

ORIGINAL ARTICLE

Open Access



Generation and Selection of Driver-Behavior-Based Transferable Motion Primitives

Haijie Guan¹, Boyang Wang^{1,2*} , Jiaming Wei¹, Yaomin Lu¹, Huiyan Chen¹ and Jianwei Gong¹

Abstract

To integrate driver experience and heterogeneous vehicle platform characteristics in a motion-planning algorithm, based on the driver-behavior-based transferable motion primitives (MPs), a general motion-planning framework for offline generation and online selection of MPs is proposed. Optimal control theory is applied to solve the boundary value problems in the process of generating MPs, where the driver behaviors and the vehicle motion characteristics are integrated into the optimization in the form of constraints. Moreover, a layered, unequal-weighted MP selection framework is proposed that utilizes a combination of environmental constraints, nonholonomic vehicle constraints, trajectory smoothness, and collision risk as the single-step extension evaluation index. The library of MPs generated offline demonstrates that the proposed generation method realizes the effective expansion of MP types and achieves diverse generation of MPs with various velocity attributes and platform types. We also present how the MP selection algorithm utilizes a unique MP library to achieve online extension of MP sequences. The results show that the proposed motion-planning framework can not only improve the efficiency and rationality of the algorithm based on driving experience but can also transfer between heterogeneous vehicle platforms and highlight the unique motion characteristics of the platform.

Keywords: Autonomous vehicle, Motion planning, Motion primitives, Driver behavior, Heterogeneous vehicle platform

1 Introduction

The ever-increasing advancement of unmanned vehicle technology, especially multivehicle cooperative technology, will significantly promote the development of future intelligent transportation systems and unmanned combat systems and will affect future travel modes and combat patterns [1–3]. Motion planning is a crucial component in the unmanned vehicle system framework. Its main function is to generate a reference trajectory that satisfies the constraints of the environment and the vehicle itself [4–6]. Building a unified motion-planning algorithm

for heterogeneous vehicle platforms and using the driving experience to guide and accelerate the completion of planning tasks is a fundamental task for the further development of unmanned vehicle motion-planning algorithms. In both the motion-planning algorithm and the driving behavior representation algorithm, the decomposition of complex motion into motion primitives (MPs) can effectively improve the efficiency of the algorithm [7–10].

The essence of MP generation is to solve a set of boundary value problems, that is, to generate a set of paths connecting different target states [11]. The difficulty in solving the above problems lies in the segmentation strategy of the state space and the form of the curve connecting the start and end states. Graph search-based motion-planning algorithms, such as the typical

*Correspondence: wbythink@hotmail.com

¹ School of Mechanical Engineering, Beijing Institute of Technology, Beijing 100081, China

Full list of author information is available at the end of the article

A* [12, 13], D* [14, 15], and state lattice [16, 17], divide the environment space into basic grids and then generate MPs based on center points, corner points, or edges. To overcome the disadvantages of a grid-based MP generation method that cannot consider nonholonomic vehicle constraints, the Hybrid A* algorithm completes the generation of MPs based on the vehicle kinematic model [7, 18]. Each primitive used in the Hybrid A* algorithm is an arc generated with a fixed time scale and a fixed front wheel angle. The discrete optimization-based method uses spline curves or polynomial curves to generate MP candidate sets based on the offset points of the road centerline, which effectively limits the search space and improves the overall efficiency of the algorithm [19–21]. In addition, some methods use numerical optimization to realize the generation of MPs while considering vehicle dynamic constraints [22–24]. Although the MPs generated by using the above methods can meet the needs of a complex driving environment, they fail to integrate the driver's behavior information into the MP generation algorithm. Moreover, these methods generally lack consideration of the differences and connections of the heterogeneous platform motion characteristics.

The quality of the final path generated by the motion-planning algorithm depends not only on the quality of the MP itself but also on the rationality of the MP selection and connection [25]. The selection of MPs is the process of first considering the cost function represented by different constraints, then combining the cost according to a particular weight factor, and finally selecting the MP with the minimum cost [26]. The Dijkstra algorithm takes the shortest path as the extension cost of the MP [27]. The A* algorithm and its variants add heuristics based on the target state and Voronoi diagram to the original Dijkstra algorithm [28]. The sampling-based RRT path planning algorithm [29], on the one hand, takes the target point as a guide to the selection and extension of MPs; on the other hand, it randomly extends in any direction with a certain probability to achieve a comprehensive search of accessible areas. The discrete optimization-based method [19] utilizes the linear combination of the three extension cost values by considering static obstacles, dynamic obstacles, and path smoothness as the basis for the selection of MPs. Although the MP selection and extension method mentioned above can generate the desired driving trajectory, no attention is paid to the guiding role of the driver's experience.

The learning and generalization of the driving experience and characteristics have attracted the attention of many researchers in recent years [30–32]. Specific to trajectory-level driving behavior research, Guo et al. [33] researched the generation of human-like trajectories with the leader vehicle as an attractive force.

Although this method can cope with complex dynamic environments, the learning object of its behavior is the guide vehicle; that is, it is a follow-up human-like trajectory-generation method. Regarding the representation of human-like trajectories that target the behavior of the ego vehicle, Schnelle et al. [34] proposed a personalized trajectory optimization generation method in two scenarios: one with a single lane change and another with a double lane change. Zhao et al. also conducted a series of studies on lane-change scenes. First, they used the support vector machine algorithm to extract lane-change primitives from the lane-change trajectory data of surrounding vehicles [35, 36]. Subsequently, using the principle of trajectory matching, the generated human-like lane-change trajectory was integrated into a real-time motion-planning system [37, 38]. However, the aforementioned research has limitations in the application of scenes. It is often aimed at a specific scene, and there is no further study on how to combine multiple driving behaviors with the vehicle motion-planning system in a general driving environment.

In this paper, for the two heterogeneous platforms of wheeled Ackermann-steering vehicles and tracked skid-steering vehicles, a unified vehicle motion differential constraint is proposed. In addition, five types of equality constraints for typical driving behaviors and inequality constraints for vehicle platform characteristics are set, and an offline optimization method for driver-behavior-based transferable motion primitive (DBTMP) generation is formed. Based on the unique MP library established for different vehicle platforms, research on the selection of MPs was conducted under the Hybrid A* algorithm framework. However, there are two significant differences in the construction of the selection framework owing to the large difference between the offline generated MP library and the online generated MP library used by the Hybrid A* algorithm. First, the MP library generated offline is a collection of multiple MP sets containing the entire speed interval; therefore, a hierarchical structure is applied to restrict the MP candidate set. Second, because each candidate set contains both behavior primitives and general primitives, different weight coefficients are applied to the above two types of primitives when selecting and evaluating a single extended MP. Finally, a motion-planning algorithm based on offline generation and online selection of the MP library is proposed.

The main contributions of this study are the following:

- A unified motion-planning algorithm framework is proposed to complete the motion-planning tasks of heterogeneous platforms, achieving a balance

between the general solution framework and the characteristics of heterogeneous vehicle platforms.

- An offline MP generation method combining discrete driving behaviors and a vehicle kinematics model is proposed, and the guiding role of behavior primitives in the online selection layer of the motion-planning algorithm is highlighted.

The remainder of this paper is organized as follows. Section 1 details the problems to be solved in this study and explains the main parameters. Section 2 introduces the offline generation method for the MP library. Section 3 introduces an online selection framework for MPs. Section 4 demonstrates the unique MP libraries of two heterogeneous vehicle platforms and the results of motion planning in two typical scenarios. Finally, the conclusions and future work are presented in Section 5.

2 Problem Statement

The offline MP generation solves the optimization problem with driver behavior B and platform motion characteristics as constraints and trajectory smoothness as the objective function g . The selection of MPs is first to select the MP set from the MP library based on the passable area and velocity connection, and then utilize the cost function J as the evaluation index to choose the corresponding single extended MP from the MP set.

