothed trajectory Red line: Amended trajectory; Black line: Optimized trajectory) (SVG, Zoom Support).

and the curvature of the lane-change trajectory is relatively large when the ego vehicle travels along the track to the left lane. Compared with case A, the overall velocity of the ego vehicle in case B is lower because the traffic scene of case B is more congested. The result is that the footprints of the ego vehicle in Fig. 11b show that the vehicle does not travel the full track in the receding horizon.

Key footprints of all vehicles along the track are given for both cases in Fig. 12. The dashed black line between the road boundary lines in the figure represents the left road reference line, the solid blue one represents the smoothed trajectory of the rear axle center of the ego vehicle, the solid red one represents the amended trajectory of that, and the solid black one represents the NLP optimization trajectory.

Fig. 12 presents clear comparisons among the smoothed trajectory, the amended trajectory, and the optimized trajectory in both cases. The majority of segments in the smoothed trajectory and the amended trajectory coincide, with only a few segments exhibiting differences. However, there are significant differences between the smoothed trajectory and the optimized trajectory. During the preparation stage of the first right turn in case A, the amended trajectory deviates from the smoothed trajectory due to the quintic polynomial curve interpolation, causing the left side of the ego vehicle to collide with the road boundary. To obtain the amended trajectory, the trajectory of the center of the rear axle was replanned by identifying free-collision waypoints along a spiral line around the smoothed trajectory. Although the curve of the amended trajectory is more rugged, the corresponding ego vehicle successfully enters the safety tunnel. Similarly, in case B, the smoothed trajectory of the ego vehicle collides with the left boundary at the initial segment of the top track. However, the amended trajectory avoids the left boundary with a bulge. At the end of the right turn for case A, the amended trajectory exhibits two discontinuous bulging segments in the left direction of the smoothed trajectory. These two bulging segments are a result of the ego vehicle colliding with Veh 2 during the overtaking maneuver. Fortunately, the optimized trajectory can completely smooth out the bulging segments mentioned above in the amended trajectory.

Fig. 13 depicts the velocities and steering angles of the front wheel trajectories for three types of trajectories in case A and case B. These trajectories include the smoothed trajectory, the amended trajectory, and the optimized trajectory,

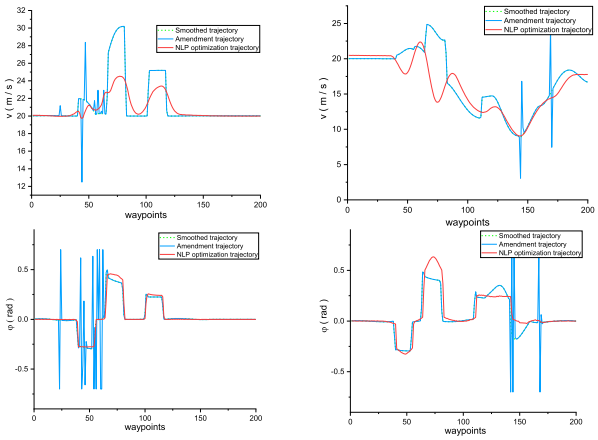

FIGURE 13. Comparison of velocities and steering angles of planned trajectories for both cases: a) Velocities (Case A (Top left) and Case B (Top right)); b) Steering angles of front wheel (Case A (Bottom left) and Case B (Bottom right)). (SVG, Zoom Support).

which are the output results of the lower, middle, and upper planners in the hierarchical trajectory planner, respectively.

In Case A of Fig. 13, besides the curve difference between waypoints No. 25-65, the trajectory curves of velocity and steering angle closely align with the lower and middle planner. The disparity in the trajectory curves between velocity and steering wheel angle primarily stems from the waypoint correction performed using the spiral search method. Following the correction, there are certain segments in the trajectory that are not smooth, leading to abrupt fluctuations in the curves of velocity and steering wheel angle.

In Case B of Fig. 13, except for the curve difference between waypoints No. 140-180, the trajectory curves of velocity and steering angle align perfectly with the lower and middle planner.

On the whole, Fig. 13 manifests that the trajectory planned by the middle planner only considers the safety of the trajectory, and there exist oscillations of velocity and steering angle, while the trajectory planned by the upper planner not only considers the safety of the trajectory, but also considers the vehicle kinematics characteristics. The upper trajectory obtained by numerical solution can not only avoid collision, but also be smooth and feasible for the on-board controller of the ego vehicle.

