ISSN:2790-1688 Volume-15-(2025)

Optimal Control Based Trajectory Planning for an Autonomous vehicle on Curvy Roads

Bozhen Li

University Of Sydney

Abstract. We proposed a trajectory optimization method for curvy road environments containing irregular embedded obstacles. A novel framework is developed to ensure safety and smoothness of trajectory by combining a Spatial Safety Corridor (SSC) with an Optimal Control Problem (OCP). The Spatial Safety Corridor (SSC) is first constructed to exclude obstacle regions from the drivable area, forming a collision-free corridor that guides feasible motion. Within this corridor, an Optimal Control Problem (OCP) is formulated based on a kinematic bicycle model to generate a trajectory that satisfies both feasibility and safety constraints. Simulation results illustrate that the proposed framework effectively handle diverse obstacle configurations, generating trajectories that are smooth and safe. Furthermore, the proposed framework demonstrates high computational efficiency, making it suitable for real-time trajectory planning.

Keywords: Trajectory Planning, Optimal Control Problem (OCP), Spatial Safety Corridor (SSC), Rectangle Expansion Algorithm, Curvy Road Environment.

1. Introduction

With the rapid advancement of artificial intelligence (AI), autonomous navigation technology has become a highly regarded research area, playing a crucial role in both civilian and military applications. Autonomous navigation refers to a robotic system's ability to perceive its surroundings, make intelligent decisions, and execute precise movements without human intervention. This technology has been widely applied in indoor environments, such as robotic vacuum cleaners and restaurant delivery robots, but these applications are typically confined to structured environments with clearly defined boundaries. Despite significant progress in autonomous navigation, major challenges remain in unstructured environments, particularly in outdoor navigation. Unlike controlled environments with well-defined paths and boundaries, real-world scenarios often involve irregular terrain, ambiguous navigable areas, and embedded obstacles. These complexities present substantial difficulties for current navigation systems, as they largely rely on structured road maps and predefined constraints. Trajectory planning for autonomous vehicles has been extensively studied through various instance, including sampling-based motion planners, search-based graph methods, and optimization-driven techniques. However, these approaches often struggle to maintain both smoothness and safety in curved road environments, especially when irregular obstacles are present.

Wen et al. [1] addressed the problem of Multi-Agent Path Planning for Car-Like Robots (CL-MAPF). This study proposes a novel CL-MAPF solving method, CL-CBS, which extends the Conflict-Based Search (CBS) framework to accommodate the kinematic constraints of car-like robots. It also introduces a new algorithm, Spatiotemporal Hybrid-State A* (SHA), as a single-agent planner to generate paths that satisfy both kinematic and spatiotemporal constraints while improving computational efficiency at the cost of slight solution optimality **Search-based method:** Ferner et al. [2] addressed the high-dimensional search space problem in multi-robot

systems (MRS) path planning and proposed an *optimized approach combining the M algorithm, Meta-Agent Constraint-Based Search (MA-CBS), and Operator Decomposition (OD)**. This method efficiently finds optimal paths in low-dimensional search spaces, improving computational efficiency and scalability compared to traditional approaches. [6] proposed an implementation of "sub dimensional expansion" for robot configuration spaces that can be represented as a graph, called the M* algorithm. It efficiently finds the minimal-cost path for multi-robot path planning. [7] introduced a new optimal solution method for MAPF—Conflict-Based Search (CBS). This method outperforms the traditional A* approach in various environments and is suitable for real-world multi-robot path planning tasks. [8] proposed. **Optimization based:** Liu et al. [3] addressed the dynamic lane-changing trajectory planning problem for autonomous vehicles and proposed a dynamically decoupled lane-changing trajectory planning method (DDLC) based on comprehensive optimization. This method adjusts trajectories in real-time considering surrounding vehicle states to handle sudden changes while optimizing computations to reduce complexity and improve real-time performance. It effectively generates smooth, safe, and efficient lane-changing trajectories. [5] proposed an automated driving trajectory planning method based on Ordinal Optimization (OO), which used coarse modeling and fast sorting strategies to select near-optimal trajectories efficiently from a set of candidates. This approach reduces computational complexity significantly while ensuring trajectory quality and improving planning efficiency. **Deep learning based**: **Zhao et al. [4]** successfully enhance the adaptability and safety of autonomous driving systems in urban environments by integrating a Bayesian inference-based motion prediction model with a reinforcement learning-driven motion planning algorithm. Tree Policy Planning (TPP) is proposed by **[10]** to address the lack of interpretability and the inability to generate high-quality trajectories in policy planning. This method is interpretable, tunable, and scalable to complex scenarios involving a large number of traffic participants. **[11]** addresses the problem that traditional prediction and planning methods can not sufficiently consider uncertainty information and proposes Uncertainty-Aware Planning (UAP-based Planning). By leveraging uncertainty information to optimize planning decisions, it enhances safety and adaptability, significantly improving prediction accuracy. A trajectory prediction method based on Graph Neural Networks and Spatial-Temporal Convolutional Networks (GSTCN) is proposed by **[12]**. It used GCN to model vehicle interactions, combined CNN and GRU for trajectory encoding and decoding, and designed a weighted adjacency matrix to describe the effect intensity between vehicles. This method is better than existing approaches in prediction accuracy, model size, and inference speed. Existing trajectory prediction methods focus only on generating individual agent trajectories, which lead to trajectory collision problem. **[13]** mentions a novel interactive traffic simulation model, InterSim, which outperforms existing methods in terms of simulation realism, interaction consistency, and computational efficiency. A Path-Based Prediction (PBP) method is proposed by [14] to address the issue of poor map compliance in goal-based trajectory prediction, leading to trajectories deviating from lanes or violating traffic rule. This approach provides a stronger inductive bias, ensuring trajectory predictions better adhere to road structures. **Reinforcement Learning based:** [8] proposes a Reinforcement learning-based predictive behavior planning framework that achieves human-like decision-making by leveraging conditional motion prediction and inverse reinforcement learning to select the optimal trajectory, enhancing the safety and interpretability of autonomous driving decision-making. [9] proposes a policy gradient-based imitation learning method for autonomous

driving in complex urban environments, utilizing a differentiable simulator for closed-loop training to enhance training efficiency and generalization ability.