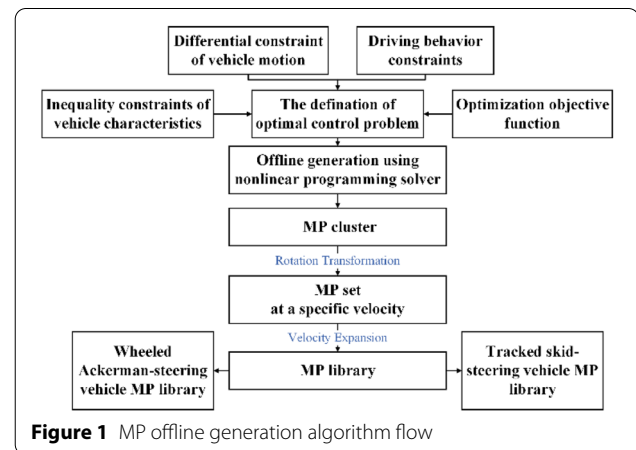
The definitions and explanations of the main parameters used in this study are as follows:

- $s(t) = [x(t), y(t), \theta(t)]^T \in R^{3 \times 1}$ is the unified state parameter set of each heterogeneous platform at time t , where $x(t)$ and $y(t)$ are the coordinate positions of the x and y axes, respectively, and $\theta(t)$ is the course angle. The defined coordinate system is the geodetic coordinate system xoy .
- $u(t) = [v_x(t), v_y(t), \omega_z(t)]^T \in R^{3 \times 1}$ is the unified control variable set of each heterogeneous platform at time t , where $v_x(t)$ and $v_y(t)$ are the velocities of the platform along the x and y axes, respectively, and $\omega_z(t)$ is the yaw rate around the z axis. The defined coordinate system is the vehicle body coordinate system x_aoy_a or x_toy_t .
- $u_a(t) = [v_{w_x}(t), \alpha(t)]^T \in R^{2 \times 1}$ is the unique control variable set of the wheeled Ackermann-steering vehicle at time t , where $v_{w_x}(t)$ is the velocity of the rear axle along the x_a axis and $\alpha(t)$ is the front wheel angle of the vehicle. The defined coordinate system is the vehicle body coordinate system x_aoy_a .
- $u_t(t) = [v_{l_x}(t), v_{r_x}(t)]^T \in R^{2 \times 1}$ is the unique control variable set of the tracked skid-steering vehicle at time t , where $v_{l_x}(t)$ and $v_{r_x}(t)$ are the velocities of the left and right tracks, respectively, along the x axis.

The defined coordinate system is the vehicle body coordinate system x_toy_t .

- $D = D_s \cdot D_{a/t}$ represents the mapping relationship between the platform state parameters and control variables, where D_s is the correlation between the unified state parameter set s and control variable set u . $D_{a/t}$ is the correlation between u and the unique control variable set $u_{a/t}$.
- $B = \{B_{SD}, B_{LC}, B_{UT}, B_{RT}, B_{TA}\}$ are the five defined types of driving behavior constraints: straight driving behavior B_{SD} , lane-changing behavior B_{LC} , U-shaped turn behavior B_{UT} , right-angle turn behavior B_{RT} , and turn-around behavior B_{TA} .
- U is the inequality constraint condition set according to the motion limit of the platform. U_a and U_t are for wheeled Ackermann-steering vehicles and tracked skid-steering vehicles, respectively.
- $J = \{J_e, J_n, J_s, J_c\}$ is the cost function when selecting a single MP, where J_e is the environmental constraint heuristic, J_n is the nonholonomic vehicle constraint heuristic, J_s is the evaluation function considering the smoothness of the curve, and J_c is the evaluation function considering the collision risk.

The method proposed in this study is dedicated to solving the following two challenges: (1) determining how to generate an MP library that can be transferred between heterogeneous platforms, considering the overall versatility and highlighting the specificity of the vehicle platform, and (2) integrating the driving behavior factors into the generation and selection algorithm of the MPs and then realizing the guidance of the driver's experience with the motion-planning algorithm.



3 Generation of MPs

The generation process of the MP library is shown in Figure 1. The entire processing flow can be decomposed into two main components: (1) the creation and solution of optimal control problems and (2) the rotation transformation and velocity expansion of a single MP cluster. The definition of the optimal control problem for a single MP generation lies at the core of the generation method.

The specific form of the optimal control problem is as follows:

$$\begin{aligned} & \min_{\mathbf{u}_{a/t}} : g(\mathbf{s}, \mathbf{u}) \\ & \text{subject to:} \\ & \begin{cases} \mathbf{s}(t) = \mathbf{B}, t \in [t_1, t_g], \\ \dot{\mathbf{s}}(t) = f(\mathbf{s}(t), \mathbf{u}(t)), t \in [t_1, t_g], \\ \mathbf{U}(t) \in \mathbf{U}, t \in [t_1, t_g], \end{cases} \end{aligned} \quad (1)$$

where $g(\mathbf{s}, \mathbf{u})$ is the objective function of the trajectory smoothness optimization based on the state and control variables; $\mathbf{s}(t) = \mathbf{B}$ is the constraint condition of the vehicle state based on the driving behavior; $\dot{\mathbf{s}}(t) = f(\mathbf{s}(t), \mathbf{u}(t))$ is the motion differential constraint describing the relationship between the control variables and the state parameters of the vehicle platform; $\mathbf{U}(t) \in \mathbf{U}$ is the inequality constraint condition set, and t_1 and t_g are the start and end times of the MP generation, respectively.

3.1 Vehicle Motion Differential Constraint

In this study, we ignore the impact of the suspension system on the vehicle motion response and regard the vehicle platform as a rigid body moving in a two-dimensional plane. The relationship between the unified state parameter set \mathbf{s} and the unified control variable set \mathbf{u} in a two-dimensional space is

$$\dot{\mathbf{s}}(t) = \mathbf{D}_s \cdot \mathbf{u}(t) = \begin{bmatrix} \cos \theta(t) & -\sin \theta(t) & 0 \\ \sin \theta(t) & \cos \theta(t) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} v_x(t) \\ v_y(t) \\ \omega_z(t) \end{bmatrix}. \quad (2)$$

According to the characteristics of heterogeneous platform actuators, the following mapping relationship between the unique control variable set $\mathbf{u}_{a/t}$ and the unified control variable set \mathbf{u} of different types of platforms can be established:

$$\mathbf{u}(t) = \mathbf{D}_{a/t} \cdot \mathbf{u}_{a/t}(t). \quad (3)$$

The relationship between the control variable \mathbf{u}_a of a typical wheeled Ackermann-steering vehicle and the unified state parameter set \mathbf{s} is

