

Trajectory Planning for Lane Change Scenarios with Dynamic Environment

1st Jianyu Huang*

Graduate School/Faculty of ISEE

Kyushu University

Fukuoka, Japan

huang.jianyu.757@s.kyushu-u.ac.jp

*Corresponding author

2nd Yutaka Arakawa

Graduate School/Faculty of ISEE

Kyushu University

Fukuoka, Japan

arakawa@ait.kyushu-u.ac.jp

Abstract—Automated driving technology has garnered increasing attention due to its potential to reduce traffic accidents, alleviate congestion, and enhance travel convenience. As a key component of automated driving technology, trajectory planning plays a pivotal role in determining feasible trajectories in different traffic scenarios. However, there is currently insufficient research on the performance of different curve models in lane-changing scenarios. In addition, the conventional speed planning method (DP&QP) often ignores vehicle kinematics and traffic scenarios when generating convex spaces, resulting in inefficiency and longer run time. In path planning, this paper compared the performance of Dubins, Sine, Bézier, and B-spline curves in lane change scenarios, ultimately selecting the B-spline curve as the lane change path model to ensure that the vehicle remains close to the center line of the road after a lane change. In speed planning, we introduce a novel method for directly generating a convex space, taking into account the kinematics of the ego vehicle and the current traffic conditions. This approach allows for adaptive adjustments to the speed planning scheme based on the traffic situation. Compared with the conventional trajectory planning method ((DP&QP)), this approach significantly reduces the run time by 32.7%.

Index Terms—Autonomous driving, Path planning, Speed planning, Cost function, Quadratic program

I. INTRODUCTION

As a critical component of the autonomous driving system, the trajectory planning module is primarily responsible for generating a safe, efficient, and comfortable drivable trajectory based on information from the localization, perception, and prediction modules [1]. Trajectory planning can be divided into global trajectory planning and local trajectory planning. Global trajectory planning focuses on generating the trajectory for the journey from the starting point to the destination. In contrast, local trajectory planning, based on the existing global path, involves planning the vehicle's specific actions in real time. It addresses tasks such as avoiding dynamic and static obstacles, overtaking, and lane changing within a short time frame. This paper focuses on local trajectory planning for lane change scenarios [2].

A. Related Work

Local trajectory planning can be further categorized into the following main groups: sampling-based approaches,

graph search-based approaches, and optimization-based approaches [3], [4], [5].

The sampling-based approach involves randomly or selectively sampling points in the planning space and then connecting these points to form trajectories [6]. This method is suitable for various complex planning problems, including high dimensionality, multiple constraints, and dynamic environments. One of the most typical algorithms is the Rapidly Exploring Random Tree (RRT) and many researchers have proposed various improved versions in recent years [7]. However, since the sampled points are generated by random or limited sampling, the quality of the generated trajectories is unstable and strongly influenced by the sampling space [8]. In addition, due to the randomness of the sampling, the computational cost can be high, and real-time performance cannot be guaranteed when dealing with complex scenarios.

The graph search-based approach involves transforming the planning space into a node graph and then searching for an optimal trajectory on this graph based on a heuristic function. Among the methods based on graph search, the Dijkstra is one of the most representative algorithms and it finds the shortest trajectory based on the principle of breadth-first search. The A* algorithm adds a heuristic function to the Dijkstra algorithm, allowing a faster search of the trajectory. Various implementations like D* and D* Lite have been proposed to enhance search efficiency in more complex scenarios [9], [10]. Although graph search algorithms exhibit strong searching capabilities and high efficiency, the trajectories they generate often lack smoothness and may not comply with vehicle kinematic or dynamic constraints. As a result, they typically require post-processing for smoothing and further optimization before they can be directly employed in path planning.

The optimization-based approach formulates the path planning problem as a convex optimization problem with multiple constraints. Typically, a cost function is defined based on the requirements of the planning scenario, and the optimal trajectory is obtained by minimizing the value of the cost function. Huang et al. adopted a polynomial curve to model the planned path and designed a safety indicator in the cost function to evaluate the influence of the obstacle size [11]. Xu et al. proposed a novel method to generate convex space for solving

the minimum-time speed planning problem [12]. Compared with the other two approaches, the optimization-based method can achieve higher efficiency, and low computation cost and has been widely utilized.

