2019 IEEE Intelligent Transportation Systems Conference (ITSC)
Auckland, NZ, October 27-30, 2019

Hybrid A*-based Curvature Continuous Path Planning in Complex Dynamic Environments

Songyi Zhang[1] and Yu Chen[1] and Shitao Chen[1*] and Nanning Zheng[1]

Abstract— **With the progress of autonomous driving technology in recent years, motion planning has been an issue in the navigation of self-driving cars. To achieve an optimal path that meets the requirements of both smoothness and safety, vehicle kinematics and dynamics constraints should be considered. This paper proposes a novel motion planning method based on Hybrid A* for real-time and curvature-contentious path planning with local post smoothing in complex dynamic environments: (1)our method introduces parametric clothoid curves precomputed offline as basic motion primitives for rapid online planning; (2)the path obtained using our method is G^2-continuous (i.e., curvature continuous) and does not have a considerable effect on the search time consumption, while also considering possible collisions and motion constraints of nonholonomic car-like vehicles; (3) the node re-expansion issue of conventional Hybrid A* is discussed and resolved by the proposed quintic spine-based local smoothing approach for complete path continuity. Hence, post smoothing and collision checking for the overall resulting path. Simulation and on-road tests have been performed to evaluate the efficiency of the proposed method. The method can be widely implemented in numerous complex scenarios.**

I. INTRODUCTION

The main challenge in motion planning for autonomous vehicles[1] is finding the shortest feasible path in a limited time with no collision, especially in a dynamic complex environment with uncertainty. Moreover, to optimize the control performance of the vehicle for minimum jitters and to ensure passenger comfort during driving, the generated path is supposed to be sufficiently smooth with at least G^2-continuity.

At present, motion planning approaches that are widely implemented on real autonomous driving platforms can be categorized into three main types. The first is interpolating curve planner that can obtain continuous planning results, such as polynomials [2], [3], splines [4], and clothoids [5]. The algorithms of these planners generally have low computational complexity, except for clothoids, which have large computational cost to calculate integrals. Interpolation-based planners are mainly applicable to structured environments such as highways or urban roads, because curve fitting relies greatly on the waypoints of the pre-constructed road networks. In addition, the planning complexity increases as the environment evolves with higher uncertainty when multiple dynamic obstacles appear.

The second type is graph-based planner, which discretizes the vehicle state space into a graph and utilizes different heuristic algorithms to achieve sub-optimal or even optimal solutions. Classic graph-based planning algorithms include state lattices [6], A* and its various variants such Hybrid A* [7] and ARA* [8]. Nevertheless, grid-based searching fails to consider vehicle kinematics constraints or lacks searching freedom in certain dimensions. For example, the A* planner allows limited expanding directions and the result is composed of a series of straight lines, implying that further path smoothing is inevitable for better path feasibility. The third type is sampling-based planner, including Probabilistic Roadmap (PRM) [9], Rapidly-exploring Random Tree (RRT) [10] and its variants. Through random sampling in the vehicle state space, a connected graph can be constructed to possibly realize the connection between the initial state and the target state. However, the safety and continuity of the planning result can not be guaranteed, requiring further collision checking and path smoothing.

Within this context, this paper proposes a novel planning method based on Hybrid A* which is combined with clothoids for G^2-continuity, to better combine the advantages of different planning approaches to achieve more effective and comprehensive solutions. First, in contrast to classic graph-based algorithms, Hybrid A* considers the motion constraints of an autonomous vehicle and searches in a vehicle state space of four dimensions $\{x, y, \theta, d\}$ ($d \in \{+1, -1\}$ is the normalized speed, which represents the driving mode: drive/reverse). The planned path of Hybrid A* has G^1-continuity(i.e., orientation continuity). Further numerical optimization and collision checking are required to avoid unstable course change due to frequent jitters. However, by applying offline calculated motion primitives which are a series of parametric clothoid segments for exploring in a five-dimensional state space $\{x, y, \theta, \kappa, d\}$, where κ is the curvature, we can ensure the curvature continuity and avoid the computational cost of online fitting when using clothoids. Further, the planned path can be optimized to G^3-continuity with a continuous curvature rate when certain restrictions are imposed on the selection of κ during planning.

II. RELATED WORK

Graph-based searching has been studied extensively, but how to improve and optimize the efficiency of searching as well as the feasibility of search results remains a challenge

*This work was supported by the National Natural Science Foundation of China(NO.61773312,61790563).

[1]S. Zhang, Y. Chen, S. Chen, N. Zheng are with the Department of Electrical and Information Engineering, Xi'an Jiaotong University, Xi'an, Shaanxi 710049, P.R. China; Email: zhangsongyi, alan1996, chenshitao@stu.xjtu.edu.cn; nnzheng@mail.xjtu.edu.cn

*Correspondence: chenshitao@stu.xjtu.edu.cn

for further research. In general, grid-based planners are capable of modelling the surrounding environments to provide a relatively accurate representation of the environmental information, while the discrete search space can lead to a discontinuous path or path curvature.