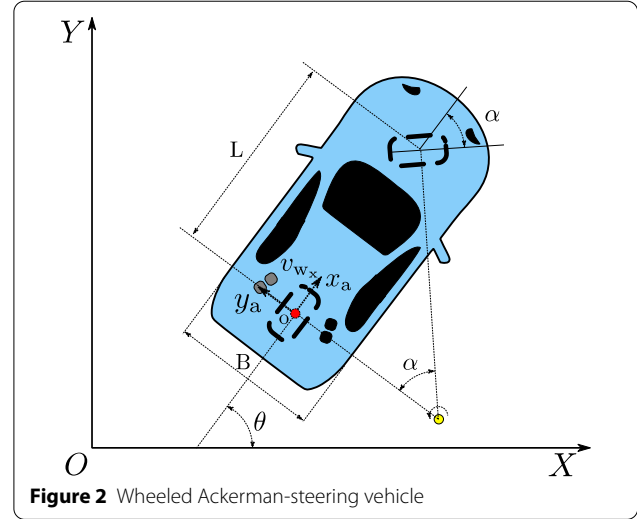


Figure 2 Wheeled Ackerman-steering vehicle

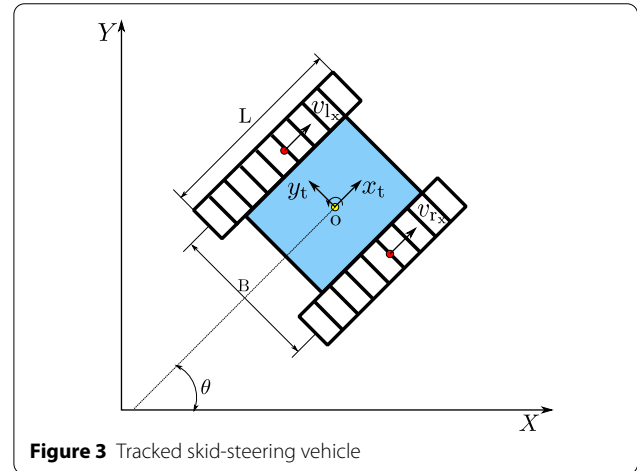


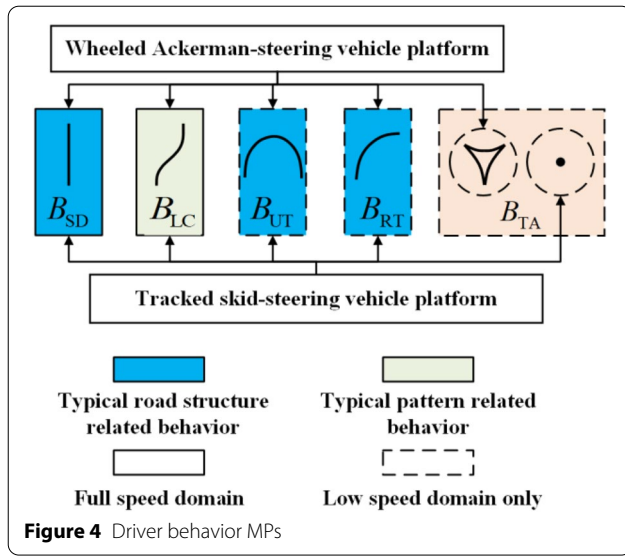
Figure 3 Tracked skid-steering vehicle

$$\dot{\mathbf{s}}(t) = \mathbf{D}_s \cdot \mathbf{D}_a \cdot \mathbf{u}_a(t) = \begin{bmatrix} v_{wx}(t) \cos \theta(t) \\ v_{wx}(t) \sin \theta(t) \\ v_{wx}(t) \tan \alpha(t)/L \end{bmatrix}, \quad (4)$$

where L is the distance between the front and rear axles of the vehicle. The related coordinate system and simplified model of the wheeled vehicle platform are shown in Figure 2.

The relationship between the control variable \mathbf{u}_t of a typical tracked vehicle and the unified state parameter set \mathbf{s} is

$$\dot{\mathbf{s}}(t) = \mathbf{D}_s \cdot \mathbf{D}_t \cdot \mathbf{u}_t(t) = \begin{bmatrix} (v_{lx}(t) + v_{rx}(t)) \cos \theta(t)/2 \\ (v_{lx}(t) + v_{rx}(t)) \sin \theta(t)/2 \\ (v_{rx}(t) - v_{lx}(t))/B \end{bmatrix}, \quad (5)$$



where B is the distance between the two tracks. The related coordinate system and simplified model of the tracked vehicle platform are shown in Figure 3.

3.2 Equality Constraints of Driving Behavior

This study defines five typical driving behaviors, as shown in Figure 4. These five behaviors can be classified into two main types: behaviors related to typical road structures (B_{SD} , B_{UT} , and B_{RT}) and behaviors related to typical driving behavior patterns (B_{LC} and B_{TA}). The MPs related to the road structure highlight the restraint effect of typical road structure factors on behavior, and the MPs related to the driving pattern highlight the restraint effect of the driver's behavior preferences. The remaining general MPs in the library are used as a supplement to the behavior primitives to improve the planning algorithm's adaptability to different driving environments. By increasing the priority of behavior primitives in the subsequent selection logic, we hope to better adapt to road structures and driving habits.

Among the above behavior primitives, B_{SD} and B_{LC} exist throughout the velocity range, and B_{UT} , B_{RT} , and B_{TA} exist only in the low-velocity range. Owing to the specificity of the vehicle structure, B_{TA} can be subdivided into B_{TAa} for wheeled vehicles and B_{TA_t} for tracked vehicles.

For straight driving behavior B_{SD} , U-shaped turn behavior B_{UT} , and right-angle turn behavior B_{RT} , only the course angle change between the start state t_1 and the end state t_g is constrained:

$$B_{SD} = \theta(t_g) - \theta(t_1) = 0, \quad (6)$$

$$B_{UT} = \theta(t_g) - \theta(t_1) = \pi, \quad (7)$$

$$B_{RT} = \theta(t_g) - \theta(t_1) = \pi/2. \quad (8)$$

For lane-changing behavior B_{LC} , in addition to restraining the course angle change deviation, the lateral distance change deviation d should also be constrained:

$$B_{LC} = \begin{bmatrix} \theta(t_g) - \theta(t_1) \\ [\cos \theta(t_g), \sin \theta(t_g)][x(t_g), y(t_g)]^T - d \end{bmatrix} = 0. \quad (9)$$

For the turn-around behavior B_{TAa} of a wheeled vehicle, it is necessary to restrict not only the beginning and end states but also the intermediate states accordingly:

$$B_{TAa} = \begin{bmatrix} \theta(t_g/3) - \theta(t_1) - \pi/3 \\ \theta(2t_g/3) - \theta(t_g/3) + 2\pi/3 \\ \theta(t_g) - \theta(2t_g/3) - \pi/3 \end{bmatrix} = 0. \quad (10)$$

Because the tracked vehicle can complete the task of turning around through the pivot turn movement, its constraint relationship B_{TA_t} is as follows:

$$B_{TA_t} = \begin{bmatrix} x(t_g) - x(t_1) \\ y(t_g) - y(t_1) \\ \theta(t_g) - \theta(t_1) - \pi \end{bmatrix} = 0. \quad (11)$$

3.3 Inequality Constraints of Vehicle Platform Characteristics

The constraints of vehicle platform characteristics mainly include four aspects (the input range of control variable \mathbf{u} , the limitation of yaw rate ω_z , the limitation of lateral acceleration a_y , and the limitation of the current generated MP velocity attribute v):

$$\mathbf{U} = \begin{cases} \mathbf{u} \in [-\mathbf{u}_{\max}, \mathbf{u}_{\max}], \\ \omega_z \leq 0.8, \\ a_y \leq 0.4g, \\ v \in [v_{\text{low}}, v_{\text{up}}]. \end{cases} \quad (12)$$

The inequality constraint relationship \mathbf{U}_a of the wheeled Ackermann-steering vehicle is

$$\mathbf{U}_a = \begin{cases} \alpha \in [-\alpha_{\max}, \alpha_{\max}], \\ v_{\omega_x} \tan \alpha / L \leq 0.8, \\ v_{\omega_x}^2 \tan \alpha / L \leq 0.4g, \\ v_{w_x} \in [v_{\text{low}}, v_{\text{up}}], \end{cases} \quad (13)$$

where α_{\max} is the maximum front wheel angle of the vehicle platform.

The inequality constraint relationship \mathbf{U}_t of the tracked skid-steering vehicle is

$$\mathbf{U}_t = \begin{cases} v_{l_x} \in [-v_{l_x_max}, v_{l_x_max}], \\ v_{r_x} \in [-v_{r_x_max}, v_{r_x_max}], \\ |v_{l_x} - v_{r_x}|/B \leq 0.8, \\ |v_{l_x}^2 - v_{r_x}^2|/2B \leq 0.4g, \\ (v_{l_x} + v_{r_x})/2 \in [v_{low}, v_{up}], \end{cases} \quad (14)$$

where $v_{l_x_max}$ and $v_{r_x_max}$ are the maximum speed limits of the left and right driving wheels, respectively.

3.4 Objective Function and Optimization Problem Solving

The objective function g mainly reflects the smoothness requirements of MP generation and includes the steering control input c_s and yaw rate of the vehicle platform ω_z :

$$g = \int_0^{t_g} (c_s^2 + \omega_z^2) dt. \quad (15)$$

These two parameters should be as small as possible because they correspond to the curvature of the generated curve and its rate of change, which directly affects the smoothness of the curve.

According to the difference in platform characteristics, the objective function g_a for the wheeled Ackermann-steering platform is given by

$$g_a = \int_0^{t_g} \alpha^2(t) + (v_{w_x}(t) \frac{\tan \alpha(t)}{L})^2 dt, \quad (16)$$

and the objective function g_t for the tracked skid-steering vehicle platform is given by

$$g_t = \int_0^{t_g} (\frac{2(v_{l_x}(t) - v_{r_x}(t))}{v_{l_x}(t) + v_{r_x}(t)})^2 + (\frac{v_{l_x}(t) - v_{r_x}(t)}{B})^2 dt. \quad (17)$$

In this study, the IPOPT [39] and CVODES [40] were applied to solve the optimal control problem.