VI. CONCLUSION

This paper proposes a hierarchical framework of trajectory planning for connected and automated vehicles in complex traffic scenarios. The proposed framework consists of three layers: the coarse trajectory layer, the smoothed trajectory layer, and the tunnel optimization layer. Each higher level in the hierarchy enhances performance and refines the trajectory generated by the lower level. All trajectory algorithms proposed in this study have been thoroughly validated through simulation experiments, which can be accessed on the attached website.

The planner proposed in this study demonstrates the following characteristics:

1)The trajectory generation using the coarse trajectory planning method, which combines the DP algorithm and quintic polynomial interpolation, demonstrates high efficiency, minimal errors in key waypoints, and maintains a continuous state.

2)The spiral search method is employed to identify safe waypoints for those with collision risk, and the amended trajectory is formed by combining the new waypoints with the original ones that do not pose a collision risk. The amended trajectory forms safety tunnels that can be utilized as zones for constrained optimization.

3)The nonlinear optimized trajectory not only guarantees collision safety, but also ensures smooth changes in velocity and the corresponding front wheel angle of the planned trajectory with minimal jitter. This facilitates trajectory tracking by the physical actuator.

Future research will focus on integrating trajectory planning with traffic perception to develop safer trajectories. This integration will involve predicting the lateral driving intentions of cooperative vehicles, which will enable the planner to generate trajectories that prioritize safety.

ACKNOWLEDGMENT

The simulation experiments of animation can be visited at https://www.bilibili.com/video/BV1FX4y1W7vX/?vd_source=9db0a80ea67589a23aa8661a4da5933f.

REFERENCES

[1] B. Paden, M. Cáp, S. Z. Yong, D. Yershov, and E. Frazzoli, "A survey of motion planning and control techniques for self-driving urban vehicles," *IEEE Trans. Intell. Vehicles*, vol. 1, no. 1, pp. 33–55, Mar. 2016.

[2] Y. Ma, Z. Li, and M. A. Sotelo, "Testing and evaluating driverless vehicles' intelligence: The Tsinghua lion case study," *IEEE Intell. Transp. Syst. Mag.*, vol. 12, no. 4, pp. 10–22, Winter 2020.

[3] J. Gao, C. Sun, H. Zhao, Y. Shen, D. Anguelov, C. Li, and C. Schmid, "VectorNet: Encoding HD maps and agent dynamics from vectorized representation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 11522–11530.

[4] J. Gregoire and E. Frazzoli, "Hybrid centralized-distributed autonomous intersection control: Using a job scheduler as a planner and inheriting its efficiency guarantees," in *Proc. IEEE CDC*, Dec. 2016, pp. 2549–2554.

[5] M. Wang, Z. Wang, J. Talbot, J. C. Gerdes, and M. Schwager, "Game-theoretic planning for self-driving cars in multivehicle competitive scenarios," *IEEE Trans. Robot.*, vol. 37, no. 4, pp. 1313–1325, Aug. 2021.

[6] A. Zyner, S. Worrall, and E. M. Nebot, "ACFR five roundabouts dataset: Naturalistic driving at unsignalized intersections," *IEEE Intell. Transp. Syst. Mag.*, vol. 11, no. 4, pp. 8–18, Winter 2019.

[7] M. Cai, Q. Xu, C. Chen, J. Wang, K. Li, J. Wang, and X. Wu, "Formation control with lane preference for connected and automated vehicles in multi-lane scenarios," *Transp. Res. C, Emerg. Technol.*, vol. 136, Mar. 2022, Art. no. 103513.

[8] F. Zhao, W. Wu, Y. Wu, Q. Chen, Y. Sun, and J. Gong, "Model predictive control of soft constraints for autonomous vehicle major lane-changing behavior with time variable model," *IEEE Access*, vol. 9, pp. 89514–89525, 2021.

[9] D. González, J. Pérez, V. Milanés, and F. Nashashibi, "A review of motion planning techniques for automated vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 4, pp. 1135–1145, Apr. 2016.

[10] W. Lim, S. Lee, M. Sunwoo, and K. Jo, "Hierarchical trajectory planning of an autonomous car based on the integration of a sampling and an optimization method," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 2, pp. 613–626, Feb. 2018.

[11] L. Ma, J. Xue, K. Kawabata, J. Zhu, C. Ma, and N. Zheng, "A fast RRT algorithm for motion planning of autonomous road vehicles," in *Proc. 17th IEEE Int. Conf. Intell. Transp. Syst.*, Oct. 2014, pp. 1033–1038.