Based on the conventional A* algorithm, the Hybrid A* planner [7], [11] is an extended version proposed and implemented on the autonomous driving platform "Junior" from Stanford, which won the second place in the 2007 DARPA Urban Challenge. By combining the traditional A* planner with Reed-Shepp curves [12], Hybrid A* introduced arcs with minimum turning radius of vehicle as motion primitives in the process of node expansion. Besides, the vehicle orientation and driving mode were taken into account as the third and fourth dimension of searching space respectively, so as to obtain reasonable solutions in accordance with the vehicle kinematics constraints. Then the Conjugate Gradient method was used for path optimization and smoothing due to the curvature discontinuity caused by incomplete and limited steering motions. Later, many researchers developed their work based on the Hybrid A* planner. Petereit et al. [13] implemented Hybrid A* on a non-holonomic outdoor robot with a real-time performance in unstructured outdoor environments that were partially or completely unknown, whereas the continuity of the start and target positions were not guaranteed. [14] defined the heuristics using A* search results and Reeds-Shepp curves, and they also increased the speed of the searching process by applying Reeds-Shepp curves in node expansion. Nevertheless, unreasonable situations such as frequent reverse driving and direction change may appear in the generated path. In [15], the collision evaluation as well as the heuristics of Hybrid A* were modified for a more rapid state expansion and accurate navigation.

However, few studies have focused on the combination of Hybrid A* with curves of higher flexibility, such as clothoids, for smoother and more real-time path planning. The clothoid curve, also called Euler spiral, is an efficient smoothing method based on Fresnel integrals and is mostly used to mitigate the curvature changes between curves and straight lines. In [16], a sequence of clothoids, arcs and straights were concatenated to generate an optimal path according to the original track, thus promoting a smooth steering control performance of their Audi TTS in real time. In the VisLab Intercontinental Autonomous Challenge (VIAC) 2010, the trajectory planner adopted by Broggi et al. [17] computed all feasible trajectories with clothoids and chose the optimal one on considering the vehicle limits, comfort constraints, and the moving cost. Hence, smooth curvature variations with respect to time were ensured to avoid the occurrence of possible vehicle jitters; however, the online calculation of curve integrals resulted in a longer computation time. [6], [18], [19] implemented clothoid curves as pre-computed motion primitives for local path planning in diverse scenarios to obtain smooth searching solutions. Based on the above, introducing offline computed clothoid curves as motion primitives in Hybrid A* for real-time path planning with G^2-continuity is of great significance and potential value. It is a promising method that simultaneously ensures rapidity, safety, and applicability of planning.

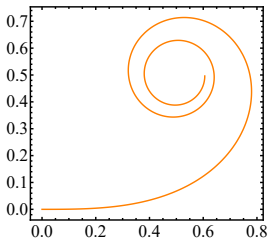


Fig. 1. The Standard Euler Spiral, with the parameter l in the range of $[0, 3]$.

III. VEHICLE STATE MODEL

Following the study of [20], we choose a curvature-based model motion to describe the kinematic movement of a car-like robot:

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\kappa} \end{bmatrix} = \begin{bmatrix} \cos\theta \\ \sin\theta \\ \kappa \\ 0 \end{bmatrix} d + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \sigma \tag{1}$$

where (x, y) is the current position of the vehicle, θ is the vehicle orientation, d denotes the two different gear status of vehicle. $d = +1$ indicates driving forward, and $d = -1$ indicates driving backward. κ is the curvature of the trajectory with a derivative σ that signifies curvature rate.

Assuming that the vehicle state at mileage l_0 is $\mathbf{x}_0 = \begin{bmatrix} x_0 & y_0 & \theta_0 & \kappa_0 \end{bmatrix}^\mathsf{T}$, and the vehicle state after moving to mileage $l_0 + L$ is $\mathbf{x}_L = \begin{bmatrix} x_0 + \Delta x & y_0 + \Delta y & \theta_0 + \Delta\theta & \kappa_0 + \Delta\kappa \end{bmatrix}^\mathsf{T}$, the increment of the state variables can be obtained by:

$$\Delta\kappa(L) = \int_{l_0}^{l_0+L} \sigma(l) dl \tag{2a}$$

$$\Delta\theta(L) = \int_{l_0}^{l_0+L} d(l)\kappa(l)\, dl \tag{2b}$$

$$\Delta x(L) = \int_{l_0}^{l_0+L} d(l)\cos\theta(l)\, dl \tag{2c}$$

$$\Delta y(L) = \int_{l_0}^{l_0+L} d(l)\sin\theta(l)\, dl \tag{2d}$$

To obtain the closed-form expression of (2), $\sigma(l)$ and $d(l)$ are assumed to be constant values σ and d during the process. In this hypothesis, the results of (2) can be directly calculated as

$$\Delta\kappa = \sigma L \tag{3a}$$

$$\Delta\theta = d\kappa_0 L + \frac{1}{2} d\sigma L^2 \tag{3b}$$

$$\Delta x = \int_{l_0}^{l_0+L} d\cos\left(\theta_0 + d\kappa_0 l + \frac{1}{2} d\sigma l^2\right) dl \tag{3c}$$

$$\Delta y = \int_{l_0}^{l_0+L} d\sin\left(\theta_0 + d\kappa_0 l + \frac{1}{2} d\sigma l^2\right) dl \tag{3d}$$

When $\sigma = 0$, the closed-form expression of (3c) and (3d) are trivial:

$$\Delta x = \begin{cases} \frac{1}{\kappa_0}\left[\sin(\theta_0 + dL\kappa_0) - \sin(\theta_0)\right] & , \kappa_0 \neq 0 \\ dL\cos\theta_0 & , \kappa_0 = 0 \end{cases} \tag{4}$$

$$\Delta y = \begin{cases} -\frac{1}{\kappa_0}\left[\cos(\theta_0 + dL\kappa_0) - \cos(\theta_0)\right] & , \kappa_0 \neq 0 \\ dL\sin\theta_0 & , \kappa_0 = 0 \end{cases} \tag{5}$$

Depending on whether the initial curvature $\kappa_0 \neq 0$, the resulting trajectory will be a circular arc of κ_0 or a line.