3.5 Generation of the MP Library

Owing to the complexity of the real driving environment, having only the defined driving behavior MPs cannot fully satisfy the motion-planning problem in a complex environment. Therefore, to improve the applicability of the MP library, general MPs are generated as a supplement with the target course angle uniformly distributed on $[0, 2\pi]$ as the constraint.

In addition, in the online MP extension process, it is necessary to make the final heading of the MP selected in the previous MP cluster consistent with the heading of

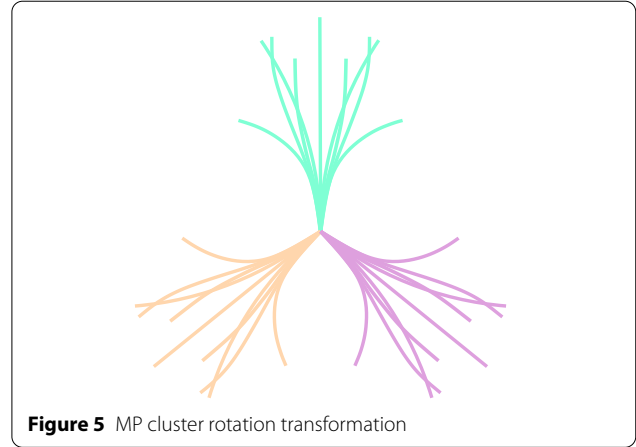


Figure 5 MP cluster rotation transformation

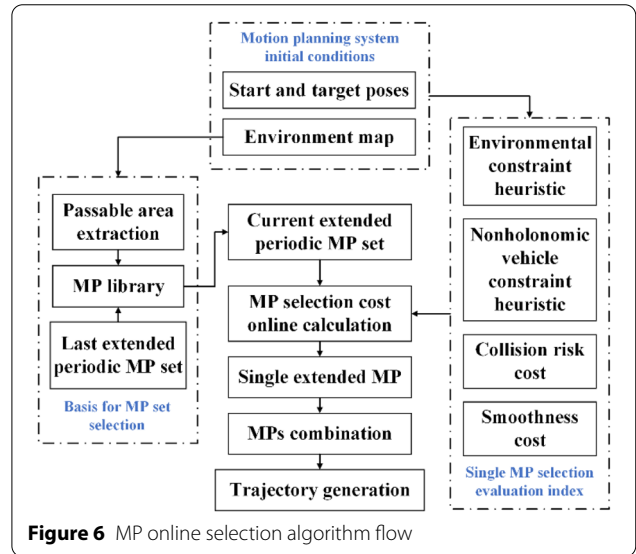


Figure 6 MP online selection algorithm flow

the straight driving behavior primitive in the next primitive cluster. This process can be performed either online or offline. To improve the efficiency of the algorithm, we used offline generation and online selection to complete the above process. After the MP cluster containing behavior primitives and general primitives is generated, an MP set covering the entire annular space with 36 evenly distributed MP clusters is generated by rotation transformation. The alignment between the MP clusters is completed by selecting the most appropriate angle from the MP set during online extension.

A simplified schematic of the MP set generation is shown in Figure 5; only a portion of the typical MPs and the rotation transformation at three positions are shown in the figure. The resulting MP library is a combination of multiple MP sets under different MP velocity attribute settings.

4 Selection of MPs

The MP selection process is shown in Figure 6. The entire processing flow can be decomposed into the selection of the MP sets, calculation of heuristics and costs, selection of a single MP, and extension of the MP sequence. The selection of MP sets refers to selecting the appropriate MP sets from the MP library based on passable areas in the environment map and the velocity attribute of the last extended periodic MP set. The heuristics include both the environmental constraint heuristic and the nonholonomic vehicle constraint heuristic. The environmental constraint heuristic is calculated from the starting and target poses in the grid map. The nonholonomic vehicle constraint heuristic is calculated based on the final state of every MP in the MP set and the target pose set by the planning system. The extension costs include the trajectory smoothing cost represented by the curve energy and collision risk cost, which is characterized by the distance from the obstacle. Finally, the linear combination of heuristics and costs is selected as the basis for single-step MP extension, and the selected multiple MPs are combined into an MP sequence to complete the motion-planning task.

4.1 Selection of MP Sets

The extraction of the single-step extended passable area is shown in Figure 7. The radius of the passable area, r_{area} , was calculated using the closest distance $d_{\min} = \min\{d_{\text{obs}_1}, d_{\text{obs}_2}, \dots, d_{\text{obs}_n}\}$ between the current position and the obstacle. The candidate set of MPs must satisfy the essential condition that the final state of every MP is within the passable area.

In addition, by considering the velocity attribute association of two adjacent periodic MPs, the MP set with a large velocity gap from the previous periodic is

eliminated from the candidate set. Finally, further narrowing of the candidate set was achieved.

4.2 Heuristics Considering Environmental and Nonholonomic Constraints

The calculation of the environmental constraint heuristic J_e takes the target position as the initial point and utilizes the breadth-first-search algorithm to realize the iterative expansion of the distance cost until it expands to the starting position. The specific algorithm flow is presented in Algorithm 1. ΔJ is the distance measurement value during the iterative process. According to the difference between the adjacent positions of the current grid, ΔJ has two values: $\{d, \sqrt{2}d\}$, where d is the grid width.

Algorithm 1 Pseudocode for computing environmental constraint heuristic ΔJ

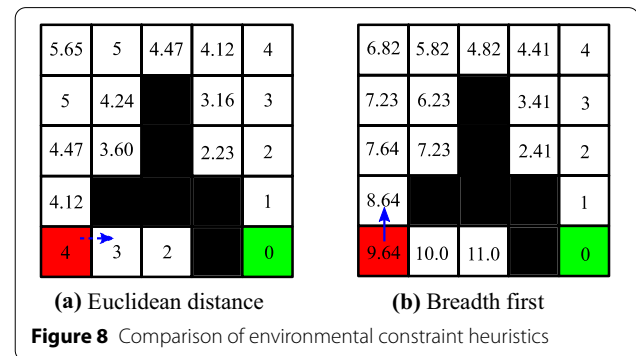
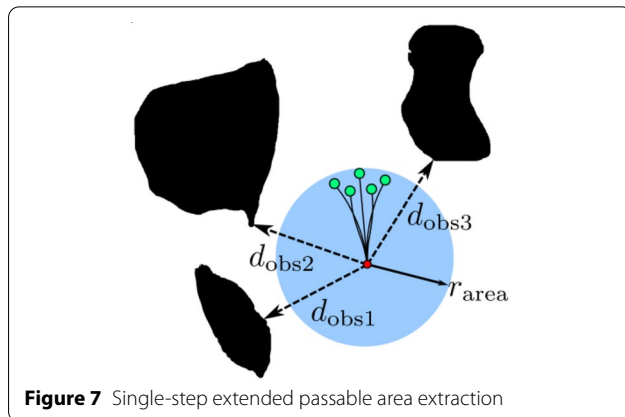
Require: starting point n_{start} ; target point n_{goal} ; open list S_{open} ; heuristic value $J_e(n)$; iteration distance value ΔJ

```

1:  $S_{\text{open}} \leftarrow \{n_{\text{goal}}\}$ 
2:  $\{J_e(n_1), J_e(n_2), \dots, J_e(n_{\text{all}})\} \leftarrow \infty, J_e(n_{\text{goal}}) \leftarrow 0$ 
3: while  $\emptyset \neq S_{\text{open}}$  do
4:    $n \leftarrow \text{Pop}(S_{\text{open}})$  //select the grid with the smallest heuristic value in the open list
5:    $\text{ExpandNode}(n, n_{\text{new}})$  // take the current grid as the center grid and expand in the direction of the surrounding eight grids
6:   if ( $\text{CollisionCheck}(n)$ ) then
7:     continue;
8:   else if ( $n_{\text{new}} = n_{\text{start}}$ ) then
9:     break;
10:  else
11:     $J_e(n_{\text{new}}) \leftarrow J_e(n) + \Delta J$  // heuristic value addition
12:     $S_{\text{open}} \leftarrow \{n_{\text{new}}\}$ 
13:  end if
14: end while

```

Figure 8 shows the comparison results of the environmental constraint heuristics calculated based on



the Euclidean distance and the breadth-first algorithm used in this study. It can be observed from Figure 8(a) that the calculated heuristics will first guide the MP to extend to the obstacle area, as shown by the dotted blue arrow in the figure. In contrast, in Figure 8(b), the MP is guided to the passable area, as indicated by the solid blue arrow in the figure. Therefore, the environmental constraint heuristic proposed in this study can avoid invalid extended searches and effectively improve the efficiency of the algorithm.