[12] S. Feraco, S. Luciani, A. Bonfitto, N. Amati, and A. Tonoli, "A local trajectory planning and control method for autonomous vehicles based on the RRT algorithm," in *Proc. AEIT Int. Conf. Electr. Electron. Technol. Automot. (AEIT AUTOMOTIVE)*, Nov. 2020, pp. 1–6.

[13] Y. Li, W. Wei, Y. Gao, D. Wang, and Z. Fan, "PQ-RRT*: An improved path planning algorithm for mobile robots," *Exp. Syst. App.*, vol. 152, no. 15, pp. 1–11, 2020.

[14] W. Sheng, B. Li, and X. Zhong, "Autonomous parking trajectory planning with tiny passages: A combination of multistage hybrid A-star algorithm and numerical optimal control," *IEEE Access*, vol. 9, pp. 102801–102810, 2021.

[15] X. Xiong, H. Min, Y. Yu, and P. Wang, "Application improvement of A* algorithm in intelligent vehicle trajectory planning," *Math. Biosciences Eng.*, vol. 18, no. 1, pp. 1–21, 2021.

[16] Y. Singh, S. Sharma, R. Sutton, and D. Hatton, "Towards use of Dijkstra algorithm for optimal navigation of an unmanned surface vehicle in a real-time marine environment with results from artificial potential field," *TransNav, Int. J. Mar. Navigat. Saf. Sea Transp.*, vol. 12, no. 1, pp. 125–131, 2018.

[17] F. Zhang, R. Xia, and X. Chen, "An optimal trajectory planning algorithm for autonomous trucks: Architecture, algorithm, and experiment," *Int. J. Adv. Robotic Syst.*, vol. 17, no. 2, pp. 1–11, Mar. 2020.

[18] Z. Dai, X. C. Liu, X. Chen, and X. Ma, "Joint optimization of scheduling and capacity for mixed traffic with autonomous and human-driven buses: A dynamic programming approach," *Transp. Res. C, Emerg. Technol.*, vol. 114, pp. 598–619, May 2020.

[19] W. B. Powell, *Approximate Dynamic Programming: Solving the Curses of Dimensionality*, 2nd ed. Hoboken, NJ, USA: Wiley, Sep. 2011, pp. 111–121.

[20] Z. Zhu, S. Gupta, N. Pivaro, S. R. Deshpande, and M. Canova, "A GPU implementation of a look-ahead optimal controller for eco-driving based on dynamic programming," in *Proc. ECC*, Jun. 2021, pp. 899–904.

[21] Y. Bian, S. E. Li, B. Xu, X. Qin, S. Li, Q. Xu, J. Wang, and K. Li, "Behavioral harmonization of a cyclic vehicular platoon in a closed road network," *IEEE Trans. Intell. Vehicles*, vol. 6, no. 3, pp. 559–570, Sep. 2021.

[22] H. Xu, W. Xiao, C. G. Cassandras, Y. Zhang, and L. Li, "A general framework for decentralized safe optimal control of connected and automated vehicles in multi-lane signal-free intersections," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 10, pp. 17382–17396, Oct. 2022.

[23] I. A. Ntousakis, I. K. Nikolos, and M. Papageorgiou, "Optimal vehicle trajectory planning in the context of cooperative merging on highways," *Transp. Res. C, Emerg. Technol.*, vol. 71, pp. 464–488, Oct. 2016.

[24] X. Hu and J. Sun, "Trajectory optimization of connected and autonomous vehicles at a multilane freeway merging area," *Transp. Res. C, Emerg. Technol.*, vol. 101, pp. 111–125, Apr. 2019.

[25] J. Xiao, Z. Xiao, D. Wang, V. Havyarimana, C. Liu, C. Zou, and D. Wu, "Vehicle trajectory interpolation based on ensemble transfer regression," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 7680–7691, Jul. 2022.

[26] R. van Hoek, J. Ploeg, and H. Nijmeijer, "Cooperative driving of automated vehicles using B-splines for trajectory planning," *IEEE Trans. Intell. Vehicles*, vol. 6, no. 3, pp. 594–604, Sep. 2021.

[27] W. Zhang, W. Wang, J. Zhu, and D. Zhao, "Multi-vehicle interaction scenarios generation with interpretable traffic primitives and Gaussian process regression," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Oct. 2020, pp. 1197–1204.