Otherwise, when $\sigma \neq 0$, the closed-form expression can be calculated using the Fresnel functions, and the resulting trajectory will be a segment from the Euler Spiral, which is shown in Fig. 1.

IV. DISCRETIZATION OF SEARCH SPACE

Similar to the Hybrid A* scheme, we discretized the searching space to map the continuous space to a discrete one in order to satisfy the requirements of performing a search in a discrete configuration space. A node N_k is defined with a quadruple as $N_k = \begin{bmatrix}\text{idx}_x & \text{idx}_y & \text{idx}_\theta & \text{idx}_\kappa\end{bmatrix}^\mathsf{T}$, and its relationship with the continuous state $\mathbf{x}_i$ is

$$\mathbf{x}_k \in N_i \Leftrightarrow \begin{bmatrix}\text{idx}_x \\ \text{idx}_y \\ \text{idx}_\theta \\ \text{idx}_\kappa\end{bmatrix} = \begin{bmatrix}\lfloor x/r \rfloor \\ \lfloor y/r \rfloor \\ \lfloor \theta/\theta_{\text{Step}} \rfloor \\ \lfloor \kappa/\kappa_{\text{Step}} \rfloor\end{bmatrix} \tag{6}$$

where $\theta_{\text{Step}} = (2\pi)/M_\theta$ is the angle step, and $\kappa_{\text{Step}} = \kappa_{\text{Max}}/M_\kappa$ is the curvature step. Considering the vehicle kinematics constraints, the curvature is restricted within the interval $[-\kappa_{\text{M}}, \kappa_{\text{M}}]$, equivalently, $\text{idx}_\kappa \in [-M_\kappa, M_\kappa]$. Then we can get a map described by $(\lceil W/r \rceil \times \lceil H/r \rceil \times M_\theta \times (2 * M_\kappa + 1))$, where W and H are the width and height of the map, respectively.

Subsequently, for node expansion during the search, the extended states are computed based on the continuous state $\mathbf{x}_0$ and (2) of the vehicle kinematic model; the information of their corresponding nodes are updated simultaneously.

Following the study of [6], to reduce the online calculation cost during expansion, especially that of Fresnel integrals, the curvature and orientation of $\mathbf{x}_0$ are pre-discretized. First, by limiting the curvature rate to $\sigma \in \{\pm\kappa_{\text{Step}}/L, 0\}$, according to (3a), the curvature rate must satisfy $\Delta\kappa \in \{\pm\kappa_{\text{Step}}, 0\}$. Then, the curvature κ generated from the initial state is discretized by κ_{Step}, indicating that the curvature of each node is solely determined by $\kappa = \text{idx}_\kappa * \kappa_{\text{Step}}$.

Further, by defining the expansion step for a node as $L = 2\theta_{\text{Step}}/\kappa_{\text{Step}}$, with (3b) we have:

$$\begin{aligned}\Delta\theta &= d\kappa_0 L + \frac{1}{2}d\sigma L^2 \\ &= d * \text{idx}_\kappa * \kappa_{\text{Step}} * \frac{2\theta_{\text{Step}}}{\kappa_{\text{Step}}} + \frac{1}{2}d * \{\pm\kappa_{\text{Step}}, 0\} * \frac{2\theta_{\text{Step}}}{\kappa_{\text{Step}}} \\ &= \pm 1 * (2 * \text{idx}_\kappa + \{\pm 1, 0\}) * \theta_{\text{Step}}\end{aligned} \tag{7}$$

Then, the orientation of each node is uniquely determined by the orientation of the starting point as well as the orientation index idx_θ.

Given that (2) is only related to the curvature and curvature rate of the initial state, the Look Up Table(LUT) method can be used for the offline calculation of all the discrete state increments before the online searching for reduced expansion time and cost.

Fig. 2(a) and Fig. 2(b) present the basic motion primitives and their combinations adopted in our method. Fig. 2(a) shows how a clothoid curve ($\kappa_0 = 0, \sigma = \sigma_{\text{Step}}$) smoothly links a straight segment ($\kappa_0 = 0, \sigma = 0$) with an arc segment ($\kappa_0 = \kappa_{\text{Step}}, \sigma = 0$). Clothoids can also form continuous transitions between arcs of different curvatures, as shown in Fig. 2(b). The motion primitives employed by the traditional Hybrid A* are listed in Fig. 2(c). Since the Hybrid A* searching is based on Reeds-Shepp curves, admissible directional motions are restricted to a large extent by the simple combination of straight lines and circular arcs. It can be seen that there are obvious curvature discontinuities at the transitions between lines and arcs in the generated path, making severe jitters and frequent steering inevitable if the planning result is used directly for path tracking. In contrast, our proposed method is effective in improving the overall smoothness of the planned paths by extending the searching space with a curvature dimension and introducing clothoids as motion primitives.

V. EXPANSION WITH REVISIT AND LOCAL SMOOTHING

Unlike the traditional A*, allowing the re-expansion of a node for possible cheaper path solutions with inconsistent heuristics is comparatively a better choice to get an optimal path.