Main contribution: existing methods are primarily designed for structured road environments, making them cannot be directly applied to trajectory planning on curvy roads. To bridge this gap, we introduce a new trajectory planning framework that integrates a spatial corridor-based obstacle avoidance strategy and an interior-point optimization approach. Our framework accounts for the challenges posed by unstructured curvy roads, ensuring feasible and smooth trajectory generation while significantly improving computational efficiency.

2. Optimal Control Problem Formulation

Achieving smooth and safe trajectory planning in curvy road environments remains a significant challenge, especially when dynamic constraints and obstacle avoidance must be simultaneously addressed. To formulate this problem within the optimal control framework, it is crucial to develop an accurate kinematic model and clearly define the state variables, control inputs, and associated safety and dynamic constraints. In the following subsections, we detail the complete problem formulation, beginning with the vehicle kinematics and subsequently introducing the relevant dynamic and collision avoidance constraints.

2.1 Vehicle Kinematics

Suppose there is a Connected and Autonomous Vehicle (CAV) passing through the curvy road.

$$\frac{d}{dt}x(t) = \begin{bmatrix} \dot{x}\ (t) \\ \dot{y}\ (t) \\ \dot{\theta}\ (t) \\ \dot{v}\ (t) \\ \dot{\phi}(t) \end{bmatrix} = \begin{bmatrix} v(t)\cdot cos\theta(t) \\ v(t)\cdot sin\theta(t) \\ \frac{v(t)}{L} tan\delta(t) \\ a(t) \\ \omega(t) \end{bmatrix}, \in [0, t_f] \ (1)$$

x_t, y_t are the CAV's position, while θ_t is the heading angle of the vehicle. The longitudinal velocity is $v(t)$, and $\delta(t)$ is the steering angle of the CAV. The longitudinal acceleration is $a(t)$, and the steering rate $\dot{\phi}(t)$, where $\omega(t) = \dot{\phi}(t)$. L is the vehicle wheelbase. Finally, t_f represents the final time of the entire planning process.

2.2 Dynamic constrains

$$\begin{cases} |a(t)| \le a_{max}, \forall t \in [0, t_f] \\ |\delta(t)| \le \delta_{max}, \forall t \in [0, t_f] \\ |\theta(t)| \le \theta_{max}, \forall t \in [0, t_f] \\ |v(t)| \le v_{max}, \forall t \in [0, t_f] \end{cases} (2)$$

To ensure the physical feasibility and safety of the planned trajectory, several constraints are imposed on the CAV's motion. First, the longitudinal acceleration $a(t)$ must below the maximum acceleration a_{max}, preventing sudden acceleration and braking. Second, the steering angle $\delta(t)$ must remain within $\pm\delta_{max}$ to avoid sharp turns. Additionally, the heading angle $\theta(t)$ is constrained by the maximum range θ_{max} to avoid the vehicle changing direction extremely. Finally, the velocity v(t) is restricted to the range$[0,v_{max}]$ to ensure forward motion within speed limits.

2.3 Collision Avoidance Constraint:

To ensure safety in curvy road environments, a collision avoidance constraint is introduced into the trajectory planning process. The CAV is modeled as a circle with radius r_{car}, and both the road boundaries and obstacle shapes are discretized into a finite set of points. N is the total number of discrete points along the road boundary, while M is the total number of discrete points of obstacles boundary.

$$B_{road} = \{(x_{b,i}, y_{b,i}) | i = 1,2,,N\}, x, y \in R \ (3)$$

$$B_{obstacles} = \{(x_{0,i}, y_{o,i}) | i = 1,2,,M\}, \ x, y \in R \ (4)$$

To unify the road boundary and obstacle constraints, we define the environment point set as the union of both discrete boundaries:

$$B_{env} = B_{road} \cup B_{obstacles} = \{(x_{e,i}, y_{e,i}) | i = 1,2,\ldots,P\}, \ x, y \in R \ (5)$$

where P = N + M is the total number of sampled points.

The CAV's center position (x(t), y(t)) must maintain a minimum distance from all points in B_{env}, which leads to the following constraint:

$$\sqrt{\left(\mathrm{x(t)} - \mathrm{x_{e,i}}\right)^2 + \left(\mathrm{y(t)} - \mathrm{y_{e,i}}\right)^2} \geq r_{car,i}, \forall \mathrm{t} \in [0, \mathrm{t_f}], \forall \mathrm{i} \in 1,2,\ldots,P \ (6)$$

where (x(t), y(t)) denotes the CAV's position at time t, and $(x_{e,i}, y_{e,i})$ represents the coordinate of the i-th environment point. The index $i \in \{1, 2, ..., P\}$ enumerates all sampled points from road and obstacle boundaries.

2.4 Initial and Terminal Moment Constraints:

We introduced the initial and terminal moment constraints to ensure the vehicle start and end at pre-defined position. The CAV is required to begin from the initial state

$$X_{initial} = X_0 = [x_0, y_0, \theta_0, v_0, a_0]^T \ (7)$$

and reach the final state

$$X_{final} = X_f = [x_f, y_f, \theta_f, v_f, a_f]^T (8)$$

Here, (x_0, y_0) and (x_f, y_f) denote the initial and final positions, respectively; θ_0, v_0, and a_0 are the initial heading angle, velocity, and acceleration; θ_f, v_f, and a_f define the required terminal state.