B. Motivations

The PVD trajectory planning approach decomposes the three-dimensional trajectory problem into two-dimensional path planning and one-dimensional speed planning [13]. This significantly reduces complexity, ensuring real-time performance. This paper follows the PVD trajectory planning framework.

In this paper, the scenario researched is a lane change, as shown in Figure 1. The ego vehicle needs to avoid the upcoming vehicles while changing lanes. In the case of lane changing, the choice of curve model in path planning is crucial, in guiding the vehicle to change lanes and at the same time need to ensure that the steering wheel is back to the right after changing lanes, but a lot of current research does not take this requirement into account when choosing the curve model. In addition, the most typical and widely used method in speed planning is first to search the $s - T$ graph using the dynamic planning (DP) algorithm, generate the convex space, and then use the quadratic planning (QP) algorithm to solve the optimal speed profile in the convex space. However, this method primarily tackles the speed planning problem from a purely mathematical point of view. In the process of creating the convex space, it ignores the current traffic scenario, the vehicle's motion state and kinematic characteristics. This leads to the complication of a relatively simple problem, resulting in a waste of computational resources.

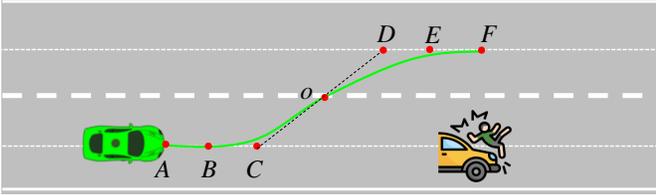


Fig. 1. An Illustration of the Lane Change Scenario with B-spline Curve.

C. Contributions

In path planning, we compared the performance of different curve models in lane-changing scenarios. Among them, we selected the B-spline curve as the lane-changing path model based on the requirements of continuous curvature and zero curvature at the starting and merging points.

In speed planning, we predict the future motion states of the ego vehicle based on its current motion state and kinematic characteristics. This information directly defines the convex space for speed planning within the $s - T$ graph.

D. Organization

The rest of this paper is organized as follows: The path planning method and speed planning of the lane change

scenario are shown separately in Section II and Section III. The simulation results are discussed in Section IV, and Section V concludes the paper and explores the potential directions for future research.

II. PATH PLANNING

A. Curve Model Selection

The commonly used curve models for path planning include the Dubins curve, Sine function curve, Bessel curve, B-spline curve, etc. However, which curve model is more suitable for lane-changing scenarios has not been fully explored in current research. Therefore, this paper compares the performance of the above curve models in lane-changing scenarios.

As shown in Figure 2, the Dubins curve, sine function curve, Bessel curve and B-spline curve can all realize the lane change task and the generated paths are smooth with no significant differences in path shape.

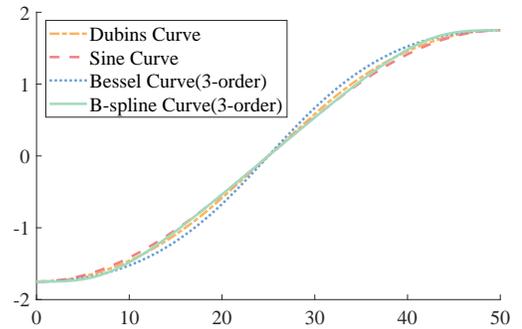


Fig. 2. Comparison of Lane Change Paths with Different Curve Models.

Figure 3 shows that the Dubins curve generates a path with two abrupt changes in curvature. Following this path would require the vehicle to stop at these discontinuities, adjust the steering and then continue, which doesn't meet practical driving requirements. In addition, the Sine and Bessel curves produce lane change paths with continuous curvature but non-zero curvature at the start and end points. This prevents the steering wheel from returning to its original position after merging and causes the vehicle to deviate from the centre line of the road. Therefore, the 3rd-order B-spline curve is chosen as the lane change model in this paper.