The consistency of a heuristic h signifies that, for any two nodes N_i and N_j, h satisfies the following conditions:

$$\begin{cases} h(N_i) \leq c(N_i, N_j) + h(N_j) \\ h(N_G) = 0 \end{cases} \tag{8}$$

where $c(N_i, N_j)$ is the motion cost of moving from N_i to N_j, and N_G is the goal node. When the consistency conditions (8) are satisfied and a node N_k is already expanded, it has been proved [21] that

$$g(N_k) = g^*(N_k) \tag{9}$$

where $g(N_k)$ is the cost from the start to N_k, $g^*(N_k)$ is the minimum cost from the start to N_k. In this case, any other node that can expand to N_k has no chance to get a shorter path, that is

$$g(N_i) + c(N_i, N_k) \geq g^*(N_k) = g(N_k) \tag{10}$$

Therefore, revisiting N_k is of no significance after it is expanded.

However, designing consistent heuristics is a challenging task for graph-based searching. When executing A* search in a free space, the consistency requirement can be satisfied by applying the Euclidean distance from the current

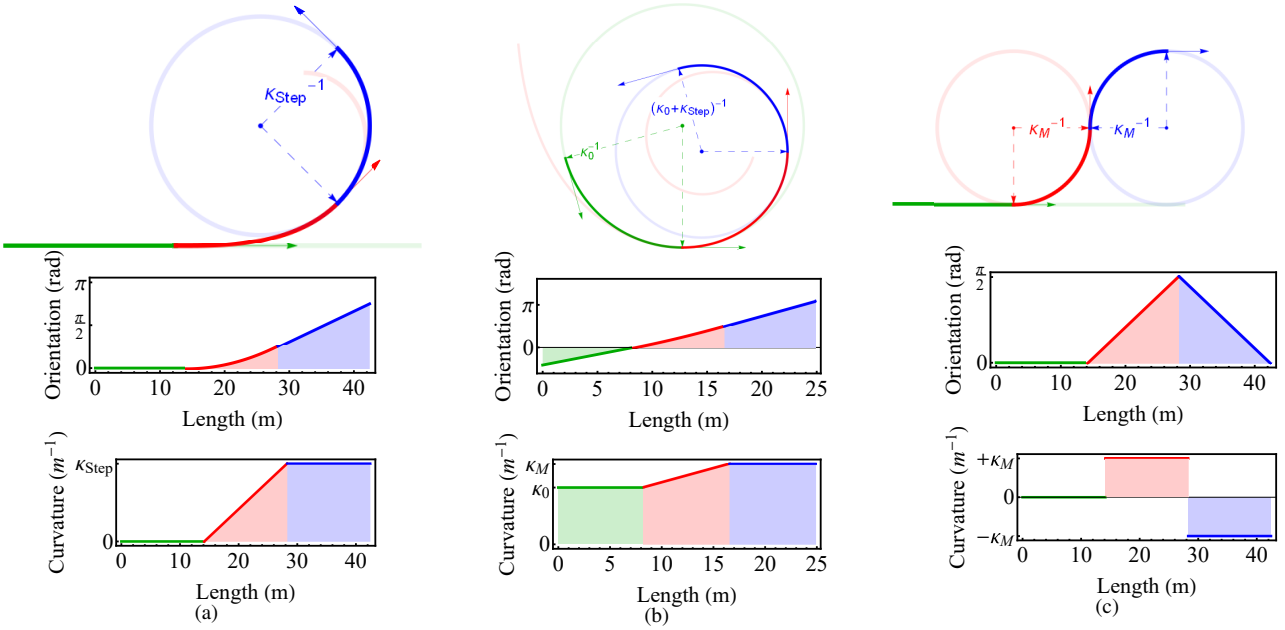


Fig. 2. Comparison of motion primitive sets. (a) A straight(green) and an arc(blue) are connected using a clothoid segment(red) in our method. (b) Arcs of different curvatures(green, blue) are connected using a clothoid segment(red) in our method. (c) A straight(green) and an arc(blue) are connected directly with another arc(red) in Hybrid A*, causing curvature discontinuity.

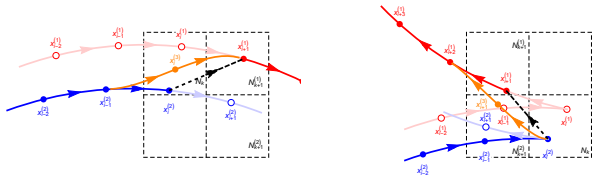


Fig. 3. Discontinuity caused by revisiting and repair: (a) Without changing the driving mode. (b) Change in the driving mode.

configuration to the goal configuration as an admissible heuristic, which turns to be unavailable in the presence of any unexpected obstacle on path. This is a more tough issue for the Hybrid A* algorithm with a searching space of even higher dimensions and more complicated configurations. For example, a series of maneuvers such as reversing and turning may be required for transferring between two nodes that are actually very close with respect to their 2D Euclidean distance, since the definition of "distance" is complicated for high-dimensional spaces.

Hence, the revisiting of nodes is practical to improved the optimality of the resulting paths with respect to a reduced tracking cost. However, according to empirical research results it should also be noted that, this operation can possibly rewrite node information and thus lead to discontinuities between the final path points. A searched node always locates in the center of a grid in the conventional A* search and is therefore in the same position during the entire planning cycle, while Hybrid A* enables a continuous position change in a discredited graph for better path smoothness. Then the problem arises that the planar position parameters of a discrete state may be modified by other states that belong to the same node, with the discrete parameters θ and κ remaining unaffected. To address the aforementioned problem, this paper has brought out a slight post smoothing scheme based on quintic spline to repair particular discontinuities.