[28] S. Choi, J. Kim, and H. Yeo, "TrajGAIL: Generating urban vehicle trajectories using generative adversarial imitation learning," *Transp. Res. C, Emerg. Technol.*, vol. 128, Jul. 2021, Art. no. 103091.

[29] B. Li and Y. Zhang, "Fast trajectory planning in Cartesian rather than frenet frame: A precise solution for autonomous driving in complex urban scenarios," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 17065–17070, 2020.

[30] T. Gu, J. Snider, J. M. Dolan, and J.-W. Lee, "Focused trajectory planning for autonomous on-road driving," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2013, pp. 547–552.

[31] X. Li, Z. Sun, D. Cao, Z. He, and Q. Zhu, "Real-time trajectory planning for autonomous urban driving: Framework, algorithms, and verifications," *IEEE/ASME Trans. Mechatronics*, vol. 21, no. 2, pp. 740–753, Apr. 2016.

[32] H. Mouhagir, V. Cherfaoui, R. Talj, F. Aioun, and F. Guillemard, "Trajectory planning for autonomous vehicle in uncertain environment using evidential grid," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 12545–12550, Jul. 2017.
[33] B. Li, Y. Ouyang, L. Li, and Y. Zhang, "Autonomous driving on curvy roads without reliance on frenet frame: A Cartesian-based trajectory planning method," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 9, pp. 15729–15741, Sep. 2022.
[34] B. Li, T. Acarman, Y. Zhang, L. Zhang, C. Yaman, and Q. Kong, "Tractor-trailer vehicle trajectory planning in narrow environments with a progressively constrained optimal control approach," *IEEE Trans. Intell. Vehicles*, vol. 5, no. 3, pp. 414–425, Sep. 2020.

FUZHOU ZHAO received the B.S. degree from the Wuhan University of Technology, China, the M.S. degree from the Nanjing University of Aeronautics and Astronautics, and the Ph.D. degree from the Nanjing University of Science and Technology, in 2010. Currently, he is a Visiting Scholar with the School of Vehicle and Mobility, Tsinghua University. He is also the Vice-Professor with the Changshu Institute of Technology. He has authored over 30 articles in China. He is a co-inventor of more than 40 patents. His recent research interests include intelligent vehicle modeling, trajectory decision, and control for autonomous vehicles.

LING HAN received the M.S. and Ph.D. degrees from the Transportation College, Jilin University, China, in 2015. She is currently a Vice-Professor with the School of Mechanical and Electrical Engineering, Changchun University of Technology. Her research interests include intelligent vehicles and autonomous driving.

MINGYANG CUI received the B.Eng. degree from the School of Vehicle and Mobility, Tsinghua University, China, in 2018, where he is currently pursuing the Ph.D. degree with the School of Vehicle and Mobility. His research interests include motion prediction and decision making for intelligent vehicles, especially in consideration of the interaction effect during traffic conflicts.

HEYE HUANG received the bachelor's degree in traffic and transportation engineering from Central South University, Changsha, in 2018. She is currently pursuing the Ph.D. degree with the State Key Laboratory of Automotive Safety and Energy, Tsinghua University. Her current research interests include connected and automated vehicles, risk assessment, decision making, situational awareness, and prediction.

SHAN ZHONG received the Ph.D. degree from Soochow University, in 2017. From March 2018 to September 2018, she was a Visiting Scholar with the School of Computer Science and Engineering, University of Wisconsin-Eau Claire. Currently, she is a Vice-Professor with the School of Computer Science and Engineering, Changshu Institute of Technology. She has presided over a project funded by the National Natural Science Foundation of China. She is an advanced member of the Computer Society of China. Her research interests include reinforcement learning, automatic control, and deep learning.

FEIFEI SU received the B.S. degree in automation from Yanshan University, in 2014, and the M.S. degree from Flinders University, Australia, in 2023. He is currently the Senior Product Manager of Baidu Apollo. His research interests include intelligent robots and automatic driving.

LEI WANG received the B.S. and M.S. degrees in computer application software from Beijing Jiaotong University. He is currently the Manager with Ziqing Intelligent Driving Technology (Beijing) Company Ltd. He has authored more than ten patents and led a team to complete a number of CAVs projects. His research interests include the X-by-wire control of vehicle chassis and the multi-sensors fusion of CAVs.