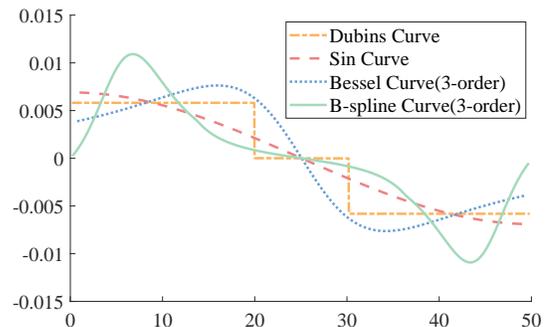


Fig. 3. Curvature Comparison of Different Curve Models in the Lane Change Scenario.

B. Candidate Path Set Generation

As shown in Figure 1, the lane change path consists of two parts: \widehat{ABCO} and \widehat{ODEF} . To simplify the path-solving process, we assume that the \widehat{ABCO} and the \widehat{ODEF} are centrally symmetric with respect to point O . Therefore, it is only necessary to determine \widehat{ABCO} . Since the position of point A depends on the current location of the ego vehicle, and point B is the midpoint between points A and C . Thus, the shape of \widehat{ABCO} depends solely on the positions of points C and O . By altering the positions of points C and O , a series of candidate path sets can be generated.

C. Cost Function

To ensure the smoothness and efficiency of the generated lane-change path, the cost function primarily considers the curvature and length of the path, denoted as Equation 1.

$$J_{\text{path}} = w_1 \sum_{i=1}^{n-2} \left| \frac{\arctan\left(\frac{y_{i+2}-y_{i+1}}{x_{i+1}-x_{i+1}}\right) - \arctan\left(\frac{y_{i+1}-y_{i+1}}{x_{i+1}-x_i}\right)}{\sqrt{(x_{i+1}-x_i)^2 + (y_{i+1}-y_i)^2}} \right| + w_2 \sum_{i=1}^{n-1} \sqrt{(x_{i+1}-x_i)^2 + (y_{i+1}-y_i)^2} \quad (1)$$

where (x_i, y_i) is the coordinates of waypoint i in the generated lane path; n is the number of waypoints of the generated lane path; w_1 and w_2 represent the weight coefficients for path smoothness and length, respectively.

III. SPEED PLANNING

For dynamic obstacles, the speed planning module calculates the intersections between the predicted trajectory of the obstacle vehicle and the planned path in Section II to generate the s - T graph. The essence of speed planning is the optimization of a non-convex problem. Since there is no global optimal solution for non-convex problems, it is necessary to first transform this non-convex problem into a convex one for resolution.

A. Convex Space Generation

To improve the response speed and rationality of path planning, we propose a novel convex space generation method. This method takes into account the vehicle's kinematic characteristics and the current traffic scenario. As shown in Figure 4, there are two convex spaces, namely, region A above the obstacle area and region B below the obstacle area. Region A implies that the ego vehicle merges into the lane before the rear obstacle vehicle arrives, while region B means that the ego vehicle yields to the rear obstacle vehicle, waits for it to pass, and then merges into the lane. However, due to constraints such as the ego vehicle's speed during lane change, the maximum allowed acceleration and deceleration in the vehicle's kinematics, the speed of the obstacle vehicle, and the distance from the obstacle vehicle, regions A and B often cannot both simultaneously meet these constraints. Therefore, the current traffic situation must be evaluated first.

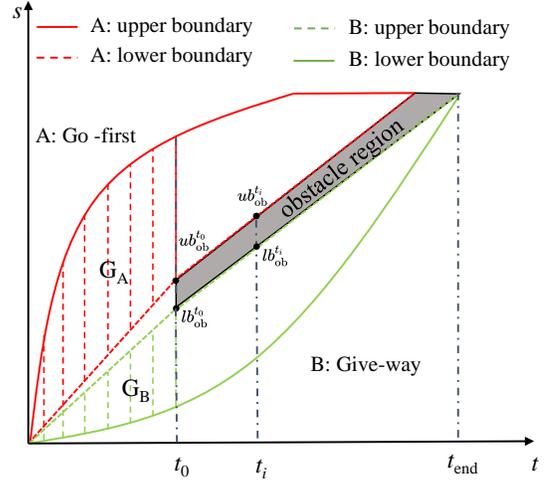


Fig. 4. Illustration of constraints on the s - T Graph.