Fig. 3 illustrates two common cases of inconsistency that appear frequently in the searching process, which depend on whether there is a driving direction change. To explain the whole process more clearly, specific notions are defined: $N_k^{(*)}$ is a discretized node, and $\mathbf{x}_i^{(*)}$ is a continuous vehicle state. The relationship between nodes and states is indicated using subscripts:

$$\text{Pred}(N_{k+1}^{(*)}) = N_k^{(*)} \tag{11}$$

$$\mathbf{x}_{i+1}^{(*)} \in \text{Succ}(\mathbf{x}_i^{(*)}) \tag{12}$$

The range of the nodes in the search graph (dashed box), a searching branch extended previously (red curve), and the current branch being extended (blue curve) are illustrated in Fig. 3. Node N_k will be revisited because the state $\mathbf{x}_i^{(2)}$ falls within N_k after discretization. After it is revisited, the continuous position information on $\mathbf{x}_i^{(1)}$, which is stored in N_k will be replace by that of $\mathbf{x}_i^{(2)}$. Subsequently, N_k will be re-expanded, leading to two possible results. In one case, if node $N_{k+1}^{(2)}$ containing the state $\mathbf{x}_{i+1}^{(2)}$ is different from node $N_{k+1}^{(1)}$ containing the state $\mathbf{x}_{i+1}^{(1)}$, $N_{k+1}^{(2)}$ will be the curvature-continuous successor of N_k, and the precursor pointer of node $N_{k+1}^{(1)}$ to N_k will remain unchanged. If $N_{k+1}^{(1)}$ is on the resulting path, an inconsistent segment containing states $\{\mathbf{x}_{i-1}^{(2)}, \mathbf{x}_i^{(2)}$, and $\mathbf{x}_{i+1}^{(1)}\}$ will appear owing to the back tracing process of the precursor pointers for constructing the final path. The corresponding result is shown by the black dashed line. Otherwise, although the state $\mathbf{x}_i^{(2)}$ in node N_k and

the state $\mathbf{x}_{i+1}^{(2)}$ in node N_{k+1} are curvature-continuous, the revisiting problem is still inevitable because it is transferred from node N_k to node N_{k+1}, and the discontinuity may still appear in the output path.

To generate a curvature-continuous and collision-free path from the initial one, classic A* or Hybrid A* perform post smoothing and a collision check in a later stage by optimizing the global cost functions that consider the curvature constraints. In contrast, our method obtains an initial planning result with several discontinuous points, which only requires a convenient partial smoothing process without repeated calculation or collision checking over the entire path.

In Fig. 3(a), in terms of the discontinuous points of the same driving mode, we consider $\mathbf{x}_{i-1}^{(2)}$ and $\mathbf{x}_{i+1}^{(1)}$ as the endpoints, and a quintic spline function is applied to generate interpolated fragments to replace the original piecewise discontinuous curve. The G^2-continuity of the curve itself, and that of the curve endpoints are satisfied simultaneously. The quintic spline $\zeta = \begin{bmatrix} x & y \end{bmatrix}^\mathsf{T}$ is described by the parametric equation

$$\begin{cases} x(r) = \sum_{i=0}^{5} a_i r^i \\ y(r) = \sum_{i=0}^{5} b_i r^i \end{cases} \quad r \in [0,1] \tag{13}$$

The coefficients in (13) can be solved using the method in [22] with the boundary conditions calculated from the states

$$\begin{cases} \zeta = \begin{bmatrix} x & y \end{bmatrix}^\mathsf{T} \\ \zeta' = d \begin{bmatrix} \cos\theta & \sin\theta \end{bmatrix}^\mathsf{T} \\ \zeta'' = \kappa \begin{bmatrix} -\sin\theta & \cos\theta \end{bmatrix}^\mathsf{T} \end{cases} \tag{14}$$

Then, the interpolation segment can be fully obtained, shown by the orange curve in Fig. 3(a).

We use an intermediate state $\mathbf{x}_i^{(3)} = \zeta(1/2)$(orange point), which is the middle point of the interpolation curve, to replace the discontinuous state $\mathbf{x}_i^{(2)}$.

Similarly, as shown in Fig. 3(b), for a discontinuous node N_{k+1} with a change in the driving direction, quintic spline interpolation is conducted between the states $\mathbf{x}_i^{(2)}$ and $\mathbf{x}_{i+2}^{(1)}$. Then, the wrong state $\mathbf{x}_{i+2}^{(1)}$ is replaced by a newly generated one $\mathbf{x}_{i+1}^{(3)}$ to generate a continuous interpolation result(orange curve).

The interpolation curves and their endpoints are proved to be strictly G^2-continuous under both conditions illustrated in Fig. 3(a) and Fig. 3(b), thus guaranteeing the curvature continuity of the overall resulting path.

VI. COST FUNCTION FOR RESTRICTIONS ON NODE EXPANSION

Human drivers seldom perform unnecessary steering or reversing frequently when driving, while such maneuvers are often present in paths obtained by Hybrid A*. To minimize the total length of the path, multiple steering or gear switching operations are not considered. Therefore, we improve the cost function of the node expansion to achieve more human-like path planning results.