2.5 Optimization Objective and Cost Function

To compute a smooth and safe trajectory, we formulate an optimal control problem that minimizes a cost function over the planning horizon $t \in [0, t_f]$. The cost function penalizes excessive longitudinal acceleration $a(t)$ and angular velocity $\omega(t)$ to promote trajectory smoothness and stability. Specifically, the objective is given by:

$$J = \int_{t=0}^{t_f} \left(\sum_{w=1}^{N_v} (a_{w_1}^2(t) + \omega_{w_2}^2(t)) \right) dt, W \geq 0 \ (9)$$

where w_1 and w_2 are non-negative weighting coefficients. A higher value of w_1 emphasizes acceleration smoothness, while w_2 penalizes rapid steering changes. In some

scenarios, an additional term may be introduced to encourage forward progress, with an overall form:

$$w=Ww_1+w_2 \quad (10)$$

The optimization is subject to constraints on vehicle kinematics, collision avoidance, road boundaries, and initial and terminal states, ensuring that the resulting trajectory is feasible, dynamically consistent, and collision-free.

Table I Summary of Optimization Constraints

Constraint Type	**Description**	**Related Equations**	**Constraint Type**
Vehicle Kinematics	Bounds on acceleration, steering, heading, and velocity	(2)	Vehicle Kinematics
Collision Avoidance	Maintain minimum distance to environment points	(6)	Collision Avoidance
Initial and Terminal States	Enforce starting and ending conditions	(7), (8)	Initial and Terminal States

Remark: This study formulates the trajectory planning problem as an Optimal Control Problem (OCP), aiming to optimize trajectory smoothness and safety while satisfying vehicle dynamics, road boundary, and obstacle avoidance constraints. The CAV is modeled as a circle with radius r, and collision detection is performed by ensuring that the distance between the vehicle center and each environment point exceeds r. When the number of environment points is large, the repeated evaluation of distance constraints can still lead to significant computational burdens. Therefore, directly applying this method to real-time trajectory planning may still lead to significant computational burdens.

3. Construction of the Safety Corridor

3.1 Motivation

Traditional collision detection methods in autonomous vehicle trajectory planning typically discretize road boundaries into dense sets of points and perform individual distance checks, which significantly increases computational cost. This challenge becomes critical in nonlinear optimal control problems, where CAV dynamics must be evaluated at each planning step. To address these challenges and improve computational efficiency, we introduced the concept of a Spatial Safety Corridor (SSC), which replaces traditional pointwise collision constraints with a simplified structure that constrains the CAV's trajectory to lie within a precomputed, collision-free region. The SSC framework reduces the need for repeated geometric intersection checks, enabling more efficient and scalable planning in real-time applications.

3.2 Initial Trajectory and Road Modeling

As shown by the light blue dashed curve in Fig. 1, an initial feasible trajectory is generated through cubic B-spline interpolation and uniform resampling. This curve serves both as the initial solution and the centerline of the road. By offsetting the centerline perpendicularly to both sides

with a constant road width, the left and right road boundaries are constructed, forming the original road geometry.

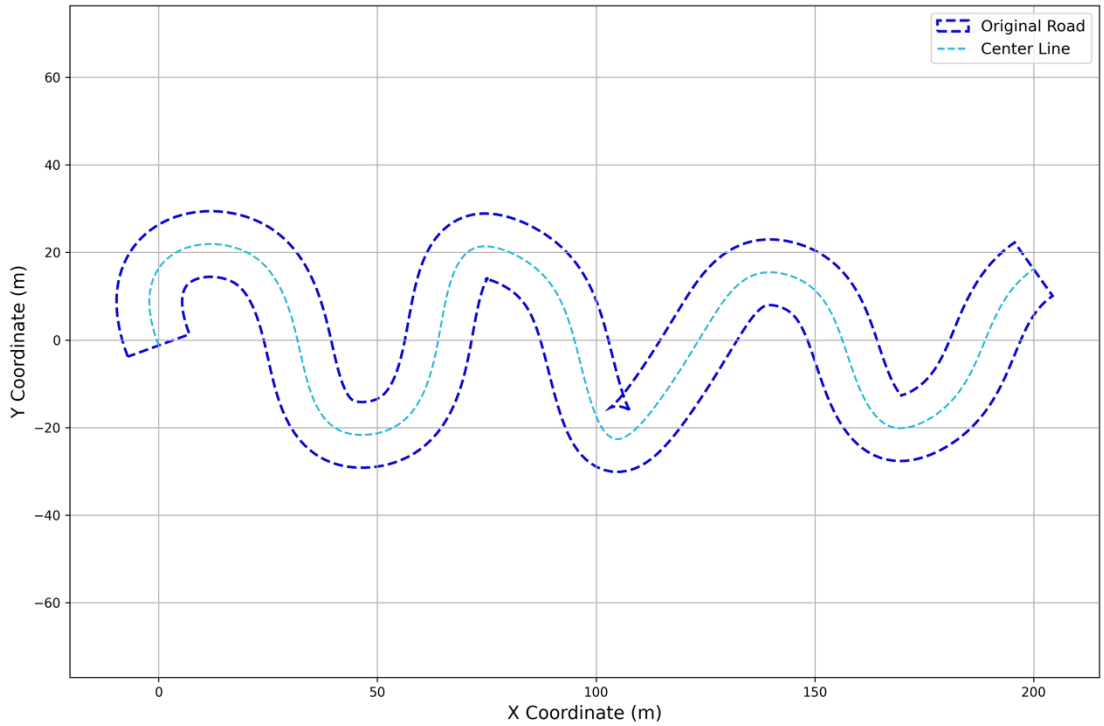

Fig. 1. Road Geometry Representation with Centerline.

3.3 Vehicle Modeling

According to geometric approximation theory, any convex polygon can be represented as the union of multiple disks. Based on this principle, the vehicle is approximated as a single circle with radius r_{car} to simplify collision detection and geometric operations. Building on this circular abstraction, we model the CAV's motion dynamics using the Kinematic Bicycle Model, which captures the vehicle's nonlinear kinematic evolution while maintaining computational efficiency. Fig. 3 illustrates the Kinematic Bicycle Model. The CAV is approximated as a rigid body with front and rear wheels aligned along a longitudinal axis. The state of the CAV is defined by its position (x,y), heading angle θ, and longitudinal velocity v. The steering angle δ controls the front wheel orientation relative to the vehicle axis, generating a lateral velocity component that induces a yaw rate ω. L is the wheelbase length, and ICR (Instantaneous Center of Rotation) represents the center of the circular path at each instant.