We also introduce the vehicle nonholonomic constraint heuristic J_n , taking the length of the Reeds-Shepp curve as the heuristic value:

$$J_n = \int_{t_{\text{start}}}^{t_{\text{goal}}} \sqrt{\dot{x}_r(t)^2 + \dot{y}_r(t)^2} dt. \quad (18)$$

The Reeds-Shepp curve is composed of a fixed curvature arc and a straight line. The curve can be defined as a set of poses $\Pi = (x_r(t), y_r(t), \theta_r(t)) \subset \mathbb{R}^2 \times [0, 2\pi)$, $t \in [t_{\text{start}}, t_{\text{goal}}]$. It can quickly generate a curve connecting the start and end points while meeting the position and course constraints of the two points.

The nonholonomic constraint heuristic introduces the evaluation of the target point's course reachability under the premise of considering the vehicle's motion characteristics, which makes up for the defect in which the environmental constraint heuristic only considers the target position's reachability.

Finally, a larger value of the environmental constraint heuristic and nonholonomic constraint heuristic is selected as the final heuristic value of the grid, that is, $J_1 = \max\{J_e, J_n\}$.

4.3 Extension Costs Considering Smoothness and Collision

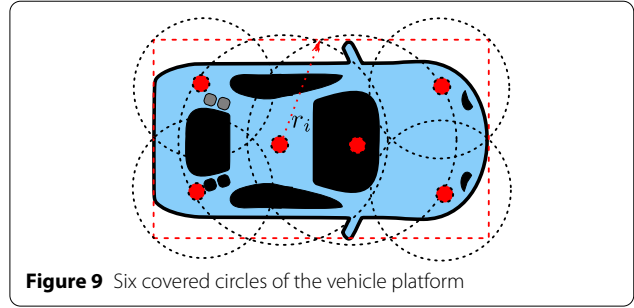
The extension cost proposed in this study mainly includes two aspects: trajectory smoothness and possible collision risk.

The extension cost value of smoothness, J_s , is calculated using the curve energy function as follows:

$$J_s = \sum_{i=2}^N \frac{(\kappa_{i-1}^2 + \kappa_i^2) \Delta s_i}{2}, \quad (19)$$

where N is the number of discrete points of the curve, κ_i is the corresponding curvature of the point, and Δs_i is the distance between two adjacent points. The essence of this function is the discrete representation of the curvature integral on the curve.

The extension cost value of collision risk, J_c , is calculated using the function



$$J_c = \frac{\sum_{i=1}^6 (d_{\text{obs}_i} - r_i)}{6}, \quad (20)$$

where d_{obs_i} is the distance from the center of each coverage circle to the nearest obstacle and r_i is the radius of each coverage circle. The vehicle platform is approximated by the six coverage circles shown in Figure 9 during the calculation process.

The above extension costs J_s and J_c are given different weights ω_s and ω_c . The linear combination after weighting is the final extension value:

$$J_2 = \omega_s J_s + \omega_c J_c. \quad (21)$$

The weighting coefficient ω_s is differentiated according to the type of MP as follows:

$$\omega_s = \begin{cases} \omega_{\text{sb}} & \text{if } \text{mp}_i = \text{Behavior MP}, \\ \omega_{\text{sg}} & \text{if } \text{mp}_i = \text{General MP}, \\ \omega_{\text{sd}} & \text{if } \text{mp}_i = \text{Reverse Direction}. \end{cases} \quad (22)$$

In principle, the relationships among the weight values of behavior MPs ω_{sb} , general MPs ω_{sg} , and MPs in reverse direction, ω_{sd} , is $\omega_{\text{sb}} < \omega_{\text{sg}} \ll \omega_{\text{sd}}$.

4.4 Evaluation Function of MP Selection

The total cost of selecting a single MP mp_i in the MP candidate set should be the sum of the corresponding heuristic and extension costs:

$$J(\text{mp}_i) = J_1(\text{mp}_i) + J_2(\text{mp}_i), \quad i \in [0, N_p], \quad (23)$$

where N_p is the number of independent MPs in the MP candidate set. In the MP extension process at each step, the MP with the lowest total cost in the MP candidate set is selected as an extension. Finally, the desired trajectory generation from the start position to the end position in the environment map can be realized.

Table 1 Platform geometric parameters and actuator constraints

| Type | L (m) | B (m) | α_{\max} (°) | $v_{l_{x_{\max}}}$ (m/s) | $v_{r_{x_{\max}}}$ (m/s) |
|-------|------------|------------|------------------------|-----------------------------|-----------------------------|
| P_a | 4.3 | 1.9 | $\pi/6$ | — | — |
| P_t | 5.2 | 3.3 | — | 16 | 16 |

5 Experimental Results and Discussion

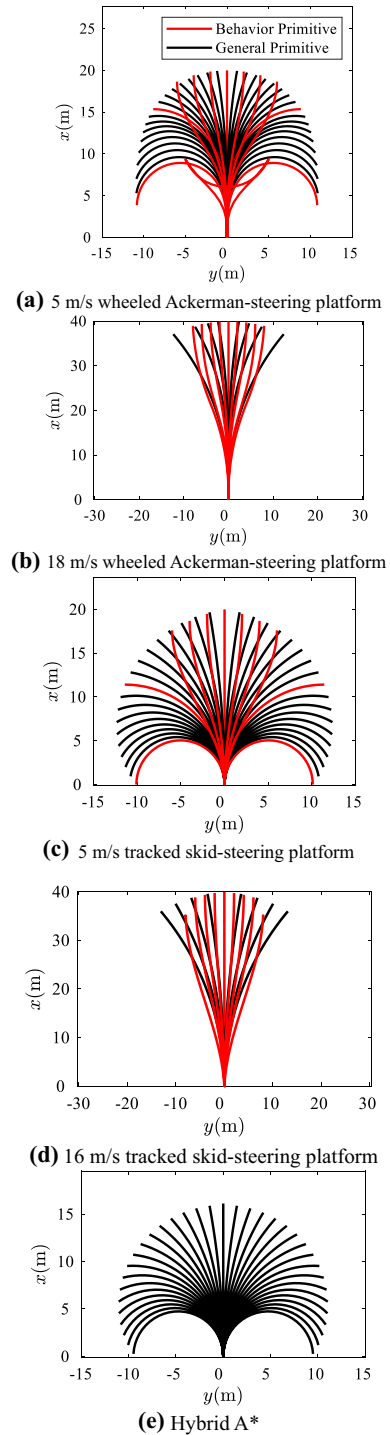
To verify the effect of the DBTMP A* motion-planning algorithm proposed in this study, the wheeled Ackermann-steering vehicle and tracked skid-steering vehicle were chosen, and the corresponding tests were conducted in both the simulation environment and the real environment. In the simulation environment, we conducted an in-depth comparative analysis of the algorithm performance in low-speed and high-speed scenes. In the real environment, we mainly focused on the applicability of the algorithm in real situations, especially the compatibility with the platform and the corresponding autonomous driving module. In addition, the classic Hybrid A* method was selected for comparison. The specific platform parameters are listed in Table 1, where P_a represents the wheeled Ackermann-steering platform and P_t represents the tracked skid-steering platform.

5.1 MP Library Offline Generation

The MP libraries for both platforms used a unified optimization framework during the generation process, and only the details of the vehicle characteristic constraints were changed during the platform transformation process.