With a basic step L of different motion primitives, specific the coefficients considering essential factors for node expansion are defined: $\mathbf{W}_{\text{Curve}}$ is the curve penalty coefficient that prefers straight driving instead of turning. The coefficient $\mathbf{W}_\sigma$ assigns high costs for curvature change, the coefficient $\mathbf{W}_{\text{Reverse}}$ for reversing penalty, and the coefficient $\mathbf{W}_{\text{Keep}}$ for a short segment length between two gear switching motions. When expanding N_k to N_{k+1}, the moving cost is calculated as

$$\begin{aligned} c(N_k, N_{k+1}) =& (1 + \mathbf{W}_{\text{Curve}}) * (1 + \mathbf{W}_\sigma) * (1 + \mathbf{W}_{\text{Reverse}}) * \\ & (1 + \mathbf{W}_{\text{Keep}}) * L \end{aligned} \tag{15}$$

The coefficients are described as

$$\mathbf{W}_{\text{Curve}} = \begin{cases} w_{\text{Curve}} & , \kappa \neq 0 \wedge \sigma \neq 0 \\ 0 & , else \end{cases} \tag{16}$$

$$\mathbf{W}_\sigma = \begin{cases} w_\sigma & , \sigma \neq 0 \\ 0 & , else \end{cases} \tag{17}$$

$$\mathbf{W}_{\text{Reverse}} = \begin{cases} w_{\text{Reverse}} & , d = -1 \\ 0 & , else \end{cases} \tag{18}$$

where w_{Curve}, w_σ, and w_{Reverse} are the weights defined by the experimental tests that are related to the corresponding coefficients. The restrictions on node expansion impose a searching process that prefers the vehicle to reach to the target point with minimum steering and reversing.

Furthermore, frequent changes in the driving mode in the resulting path by Hybrid A* are usually redundant and unnecessary compared with human driving behavior, while switching gears in a short distance will also make it difficult for the autonomous vehicle to perform tracking. Therefore, an additional propriety that requires the vehicle to take a reasonable number of steps before switching the driving mode is defined as "keep". When node N_{k+1} is the successor of node N_k, the "keep" value of N_{k+1} is defined as

$$\text{keep}(N_{k+1}) = \begin{cases} \text{keep}(N_k) + 1 & , d(N_{k+1}) = d(N_k) \\ 0 & , d(N_{k+1}) \neq d(N_k) \end{cases} \tag{19}$$

Thus, we have

$$\mathbf{W}_{\text{Keep}} = \begin{cases} (M_{\text{Keep}} - \text{keep}) w_{\text{Keep}} & , \text{keep} < M_{\text{Keep}} \\ 0 & , else \end{cases} \tag{20}$$

where M_{Keep} is the minimum steps a vehicle should follow when driving either forward or backward, and w_{Keep} is the weight determined experimentally. The introduction of a "keep" penalty minimizes the possibility of frequent gear switches that are unnecessary and may possibly cause unfeasible path segments.

VII. EXPERIMENTS

A. Comparison with Classic Hybrid A*

Through clothoid interpolation and local quintic spline smoothing, our method achieves higher curvature smoothness

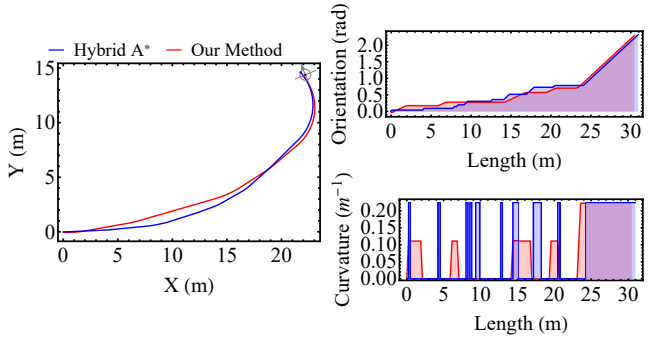


Fig. 4. Comparison of curvature with traditional Hybrid A*, in a free space.

of planning results compared with Hybrid A*. To validate this superiority, we perform experiments in a virtual free space scenario to compare the two abovementioned methods. The basic parameters of the experimental environment for both planners should be exactly the same, including map size, search resolution, start position, target position, etc. Then, given the same target point, each planner provides a reasonable solution with perceptual information gathered from sensors. We repeated the experiments multiple times to ensure the universality and reliability of the experimental results. Fig. 4 shows the analysis and comparison of the resulting paths, path angle changes, and path curvature changes of two planners. Frequent shifts of curvature are inevitable in the results of Hybrid A* searching, which is of G^1-continuity, while the proposed method achieves G^2-continuity without frequent irregular curvature change.

B. ROBUSTNESS AND REAL-TIME PERFORMANCE IN COMPLEX SCENARIOS

Under the combined effect of search space enlargement and a targeted expansion strategy, our algorithm can deal with obstacles in complex dynamic scenarios more flexibly and safely in a short period to meet real-time planning and re-planning requirements. Several typical complex traffic scenarios were simulated on our self-constructed simulation platform with hardware in the loop [23], [24] to assess the adaptability of our algorithm to complex environments.

The first scenario is a narrow road with a dead end. The vehicle is supposed to turn around to the opposite direction in a limited space through maneuvers such as a three-point turn (i.e., a K-turn), which demands multiple gear switching. The second scenario is a construction section with cones for guidance. This tests whether the vehicle can circumvent the construction area and respond rapidly to possible dynamic environmental changes. Finally, the last scenario is an open parking lot. The vehicle should complete a smooth and precise autonomous parking operation to reach the designated parking spot. Table I and Fig. 5 compare the planning results of two planners in different scenarios.