Assuming the ego vehicle moves with uniform acceleration or deceleration and does not reverse if the ego vehicle can merge into the lane before the rear obstacle vehicle arrives, it must satisfy Equation 2.

$$v_0 t_0 + \frac{1}{2} \bar{a} t_0^2 > ub_{\text{ob}}^{t_0} \quad (2)$$

where v_0 is the initial speed of the ego vehicle; t_0 is the start time of the obstacle region; \bar{a} is the maximum acceleration of the ego vehicle; $ub_{\text{ob}}^{t_0}$ is the upper bound of the obstacle region at time t_0 .

The upper boundary $S_A^{\text{ub}}(t_i)$ of the convex space in region A can be expressed as Equation 3, if the Equation 2 is true.

$$S_A^{\text{ub}}(t_i) = \begin{cases} v_0 t_i + \frac{1}{2} \bar{a} t_i^2, & v_0 t_i + \frac{1}{2} \bar{a} t_i^2 < S_L \\ S_L, & v_0 t_i + \frac{1}{2} \bar{a} t_i^2 > S_L \end{cases} \quad (3)$$

where S_L is the length of the planned path.

The lower boundary $S_A^{\text{lb}}(t)$ of the convex space in region A can be expressed as Equation 4.

$$S_A^{\text{lb}}(t_i) = \begin{cases} v_0 t_i + \left(\frac{ub_{\text{ob}}^{t_0} - v_0 t_0}{t_0^2}\right) t_i^2 & t_i < t_0 \\ ub_{\text{ob}}^{t_0} & t_i > t_0 \end{cases} \quad (4)$$

where $ub_{\text{ob}}^{t_i}$ is the upper bound of obstacle region at time t_i .

If the ego vehicle can yield to the rear obstacle vehicle, it must satisfy Equation 5.

$$v_0 t_0 + \frac{1}{2} \underline{a} t_0^2 < lb_{\text{ob}}^{t_0} \quad (5)$$

where \underline{a} is the maximum deceleration of the ego vehicle.

The upper boundary $S_B^{\text{ub}}(t_i)$ of the convex space in region B can be expressed as Equation 6, if the Equation 5 is true.

$$S_B^{\text{ub}}(t_i) = \begin{cases} v_0 t_i + \left(\frac{lb_{\text{ob}}^{t_0} - v_0 t_0}{t_0^2}\right) t_i^2 & t_i < t_0 \\ lb_{\text{ob}}^{t_0} & t_i > t_0 \end{cases} \quad (6)$$

The lower boundary $S_B^{lb}(t_i)$ of the convex space in region B can be expressed as Equation 7.

$$S_B^{lb}(t_i) = \begin{cases} v_0 t_i + \frac{1}{2} \underline{a} t_i^2 & v_0 + \underline{a} t > 0 \\ -\frac{1}{2} \frac{v_0^2}{\underline{a}} & v_0 + \underline{a} t_i \leq 0 \end{cases} \quad (7)$$

If Equations 2 and 5 are both true, it is necessary to evaluate the quality of the convex spaces in regions A and B. In this paper, the area enclosed by the upper and lower boundaries of the convex spaces in regions A and B within the time interval $0 \sim t_0$ is used as a measure, as shown in Figure 4. This indicator reflects the system's ability to handle unexpected scenarios and the safety of speed planning solutions. A larger area implies a stronger ability to handle sudden situations and a safer speed planning solution.

$$\begin{cases} G_A = \int_0^{t_0} (S_A^{ub}(t) - S_A^{lb}(t)) dt \\ G_B = \int_0^{t_0} (S_B^{ub}(t) - S_B^{lb}(t)) dt \end{cases} \quad (8)$$