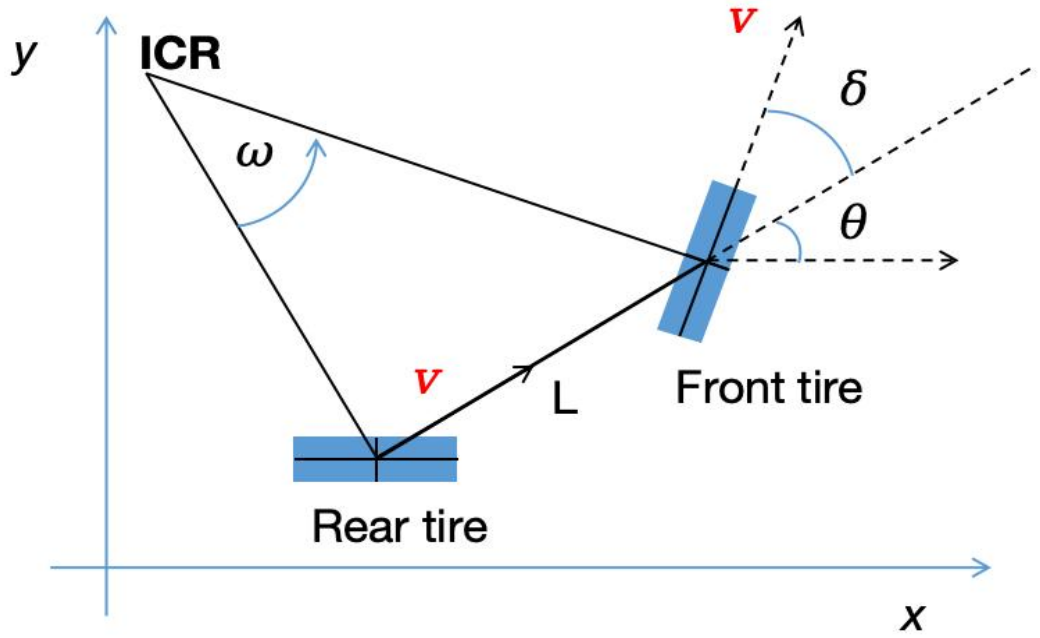

Fig. 2. Bicycle Kinematic Model.

3.4 Safe Region Construction

Based on the circular vehicle approximation, collision avoidance is formulated by ensuring that the distance between the vehicle center and any obstacle or boundary point remains greater than r_{car}. Consequently, the road boundaries are inwardly offset by r_{car}, forming an inflated map that defines the feasible region for the vehicle center, as illustrated by the shaded area in Fig. 3. This inflated road geometry guarantees that if the CAV's center stays within the feasible region, the entire vehicle body remains collision-free.

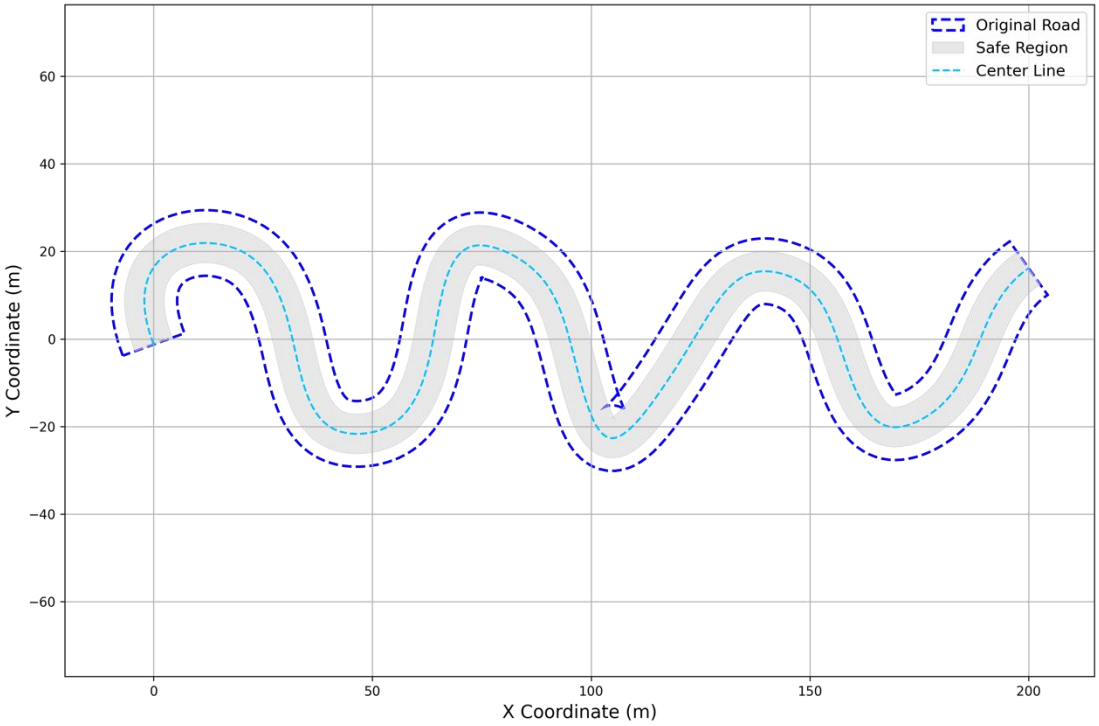

Fig. 3. Road Geometry Representation with Centerline, Safe Region, and Original Road.

3.5 Spatial Safety Corridor Construction

After constructing the safe region, we build a Spatial Safety Corridor (SSC) based on the reference trajectory, i.e., the centerline. The centerline is uniformly resampled to generate a sequence of discrete waypoints, each corresponding to a specific planning time step. At each waypoint, we expand in four directions—left, right, forward, and backward—to determine the largest inscribed rectangle that remains entirely within the safe region. By connecting these rectangles along the path, a continuous strip-like corridor is formed, as shown in Fig. 4.