Several conclusions can be drawn from the comparison results. First, in the construction area scenario with guidance cones, the path obtained by the proposed method is more in accordance with the direction of the guidance curve formed by cones and is closer to the operation of human drivers. Second, because of an enlarged searching space in dimension and reduced searching motion constraints, our method tends to visit more nodes compared with Hybrid A* for the same planning situation. This phenomenon is more apparent in highly restricted scenes, such as K-turn or parking. In these situations, the nodes visited by our planner are generally two or three times more than those visited by Hybrid A*. Third, because the improved cost functions and the discretization of the curvature contribute to a lower computational cost for node expansion, the time cost of our method is approximately the same as that of Hybrid A*, even if three times greater number of nodes are visited. For most unstructured scenarios, our method can complete the entire planning process in a shorter time.

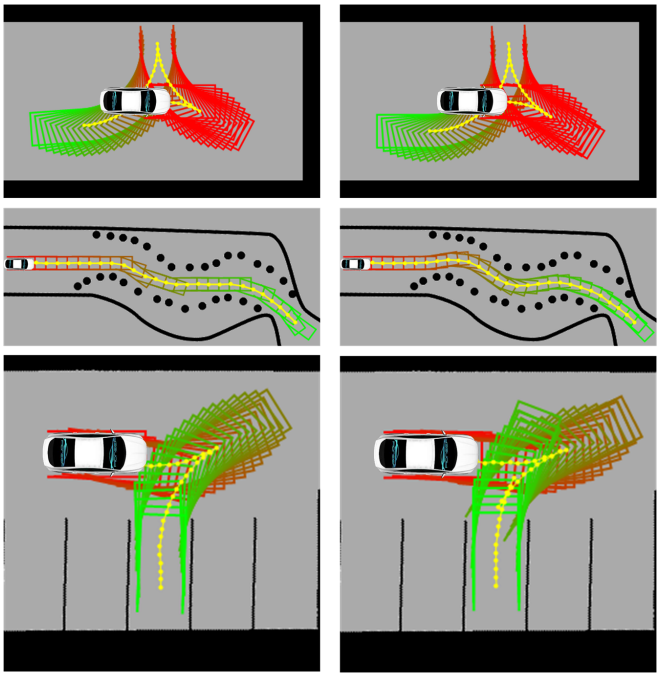


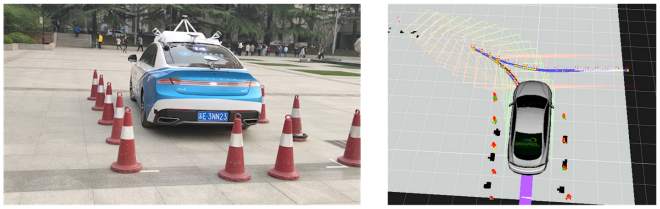
Fig. 5. Path planning results in three complex scenarios: narrow road with dead end, construction area, and parking lot. (a) Planning results of Hybrid A*. (b) Planning results of the proposed method.

TABLE I

COMPARISON OF THE PROPOSED METHOD WITH TRADITIONAL HYBRID A* IN DIFFERENT TYPICAL SCENARIOS

Scenario	Method	Searched Nodes	Time Cost (ms)	Path Length (m)	Kappa Discontinuity
Freespace	Hybrid A*	331335	4309	31.05	19
	Ours	705860	3085	30.63	0
K-Turn	Hybrid A*	90158	998	14.14	3†
	Ours	168967	789	16.02	2†
Construction area	Hybrid A*	71	1.18	41.47	11
	Ours	236	1.03	42.10	0
Parking lot	Hybrid A*	19011	500	11.62	5†
	Ours	56807	476	15.08	3†

†: Including points with gear switching.



(a) (b)

Fig. 6. Validation result of parking scenario, on our "Pioneer" autonomous driving platform.

C. Validation on Real Autonomous Driving Platform

Apart from the validation through simulation, we also immigrated our planning algorithm to a real autonomous driving platform, "Pioneer", for further on-road testing. A designated parking spot surrounded with cones is located on a square. Unlike the simulation, information of all the static and dynamic obstacles were gathered through the real-time detection results of the sensors mounted on the real autonomous driving platform, without a global cost map that provides prior information of the scenario.Owing to the limited sensing range and uncertain environmental changes, the planner runs in real time for timely planning and re-planning according to the detected obstacles. From the results shown in Fig. 6 and the video in supplementary materials, it can be seen that our planner can complete similar tasks successfully.

VIII. CONCLUSION

In this paper we proposed a new smooth path planning method based on Hybrid A*. First, clothoids were introduced as motion primitives for graph searching using Hybrid A*, to generate G^2-continuous paths that arein accordance with vehicle kinematics limits. Next, we reduced the time and complexity of computation for node expansion by pre-discretizing the orientation and curvature of motion primitives, and effectively reduced the search time. Then, we analyzed the problem of node re-expansion faced by traditional Hybrid A* and proposed a quintic spline interpolation scheme for local smoothing, thus preventing costly global optimization and collision checking. Through simulation and real-world experiments, our proposed method was proven to be effective in diverse complex dynamic environments.

REFERENCES

[1] C. Shitao, Z. Nanning, J. Zhiqiang, H. Yuhao, C. Yu, and Z. Zhuoli, "Autonomous driving: Cognitive construction and situation understanding," *SCIENCE CHINA Information Sciences*.