B. Speed Profile Formulation

This paper employs the piecewise-jerk method [14] to model the velocity profile. This approach relies on the Taylor series to establish the relationship between consecutive points (t_i, s_i) and (t_{i+1}, s_{i+1}) on the velocity profile, as shown in Equation 9. Furthermore, this method can effectively constrain every point on the velocity profile, providing greater flexibility than directly using curve models for modelling.

$$\begin{pmatrix} 1 & 0 \\ \Delta t & 1 \\ \frac{1}{3} \Delta t^2 & \frac{1}{2} \Delta t \\ -1 & 0 \\ 0 & -1 \\ \frac{1}{6} \Delta t^2 & \frac{1}{2} \Delta t \end{pmatrix}^T \begin{pmatrix} s(t_i) \\ s'(t_i) \\ s''(t_i) \\ s(t_{i+1}) \\ s'(t_{i+1}) \\ s''(t_{i+1}) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad (9)$$

Where $\Delta t = t_{i+1} - t_i$; $s'(t)$, $s''(t)$ and $s'''(t)$ are the first-order, second-order, and third-order of speed profile function $s(t)$ with respect to time parameter t .

C. Optimization Objective Function

We employ the QP algorithm to search for the optimal speed profile. The objective function takes into account the distance, speed, acceleration, and jerk of each speed point. The objective function is denoted as Equation 10.

$$\begin{aligned} \min J_{QP} &= w_3 \Sigma s^2(t_i) + w_4 \Sigma (s'(t_i))^2 \\ &\quad + w_5 \Sigma (s''(t_i))^2 + w_6 \Sigma (s'''(t_i))^2 \\ \text{s.t.} &\begin{cases} s(t_0) = 0 \\ s'(t_0) = v_0 \\ S^{lb}(t_i) \leq s(t_i) \leq S^{ub}(t_i) \\ \underline{v} \leq s'(t_i) \leq \bar{v} \\ \underline{a} \leq s''(t_i) \leq \bar{a} \\ \underline{jerk} \leq s'''(t_i) \leq \bar{jerk} \end{cases} \end{aligned} \quad (10)$$

Where \bar{v} and \underline{v} are the maximum and minimum speed, respectively; \bar{jerk} and \underline{jerk} are the maximum and minimum jerk values.

IV. EXPERIMENT RESULTS

In this section, we analyze the results of path planning and speed planning algorithms proposed in this paper. The simulations were performed in Matlab on an Intel Core i9 at 3.00 GHz.

A. Path Planning Results

As depicted in Figure 5, a cluster of candidate paths is generated. Utilizing Equation 1, the optimal path is derived from these candidates. This optimal path enables the ego vehicle to change lanes smoothly.

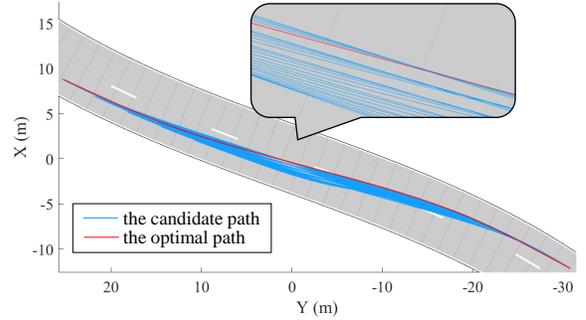


Fig. 5. The optimal path selection.

B. Speed Planning Results

In case 1, the initial speed of the ego vehicle is 9 m/s, since the initial speed is low, in the $s - T$ graph, the proposed speed planning algorithm chooses the region below the obstacle area as the convex space and successfully generates the speed profile. As illustrated in Figure 6 (b), (c), and (d), the ego vehicle first decelerates to yield the right-of-way to the approaching obstacle vehicle, then accelerates to change lanes after the obstacle vehicle passes, and finally maintains a constant speed. The velocity, acceleration, and jerk profiles are smooth and continuous making the lane change process stable and feasible.