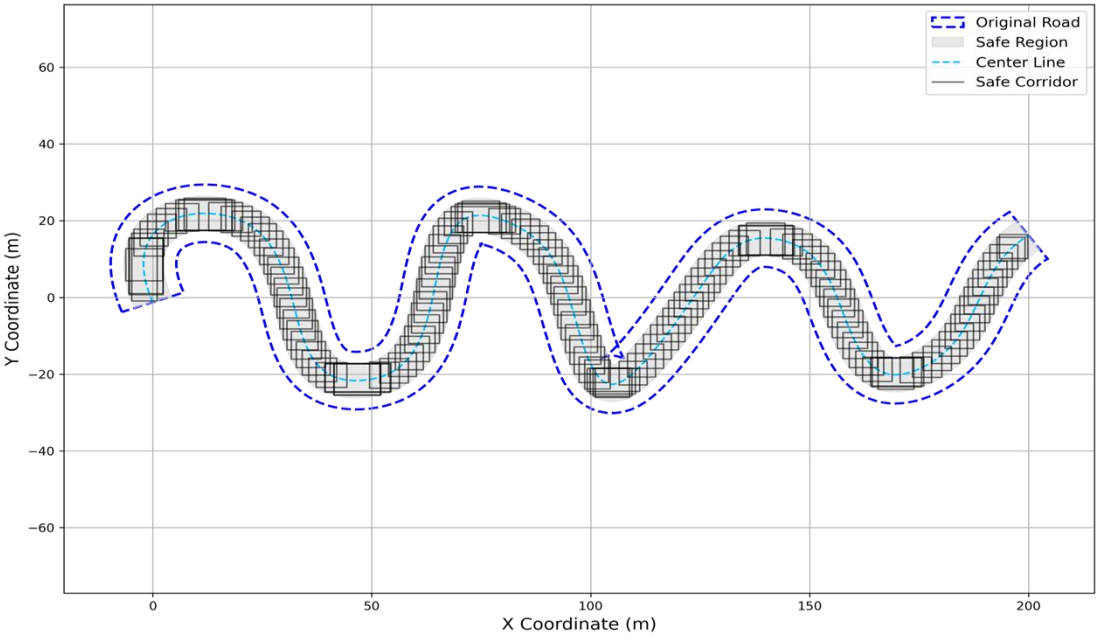

Fig. 4. Illustration of the Spatial Safety Corridor.

3.5.1 Safe Rectangle Generation Process

The detailed workflow for generating safe rectangles is shown in Fig. 5. Starting from each waypoint, a candidate rectangle is initialized and iteratively expanded along the normal and tangential directions of the path. Expansion continues until any edge reaches the boundary of the safe region. The final rectangles are guaranteed to be collision-free by construction, ensuring full containment within the precomputed inflated map.

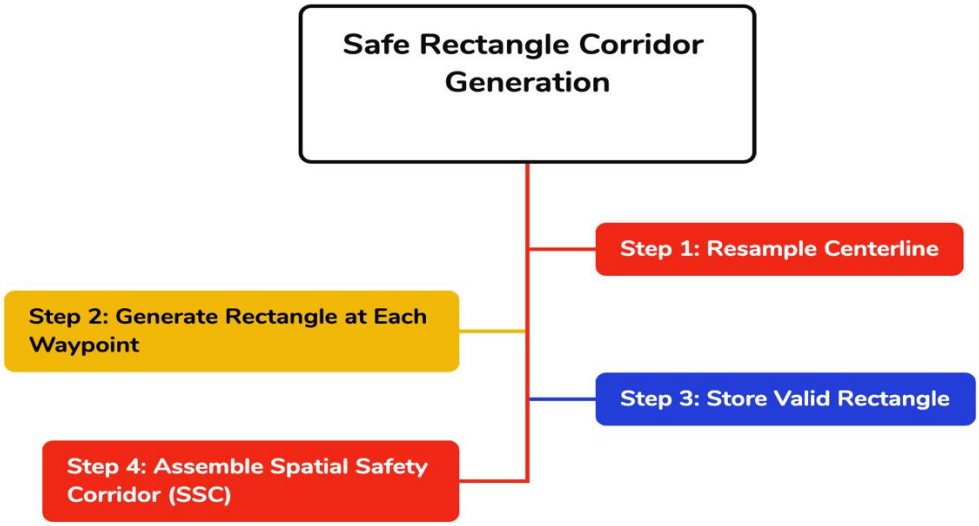

Fig. 5. Flowchart of Safe Rectangle Region Generation

3.5.2 Rectangle Expansion Algorithm

To efficiently compute the maximum inscribed rectangle at each waypoint, we employ a lightweight expansion strategy, outlined as follows:

Algorithm 1: Rectangle Expansion in a Polygon

Input: Initial rectangle R_0, polygon region P, step size Δt	
Output: Expanded maximal rectangle R	
1	$R \leftarrow R_0$
2	for direction in {left, right, up, down} do
3	while True do
4	$R' \leftarrow$ attempt_expand(R, direction, Δt)
5	if R' ⊆ P then
6	$R \leftarrow R'$
7	else
8	break
9	return R

Here, rect denotes the current candidate rectangle and polygon represents the safe region. In each direction, the rectangle is expanded incrementally. If the expanded rectangle remains fully within the safe region (as verified by is_inside_polygon()), the expansion is accepted; otherwise, expansion in that direction is stopped. This process continues until no further enlargement is possible.