[2] M. Liang, Y. Jing, and Z. Meng, "A two-level path planning method for on-road autonomous driving," in *Second International Conference on Intelligent System Design & Engineering Application*, 2012.

[3] A. Piazzi, C. G. L. Bianco, and M. Romano, "eta3-splines for the smooth path generation of wheeled mobile robots," *IEEE Transactions on Robotics*, vol. 23, no. 5, pp. 1089–1095, 2007.

[4] T. Maekawa, T. Noda, S. Tamura, T. Ozaki, and K.-i. Machida, "Curvature continuous path generation for autonomous vehicle using b-spline curves," *Computer-Aided Design*, vol. 42, no. 4, pp. 350–359, 2010.

[5] T. Gu and J. M. Dolan, "On-road motion planning for autonomous vehicles," in *International Conference on Intelligent Robotics and Applications*. Springer, 2012, pp. 588–597.

[6] M. Pivtoraiko, R. A. Knepper, and A. Kelly, "Differentially constrained mobile robot motion planning in state lattices," *Journal of Field Robotics*, vol. 26, no. 3, pp. 308–333, 2010.

[7] D. Dolgov, S. Thrun, M. Montemerlo, and J. Diebel, "Path planning for autonomous vehicles in unknown semi-structured environments." *International Journal of Robotics Research*, vol. 29, no. 5, pp. 485–501, 2010.

[8] M. Likhachev, G. J. Gordon, and S. Thrun, "Ara*: Anytime a* with provable bounds on sub-optimality," in *Advances in neural information processing systems*, 2004, pp. 767–774.

[9] L. E. Kavraki, P. Svestka, J.-C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high-dimensional configuration spaces," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 4, 1996.

[10] Y. Kuwata, G. A. Fiore, E. Frazzoli, J. P. How, and J. Teo, "Motion planning for urban driving using rrt," in *IEEE/RSJ International Conference on Intelligent Robots & Systems*, 2008.

[11] M. Montemerlo, J. Becker, S. Bhat, H. Dahlkamp, D. Dolgov, S. Ettinger, D. Haehnel, T. Hilden, G. Hoffmann, B. Huhnke *et al.*, "Junior: The stanford entry in the urban challenge," *Journal of field Robotics*, vol. 25, no. 9, pp. 569–597, 2008.

[12] J. Reeds and L. Shepp, "Optimal paths for a car that goes both forwards and backwards," *Pacific journal of mathematics*, vol. 145, no. 2, pp. 367–393, 1990.

[13] J. Petereit, T. Emter, C. W. Frey, T. Kopfstedt, and A. Beutel, "Application of hybrid a* to an autonomous mobile robot for path planning in unstructured outdoor environments," in *ROBOTIK 2012; 7th German Conference on Robotics*. VDE, 2012, pp. 1–6.

[14] K. Kurzer, "Path planning in unstructured environments: A real-time hybrid a* implementation for fast and deterministic path generation for the kth research concept vehicle," 2016.

[15] D. Nemec, M. Gregor, E. Bubeníková, M. Hruboš, and R. Pirník, "Improving the hybrid a* method for a non-holonomic wheeled robot," *International Journal of Advanced Robotic Systems*, vol. 16, no. 1, p. 1729881419826857, 2019.

[16] J. Funke, P. Theodosis, R. Hindiyeh, G. Stanek, K. Kritatakirana, C. Gerdes, D. Langer, M. Hernandez, B. Müller-Bessler, and B. Huhnke, "Up to the limits: Autonomous audi tts," in *2012 IEEE Intelligent Vehicles Symposium*. IEEE, 2012, pp. 541–547.

[17] A. Broggi, P. Medici, P. Zani, A. Coati, and M. Panciroli, "Autonomous vehicles control in the vislab intercontinental autonomous challenge," *Annual Reviews in Control*, vol. 36, no. 1, pp. 161–171, 2012.

[18] D. Kiss and G. Tevesz, "Autonomous path planning for road vehicles in narrow environments: An efficient continuous curvature approach," *Journal of Advanced Transportation*, vol. 2017, no. 2, pp. 1–27, 2017.

[19] D. Kiss and D. Papp, "Effective navigation in narrow areas: A planning method for autonomous cars," in *IEEE International Symposium on Applied Machine Intelligence and Informatics*, 2017.

[20] T. Fraichard and A. Scheuer, "From reeds and shepp's to continuous-curvature paths," *IEEE Transactions on Robotics*, vol. 20, no. 6, pp. 1025–1035, 2004.

[21] S. J. Russell and P. Norvig, *Artificial intelligence: a modern approach*. Malaysia; Pearson Education Limited,, 2016.

[22] A. Piazzi and C. G. L. Bianco, "Quintic g2-splines for trajectory planning of autonomous vehicles," in *Proceedings of the IEEE Intelligent Vehicles Symposium 2000 (Cat. No. 00TH8511)*. IEEE, 2000, pp. 198–203.

[23] S. Chen, N.-N. Zheng, Y. Chen, and S. Zhang, "A novel integrated simulation and testing platform for self-driving cars with hardware in the loop," *IEEE Transactions on Intelligent Vehicles*, 2019.

[24] Y. Chen, S. Chen, T. Zhang, S. Zhang, and N. Zheng, "Autonomous vehicle testing and validation platform: Integrated simulation system with hardware in the loop," in *2018 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2018, pp. 949–956.