In case 2, as illustrated as Figure 7 (a), the initial speed of the ego vehicle is 15 m/s, and since the initial speed is high, the proposed speed planning algorithm selects the region above the obstacle area as the convex space to search the optimal speed profile. In addition, as shown in Figure 7 (b), (c), and (d), to make the process of lane change safe and comfortable, the ego vehicle first decelerates slowly to change lanes, then accelerates to merge into the target lane. Throughout the process, the velocity, acceleration, and jerk profiles are smooth and continuous, ensuring feasibility during actual driving.

As shown in Table I, the conventional method (DP&QP) requires 107.4 ms to generate the optimal velocity profile. In contrast, our proposed method achieves the same task in just 72.3 ms. This represents a remarkable reduction in the average running time, with a decline rate of 32.7%.

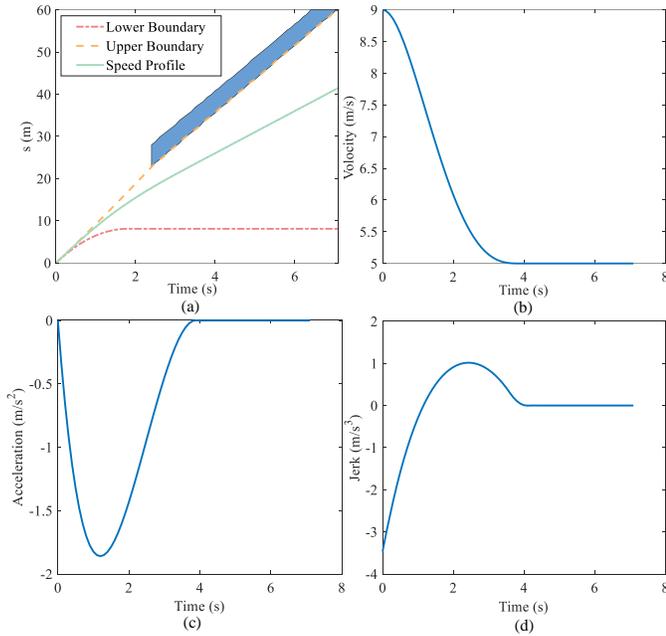


Fig. 6. The speed planning results in case 1: (a) the speed profile in $s - T$ graph; (b) the optimal velocity; (c) the optimal acceleration; (d) the optimal jerk.

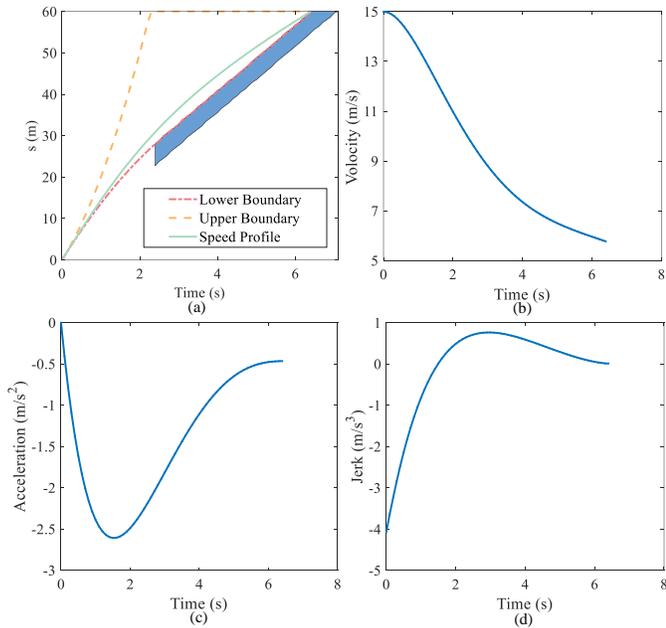


Fig. 7. The speed planning results in case 2: (a) the speed profile in $s - T$ graph; (b) the optimal velocity; (c) the optimal acceleration; (d) the optimal jerk.

V. CONCLUSION

In this paper, we propose a novel trajectory planning algorithm using the PVD framework. For path planning, we compared the performance of Dubins curves, Sine curves, Bezier curves, and B-spline curves for lane-changing scenarios. Ultimately, we selected the B-spline curve as the lane-

TABLE I
AVERAGE RUN-TIME COMPARISON.