Table 2 Generated MP library

| Platform | V_m (m/s) | d_m (m/s) | N_B | N_m | \bar{J}_s |
|----------|-------------|-------------------|-------|-------|-------------|
| P_a | 5 | 20 | 16 | 64 | 0.0643 |
| P_a | 10 | 25 | 23 | 20 | 0.1099 |
| P_a | 12 | 30 | 25 | 20 | 0.1748 |
| P_a | 14 | 35 | 25 | 20 | 0.1473 |
| P_a | 16 | 40 | 25 | 20 | 0.1242 |
| P_a | 18 | 40 | 9 | 8 | 0.1067 |
| P_a | 22 | 40 | 9 | 6 | 0.0933 |
| P_a | 26 | 40 | 9 | 6 | 0.0824 |
| P_a | 30 | 40 | 9 | 6 | 0.0784 |
| P_t | 5 | 15 | 18 | 64 | 1.3136 |
| P_t | 10 | 25 | 23 | 20 | 0.2426 |
| P_t | 12 | 30 | 25 | 20 | 0.1694 |
| P_t | 14 | 35 | 9 | 8 | 0.1307 |
| P_t | 16 | 40 | 9 | 8 | 0.1065 |
| Hybrid | — | Online adjustment | — | 37 | 0.2095 |

**Figure 10** Comparison of MP cluster generation results

The overall situation of the two platform MP libraries is given in Table 2, where v_m (m/s) is the velocity attribute of the MP cluster, d_m (m/s) is the longest distance covered by the MP cluster, N_B is the number of behavioral

primitives in the MP cluster, N_m is the number of general primitives in the MP cluster, and \bar{J}_s is the average curve energy value of each MP in the cluster. At the end of Table 2, the overall situation of the MP library used by the Hybrid A* method, which does not distinguish between vehicle platforms and velocity attributes, is presented. In Table 2, listed are the composition structures of the MP library generated offline for the wheeled Ackermann-steering platform and tracked skid-steering platform. The structure of the online generated MP library used by the Hybrid A* algorithm is also introduced as a comparison reference.

To show the generated MP clusters more intuitively, Figure 10 presents low-speed and high-speed typical MP clusters with wheeled platform velocity attributes of 5 and 18 m/s and tracked platform velocity attributes of 5 and 16 m/s. It also includes the MP cluster used by the Hybrid A* method.

The offline generation result of the MP library demonstrated that, regardless of the type of vehicle platform, the number of MPs gradually decreased with an increase in velocity. This is because the vehicle can achieve more diversified steering movements at low speeds, but the number of MPs at high speeds is greatly reduced owing to the weakening of the steering ability. Because the MPs are defined in discrete velocity ranges, an additional velocity-planning algorithm needs to be introduced in the actual application process to achieve a smooth velocity level transition.

In addition, when the velocity is 5 m/s, the MP cluster composition is significantly different from the composition at other velocities. The number of behavior primitives, N_{bp} , is reduced, while the number of general primitives, N_m , is significantly increased (including reverse primitives). When the vehicle speed is low, it is often necessary to address a more complicated environment, and general primitives can effectively improve the success rate of the planning algorithm in the corresponding situation.

Table 2 indicates that the average energy of the wheeled platform MPs in the low-speed range was lower than that of the tracked platform. Although the trajectory of the wheeled platform is smoother, and the adjustment of the course is gentle, the tracked platform has stronger and more aggressive steering adjustment ability at low speeds. In the high-speed range, the average energy of the MPs of the two platforms was basically the same, and the steering adjustment tended to be the same.

Hybrid A* utilized a circular arc generated by a fixed curvature as the basic MP. It realized neither the differentiation of the characteristics of heterogeneous platforms nor the separation of different speed intervals. In general, the types of MPs are relatively limited and do not have

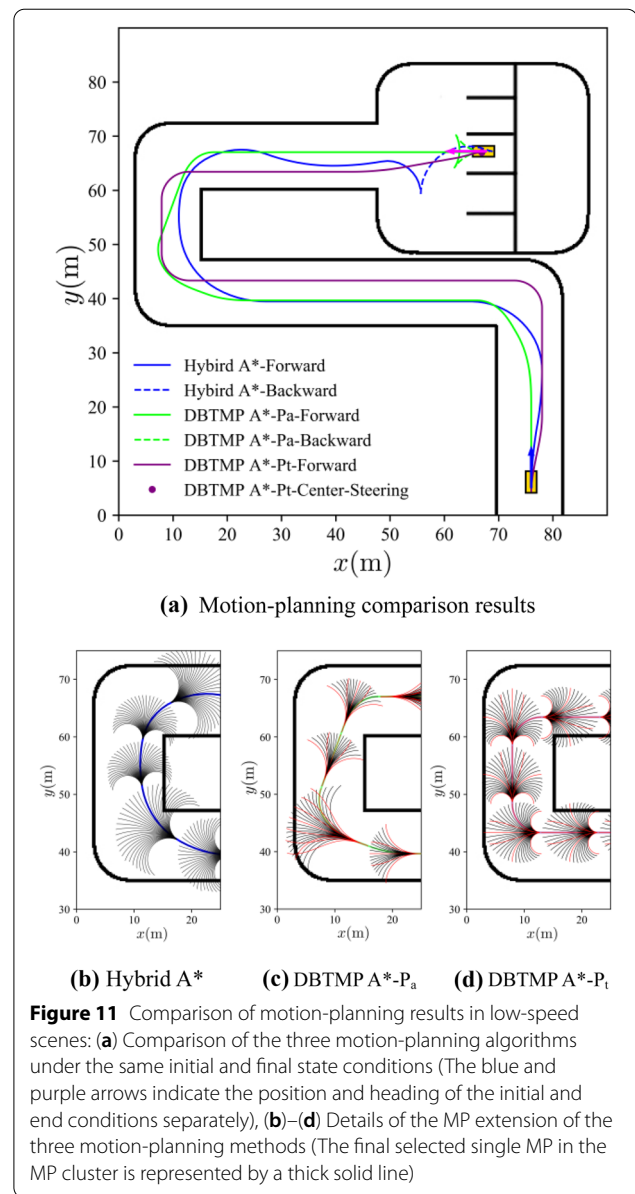


Figure 11 Comparison of motion-planning results in low-speed scenes: (a) Comparison of the three motion-planning algorithms under the same initial and final state conditions (The blue and purple arrows indicate the position and heading of the initial and end conditions separately), (b)–(d) Details of the MP extension of the three motion-planning methods (The final selected single MP in the MP cluster is represented by a thick solid line)

Table 3 Comparison of MP online selection results in a low-speed scene

| Method | \bar{J}_s | N_{be}/N_e | T |
|-------------------------|-------------|--------------|--------|
| Hybrid A* | 1.9371 | 0/23 | 132.43 |
| DBTMP A*-P _a | 0.0602 | 12/16 | 44.38 |
| DBTMP A*-P _t | 0.0578 | 17/18 | 48.15 |

the diversity of MPs in the MP library proposed in this study.

5.2 Comparative Analysis of Online Selection of MPs in the Simulation Environment

To demonstrate the performance of the MP online selection algorithm proposed in this study, a low-speed scene and a high-speed scene were designed to evaluate the motion-planning algorithm. In each scene, the motion-planning results of Hybrid A*, DBTMP A*-P_a, and DBTMP A*-P_t were compared. The planning results were solved using a laptop Lenovo IdeaPad Y700-15ISK with 16 gigabytes of memory and an eight-core Intel Core i7-6700 operating at 2.6 GHz, running under 64-bit Ubuntu 16.04. The entire motion-planning algorithm was written in C++ in the ROS system.

Figure 11 shows the results of motion planning in low-speed scenes. Table 3 presents the comparison results of the corresponding evaluation indicators, including the average curve energy \bar{J}_s , total number of MP extensions, N_e , number of behavior primitives during extension, N_{be} , and the solution time of the corresponding motion-planning algorithm, T (ms).

From the perspective of the smoothness of the motion-planning results, regardless of the platform used, the trajectory generated by the DBTMP A* algorithm is smoother. One of the reasons for this is that the overall MP library used by the DBTMP A* algorithm is smoother than that of the Hybrid A* method. Another reason is that the straight driving behavior primitives and general primitives use different weighting coefficients in the evaluation (as shown in Eq. (22)); therefore, the straight driving behavior primitives occupy a larger proportion of the entire trajectory, which makes the evaluation index of smoothness significantly lower than that of the Hybrid A* algorithm.