3.5.3 Transformation of Collision Avoidance Constraints via SSC

Traditional collision avoidance constraints require maintaining a minimum distance between the CAV and all obstacle or boundary points at every planning step. Although accurate, this approach introduces a large number of nonlinear distance constraints, leading to substantial computational overhead, especially in environments with dense obstacles. In contrast, by constructing a Spatial Safety Corridor (SSC), collision avoidance is fundamentally simplified. Instead of enforcing multiple distance constraints, it suffices to ensure that the vehicle center (x_k,y_k) remains within a precomputed safe rectangle at each time step:

$$x_k \in \left[x_k^{\text{left}}, x_k^{\text{right}}\right], \quad y_k \in \left[y_k^{\text{down}}, y_k^{\text{up}}\right], (11)$$

This replacement transforms a high-dimensional, nonlinear collision checking problem into a sequence of simple box constraints, greatly reducing computational complexity and enabling more efficient real-time trajectory optimization.

3.6 Integration into the Optimal Control Framework

To incorporate the continuous-time objective defined in (9) into a numerically solvable optimal control framework, we apply a global discretization over a finite planning horizon. The continuous integral is thus approximated by a finite sum of the control effort at each step:

$$J \approx min \sum_{k=0}^{N-1} \Delta t \left(w_1 a_k^2 + w_2 \omega_k^2\right), (12)$$

where $\Delta t=t_f/N$ is the uniform time step.

For simplicity, Δt can be absorbed into the weighting coefficients without loss of generality, resulting in the final discrete objective:

$$J = min \sum_{k=0}^{N-1} \left(w_1 a_k^2 + w_2 \omega_k^2\right), (13)$$

Subject to:

$$\begin{cases} |a(t)| \le a_{max}, \forall t \in [0, t_f] \\ |\delta(t)| \le \delta_{max}, \forall t \in [0, t_f] \\ |\theta(t)| \le \theta_{max}, \forall t \in [0, t_f] \\ |v(t)| \le v_{max}, \forall t \in [0, t_f] \end{cases} (2)$$

$$X_{initial} = X_0 = [x_0, y_0, \theta_0, v_0, a_0]^T (7)$$

$$X_{final} = X_f = \left[x_f, y_f, \theta_f, v_f, a_f\right]^T (8)$$

$$x_k \in \left[x_k^{left}, x_k^{right}\right], \quad y_k \in \left[y_k^{down}, y_k^{up}\right] (9)$$

where a_k and ω_k are the acceleration and yaw rate at step k, and w_1,w_2 are their corresponding weighting coefficients. This cost function formulation promotes both smoothness and energy efficiency throughout the trajectory. **Remark**: The introduction of time-indexed rectangular constraints derived from the SSC provides a computationally efficient alternative to conventional nonlinear collision avoidance formulations. By precomputing feasible spatial corridors offline, the optimization problem eliminates geometric computations and significantly reduces problem

non-convexity. The SSC approach is particularly beneficial for planning in environments with curvy roads and embedded static obstacles, where the obstacle shapes are irregular but fixed.

4. Simulation and Results

4.1 Experimental Setup

To evaluate the proposed trajectory planning method, we first construct a simulation environment. The road centerline is generated based on a perturbed sinusoidal curve (14), resampled using cubic B-spline interpolation. Specifically, we sample 20 control points along the X-axis from 0 to 200 meters, and define the Y-axis with a sinusoidal shape perturbed by uniform noise:

$$[x_i=\text{linspace}(0,200,20),\quad y_i=\sin(x_i/10)\times 20+\epsilon_i,\quad \epsilon_i\sim U(-5,5)],\ (14)$$

where ϵ_i represents uniformly distributed random noise to simulate realistic lateral perturbations.

A fixed road width of $W_{\text{road}} = 15\,\text{m}$ is adopted, and the safe region is defined by offsetting the road boundaries inward by the vehicle's safety margin. The vehicle is modeled using a kinematic bicycle model with a wheelbase $L = 2.5\,\text{m}$, five state variables, and two control inputs. The trajectory planning horizon N is dynamically determined based on the total path length and the reference speed (15), resulting in a total of N = 405 steps and a total simulation time of approximately $t = 40.5\,\text{s}$.

$$N = \left\lceil \frac{L_{\text{path}}}{v\cdot\Delta t} \right\rceil \quad (15)$$

The complete configuration of the modeling environment, solver settings, computational platform and other settings are summarized in Table I.

Table II Baseline Simulation Environment Settings

Parameter(s)	Description	Setting(s)
L_{road}	Centerline X-axis range	200 m
Number of control points	Number of sampled control points	20
Centerline Y−axis generation	Centerline generation formula	*Equation* (14)
Interpolation method	Centerline interpolation method	*Cubic B−spline*, *max inter−point dist*
W_{road}	Road width	15 m
r_{vehicle}	Vehicle safety margin (radius)	3 m
Δt	Discrete time step	0.1 s
Trajectory planning horizon N	Number of planning steps	$\left\lceil \frac{L_{\text{path}}}{v\cdot\ \Delta t} \right\rceil$=405 *steps*
Initial state X_0	Initial vehicle state	$[-0.56,0.29,1.90,10.0,0.0]^T$
Terminal state X_f	Terminal vehicle state	$[199.24,15.63,0.64,10.0,0.0]^T$
L	Wheelbase of the vehicle	2.5 m

v_{max},a_{max}	Maximum velocity and acceleration	10 m/s,±3 m/s^2
ϕ_{max},ω_{max}	Maximum steering angle and steering rate	±0.6 rad,±0.3 rad/s
θ *constraint*	Heading angle constraint	$[-\pi,\pi]$
$v(t)$ *constraint*	Velocity constraint	[0,15] m/s
$a(t)$ *constraint*	Acceleration constraint	[0,3] m/s^2
t	Total time	40.5 s
Solver	Optimization solver	IPOPT 3.14 via CasADi 3.7.0
Computational environment	Hardware setup	Google Colab, 2-core CPU, 12GB RAM
Python version	Programming language version	Python 3.9

4.2 Trajectory Planning in Obstacle-Free Environment

To obtain the optimal trajectory, we solve the discrete-time optimal control problem (13), subject to vehicle dynamics, physical limits, and spatial constraints specified in (2), (7), (8), (11) and summarized in Table I. The trajectory is discretized over N=405 steps with a fixed time step of Δt=0.1 s, guiding the CAV from the initial state X_0 to the terminal state X_f along the resampled centerline. In the baseline scenario without obstacles, the proposed trajectory planning framework successfully generates a smooth, dynamically feasible, and spatially constrained optimal trajectory. As shown in Fig. 6, the optimized trajectory (red curve) remains fully contained within the Spatial Safety Corridor (SSC) (gray shaded region), successfully enforcing both dynamic feasibility and spatial safety constraints.