Method	Time (ms)	Decline Rate
DP&QP	107.4	/
Ours	72.3	32.7%

changing path model, ensuring that the planned path has zero curvature at both the starting and ending points, thus preventing the ego vehicle from deviating from the lane center after changing lanes.

For speed planning, we propose a new method to generate the convex space faster. This method can adaptively create a convex space based on the current kinematic characteristics of the ego vehicle and the traffic scenario. Compared with the conventional method (DP&QP), the algorithm achieves a shorter average run time, with a decrease of 32.7 %.

In future work, we intend to employ and further develop the proposed idea of convex space generation in lane-changing scenarios involving multiple dynamic obstacles, thereby expanding the applicability of this approach.

REFERENCES

- [1] L. Claussmann, M. Revilloud, D. Gruyer, and S. Glaser, "A review of motion planning for highway autonomous driving," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 5, pp. 1826-1848, May 2020.
- [2] O. Sharma, N.C. Sahoo, and N.B. Puhan, "Recent advances in motion and behavior planning techniques for software architecture of autonomous vehicles: a state-of-the-art survey," *Eng. Appl. Artif. Intell.*, vol. 101, pp. 104211, May 2021.
- [3] S. Teng, X. Hu, P. Deng, B. Li, Y. Li, and Y. Ai et al., "Search-based task and motion planning for hybrid systems: agile autonomous vehicles," *Eng. Appl. Artif. Intell.*, vol. 121, pp. 105893, May 2023.
- [4] W. Lim, S. Lee, M. Sunwoo, and K. Jo, "Hybrid trajectory planning for autonomous driving in on-road dynamic scenarios," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 1, pp. 341-355, January 2021.
- [5] B. Paden, M. Čáp, S. Z. Yong, D. Yershov and E. Frazzoli, "A Survey of Motion Planning and Control Techniques for Self-Driving Urban Vehicles," *IEEE Trans. Intell. Veh.*, vol. 1, no. 1, pp. 33-55, March 2016.
- [6] S. Karaman, and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," *Int. J. Rob. Res.*, vol. 30, no. 7, pp. 846-894, 2011.
- [7] L. Ma, J. Xue, K. Kawabata, J. Zhu, C. Ma and N. Zheng, "Efficient sampling-based motion planning for on-road autonomous driving," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 4, pp. 1961-1976, Aug. 2015.
- [8] S. M. LaValle, and J. J. Kuffner, "Randomized Kinodynamic Planning," *Int. J. Robot. Res.*, vol. 20, no. 5, pp. 378-400, May 2001.
- [9] S. Aine, S. Swaminathan, V. Narayanan, V. Hwang, and M. Likhachev, "Multi-Heuristic A*," *Int. J. Rob. Res.*, vol. 35, no. 1-3, pp. 224-243, August 2016.
- [10] P. E. Hart, N. J. Nilsson and B. Raphael, "A Formal Basis for the Heuristic Determination of Minimum Cost Paths," *IEEE Trans. Syst. Man Cybern.*, vol. 4, no. 2, pp. 100-107, July 1968.
- [11] J. Huang, Z. He, Y. Arakawa, and B. Dawton, "Trajectory Planning in Frenet Frame via Multi-Objective Optimization," *IEEE Access*, vol. 11, pp. 70764-70777, July, 2023.
- [12] W. Xu, and J. M. Dolan, "Speed planning in dynamic environments over a fixed path for autonomous vehicles," in *2022 IEEE Int. Conf. Robot. Autom. (ICRA)*, Philadelphia, PA, USA, 2022, pp. 3321-3327.
- [13] K. Kant, and S. W. Zucker, "Toward efficient trajectory planning: The path-velocity decomposition," *Int. J. Rob. Res.*, vol. 5, no. 3, pp. 72-89, 1986.
- [14] J. Zhou, R. He, S. Jiang, Z. Zhu, J. Hu, and J. Miao et al., "Autonomous driving trajectory optimization with dual-loop iterative anchoring path smoothing and piecewise-Jerk speed optimization," *IEEE Robot. Autom. Lett.*, vol. 6, no. 2, pp. 439-446, April 2021.