From the perspective of the extension times of MPs, the extension times of DBTMP A*-P_a and DBTMP A*-P_t were only 69.56% and 78.26%, respectively, of that of the Hybrid A* method. This is mainly because the introduction of behavior primitives increases the extension category of a single MP and improves its adaptability to the environment so that it can complete the trajectory-planning task with fewer MP combinations. This is most evident in the local scene of the parking scene. Regardless of the vehicle platform, the DBTMP A* algorithm used only one MP to complete the planning task, whereas Hybrid A* achieved the task by joining five independent MPs and did not highlight the characteristics of the two platforms.

From the perspective of the solution time of motion-planning tasks, the DBTMP A* algorithm only accounted for 35% of the overall solution time of the Hybrid A* algorithm. The main reason for this difference is that the MP sets used by the Hybrid A* algorithm are generated online based on the passable radius, while the MP

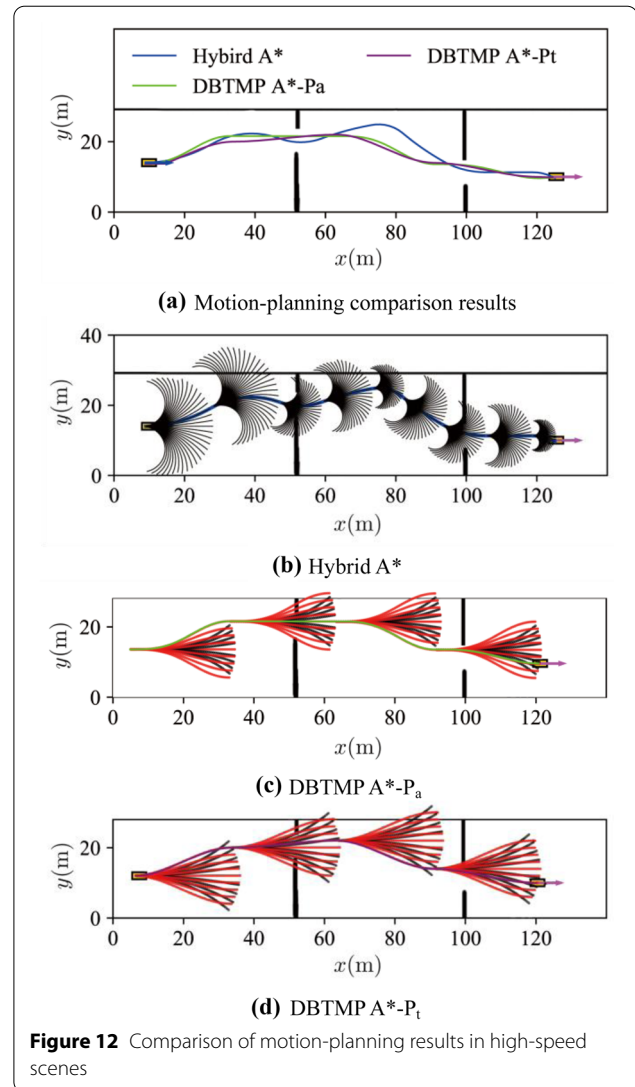


Figure 12 Comparison of motion-planning results in high-speed scenes

Table 4 Comparison of MP online selection results in a high-speed scene

| Method | \bar{J}_s | N_{be}/N_e | T |
|-------------------------|-------------|--------------|-------|
| Hybrid A* | 0.0498 | 0/9 | 94.32 |
| DBTMP A*-P _a | 0.0093 | 4/4 | 57.28 |
| DBTMP A*-P _t | 0.0071 | 4/4 | 55.64 |

sets selected by the DBTMP A* algorithm are generated offline.

The comparison results of motion planning in high-speed scenes are shown in Figure 12, and Table 4 presents the corresponding evaluation indicators.

From the motion-planning results of Figure 12 and Table 4, one can observe that the algorithm proposed

in this study took advantage of the DBTMP library to achieve lower curve energy, fewer extension times, shorter solution time, and more reasonable planning results. This is mainly because the DBTMP library restricts and distinguishes the MPs that can be selected according to the velocity attribute and introduces the standard lane-change driving behavior primitives in high-speed scenes as supplements. This enables the driving experience to be successfully integrated into the MP generation and selection process proposed in this study. Although the Hybrid A* algorithm can also plan the corresponding collision-free trajectory, it does not conform to the driver's driving experience in high-speed scenes. The several imperfections of the Hybrid A* algorithm in high-speed scenes are mainly due to the composition of primitive clusters, which do not distinguish between high-speed and low-speed scenes.

5.3 Applicability of the Algorithm in a Real Environment

The performance of the proposed algorithm in a real environment was verified by using two vehicle platforms the wheeled Ackermann-steering platform shown in Figure 13(a) and the tracked skid-steering platform shown in Figure 13(b). The two platforms are equipped with the same equipment, including a RoboSense 32-lidar system and two RoboSense 16-lidar systems, a Simpak982 GNSS

receiver, an FOSN2 inertial navigation system, an AVT-1290c camera, and two industrial personal computers.

Before the experiment, we obtained the environmental map required for the entire planning task using a multi-source simultaneous localization and mapping algorithm. In the experiment, the fusion positioning system provided centimeter-level real-time positioning accuracy at a frequency of 50 Hz. Based on an a priori map, the environment perception equipment carried by the vehicle added real-time environmental information to the map at a frequency of 5 Hz. Model predictive control was selected as the vehicle motion control algorithm, and the comprehensive tracking error in the off-road environment was < 0.5 m.

The purpose of the experiment in real scenes is to test the performance and efficiency of the algorithm under complex environmental conditions. Figure 14 shows the comparison results of the motion-planning algorithm at two time points during unmanned driving. These two moments were the starting moment of planning and the trigger moment of re-planning after turning. Because the Hybrid A* algorithm only made extremely limited adaptability adjustments for the two platforms, only the wheeled Ackermann-steering platform was selected as the verification platform. T_i represents the initial moment of the planning, and T_r represents the moment of replanning. Figure 15 shows the efficiency indices of the three motion-planning algorithms. Where the single-step extension time refers to the time required to select the most appropriate MP from the MP library at each MP extension step.

Judging from the experimental results in real scenes, the algorithm proposed in this study can not only be applied to unmanned systems in real time but can also achieve higher efficiency and more reasonable planning results. The experimental conclusions in the real environment are consistent with the simulation results.

6 Conclusions

A unified motion-planning algorithm framework based on MP generation and selection for heterogeneous vehicle platforms is proposed.

- (1) An optimized generation method is applied to realize the offline construction of a heterogeneous platform MP library. The generated MP library not only distinguishes MPs according to the velocity attributes and platform characteristics, but it also introduces five types of behavior primitives to expand the types of MPs.
- (2) The layered unequal-weighted MP online selection framework makes full use of the characteristics of