The optimization was performed using IPOPT via CasADi, achieving convergence within 49 iterations and completing in 4.53 seconds. Simulation results indicate that the SSC-based spatial constraints effectively limit the CAV's feasible region while preserving continuity and smoothness of the trajectory. Even in areas with high curvature, the optimized trajectory maintains smooth transitions without abrupt changes. This lays a solid foundation for extending the method to more complex scenarios involving obstacle avoidance.

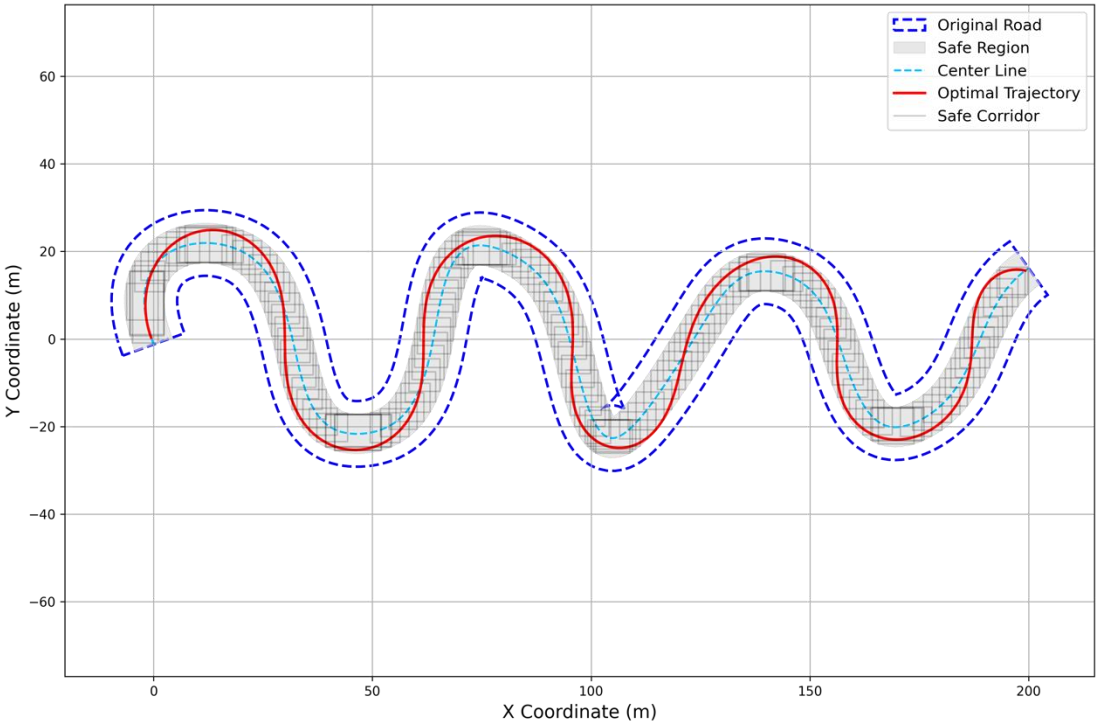

Fig. 6. Safe Corridor with Optimal Trajectory (Rectangles Sampled)

4.3 Trajectory Optimization with Embedded Obstacles and SSC

Building upon the scenario without obstacles in Section 4.2, this section investigates the generalizability of the proposed method in more complex environments. Specifically, five irregular polygonal obstacles are embedded along the road, creating challenging collision avoidance scenarios. As shown in Fig. 7, the optimized trajectory (red curve) remains fully contained within the SSC region while successfully avoiding all embedded obstacles. The blue dashed curves represent the original road boundaries, and the yellow polygons illustrate the obstacle locations. The optimal trajectory is solved using the same five-dimensional kinematic bicycle model and the CasADi + IPOPT framework introduced in Section 4.1. Optimization convergence is achieved within 54 iterations, requiring approximately 6.08 seconds.

These results demonstrate that the SSC-OCP can generate smooth, safe, and dynamically feasible trajectories even in the presence of irregular, non-convex obstacles.

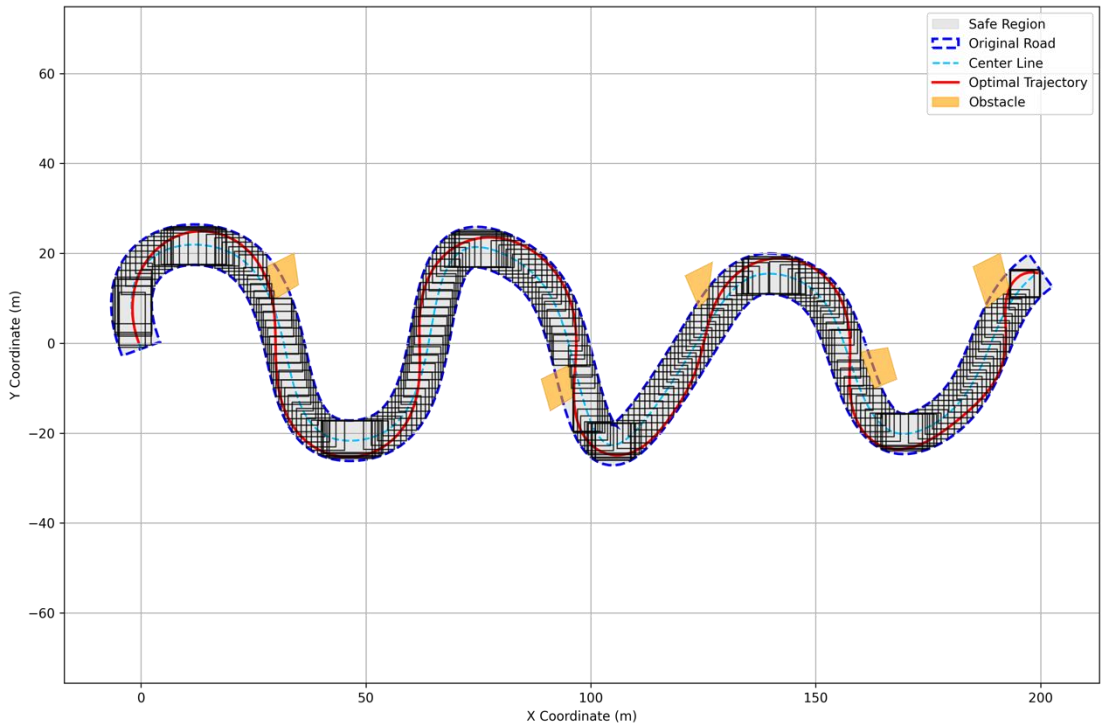

Fig. 7. Optimal Trajectory within Safety Corridor under Static Obstacles

4.4 Computational Environment & Performance

All trajectory optimization experiments were conducted on Google Colab, a cloud-based Jupyter notebook environment running Python 3.9. The modeling was implemented using CasADi 3.7.0, and the nonlinear programming problem was solved via IPOPT 3.14. The backend system typically provides 2-core CPUs and 12GB of RAM on a virtualized Linux container. In the obstacle-free scenario, the solver converged in 49 iterations, with a total wall-clock time of approximately 0.88 seconds.

Conclusion

In this study, we addressed the trajectory planning problem in complex road environments by proposing a novel approach that integrates a Spatial Safety Corridor (SSC) with an Optimal Control Problem (OCP). This method specifically tackles dynamic planning needs when irregular obstacles are present within the roadway. By preprocessing the obstacle and drivable area boundaries to create a continuous and safe driving corridor (SSC), we simplified the spatial constraints in trajectory optimization, enhancing computational efficiency. Simulation results indicate that the proposed method reliably generates smooth and safe trajectories that satisfy real-time dynamics constraints. Compared to traditional pointwise distance-based approaches, our method demonstrates notable improvements in modeling complexity and optimization efficiency, indicating strong generalization capabilities suitable for practical applications in autonomous driving and robotic navigation systems.

References

[1] L. Wen, Y. Liu, and H. Li, "CL-MAPF: Multi-Agent Path Finding for Car-Like Robots with Kinematic and Spatiotemporal Constraints," *Robotics and Autonomous Systems*, vol. 150, p. 103997, Dec. 2022, doi: 10.1016/j.robot.2021.103997.

[2] C. Ferner, G. Wagner, and H. Choset, "ODrM*: Optimal multirobot path planning in low dimensional search spaces," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Karlsruhe, Germany, May 2013, pp. 3854–3860.

[3] Y. Liu, B. Zhou, X. Wang, L. Li, S. Cheng, Z. Chen, G. Li, and L. Zhang, "Dynamic lane-changing trajectory planning for autonomous vehicles based on discrete global trajectory," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 7, pp. 8513-8522, Jul. 2022, doi: 10.1109/TITS.2021.3083541.

[4] Y. Du, K. Tang, L. Li, and M. Tomizuka, "Self-Adaptive Motion Prediction-Based Proactive Motion Planning for Autonomous Driving in Urban Environments," IEEE Transactions on Intelligent Vehicles, vol. 6, no. 1, pp. 137-148, Mar. 2021, doi: 10.1109/TIV.2020.3048896.

[5] X. Fu, Y. Jiang, D. Huang, K. Huang, and J. Wang, "Trajectory planning for automated driving based on ordinal optimization," Tsinghua Science and Technology, vol. 22, no. 1, pp. 62–72, Feb. 2017, doi: 10.23919/TST.2017.7886772.

[6] G. Wagner, H. Choset, M*: A complete multirobot path planning algorithm with performance bounds, in: 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, 2011, pp. 3260–3267.

[7] G. Sharon, R. Stern, A. Felner, N.R. Sturtevant, Conflict-based search for optimal multi-agent pathfinding, Artificial Intelligence 219 (2015) 40–66

[8] Z. Huang, H. Liu, J. Wu, and C. Lv, "Conditional predictive behavior planning with inverse reinforcement learning for human-like autonomous driving," IEEE Transactions on Intelligent Transportation Systems, 2023.

[9] O. Scheel, L. Bergamini, M. Wolczyk, B. Osinski, and P. Ondruska, "Urban driver: Learning to drive from real-world demonstrations using policy gradients," in Conference on Robot Learning. PMLR, 2022, pp.718–728.

[10] Y. Chen, P. Karkus, B. Ivanovic, X. Weng, and M. Pavone, "Tree-structured Policy Planning with Learned Behavior Models," arXiv preprint arXiv:2301.11902, 2023.

[11] J. Hardy and M. Campbell, "Contingency planning over probabilistic hybrid obstacle predictions for autonomous road vehicles," in 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2010, pp. 2237–2242.

[12] Z. Sheng, Y. Xu, S. Xue, and D. Li, "Graph-based spatial-temporal convolutional network for vehicle trajectory prediction in autonomous driving," IEEE Transactions on Intelligent Transportation Systems, vol. 23, no. 10, pp. 17 654–17 665, 2022.

[13] J. Gu, C. Sun, and H. Zhao, "Densetnt: End-to-end trajectory prediction from dense goal sets," in Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 15 303–15 312.

[14] S. Afshar, N. Deo, A. Bhagat, T. Chakraborty, Y. Shao, B. R. Buddharaju, A. Deshpande, and H. Cui, "PBP: Path-based trajectory prediction for autonomous driving," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2024.