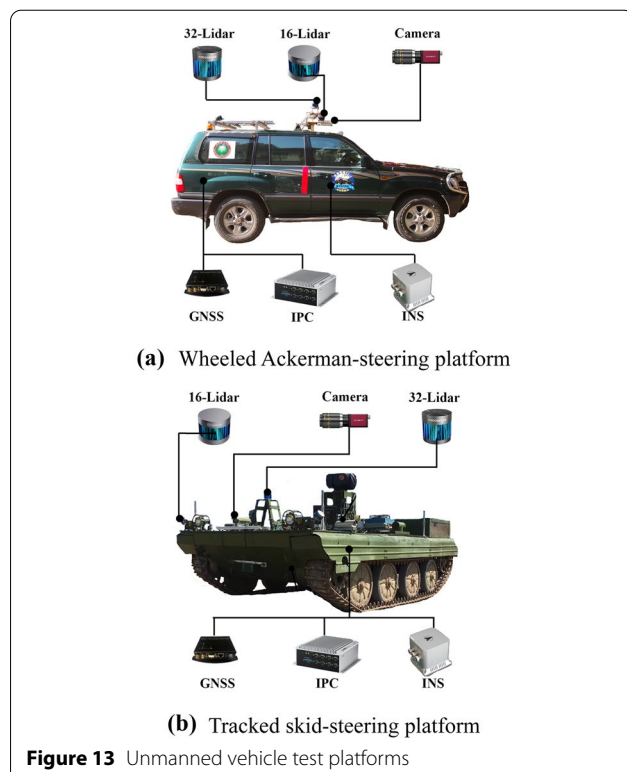


Figure 13 Unmanned vehicle test platforms

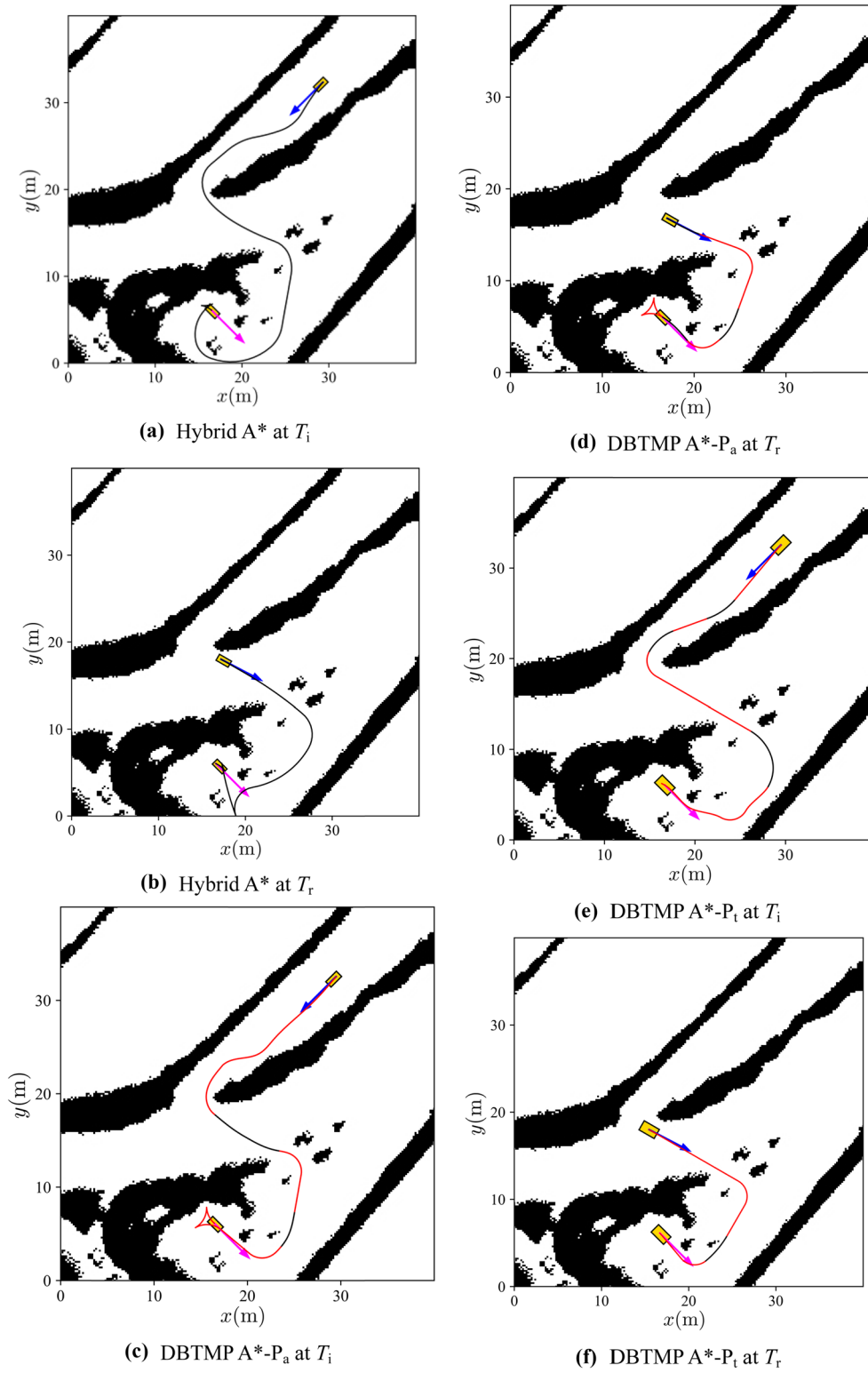


Figure 14 Comparison results of the three motion-planning algorithms in real environments

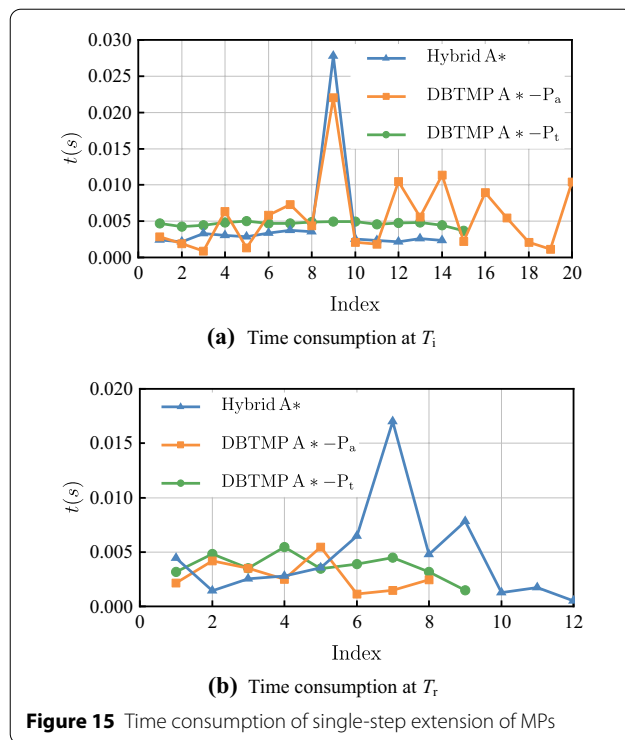


Figure 15 Time consumption of single-step extension of MPs

the platform-specific MP library generated offline. The velocity connection relationship of the MP set is limited, and the selection weight of the single extended behavior primitive and general primitive is distinguished.

- (3) The DBTMP A* motion-planning algorithm proposed in this study not only retains the strong environmental adaptability of the original Hybrid A* algorithms but also highlights the characteristic differences between heterogeneous vehicle platforms. Moreover, the proposed method effectively utilizes the driving experience to complete the reasonable guidance of the planning results, improving the trajectory smoothness and significantly reducing the required solution time.

Acknowledgements

Not applicable.

Authors' Contributions

BW was in charge of the whole trial; HC and JG guided the writing of the manuscript; HG and BW wrote the manuscript; JW and YL assisted with the simulation analyses. All authors read and approved the final manuscript.

Author Information

Haijie Guan, born in 1995, is currently a Ph.D. candidate at School of Mechanical Engineering, Beijing Institute of Technology, China. He received his master degree from Beijing Institute of Technology, China, in 2019.

Boyang Wang, born in 1991, is currently a postdoctoral researcher at Peking University, China. He received his Ph.D. degree from Beijing Institute of Technology, China, in 2020.

Jiaming Wei, born in 1996, is currently an engineer at Aviation Industry Corporation of China, China. He received his master degree from Beijing Institute of Technology, China, in 2021.

Yaomin Lu, born in 1996, is currently a master candidate at School of Mechanical Engineering, Beijing Institute of Technology, China. She received her bachelor degree from Hunan University, China, in 2019.

Huiyan Chen, born in 1961, is currently a professor at School of Mechanical Engineering, Beijing Institute of Technology, China. He received his Ph.D. degree from Beijing Institute of Technology, China, in 2004.

Jianwei Gong, born in 1969, is currently a professor at School of Mechanical Engineering, Beijing Institute of Technology, China. He received his Ph.D. degree from Beijing Institute of Technology, China, in 2002.

Funding

Supported by National Natural Science Foundation of China (Grant Nos. 91420203 and 61703041).

Competing Interests

The authors declare no competing financial interests.

Author Details

¹School of Mechanical Engineering, Beijing Institute of Technology, Beijing 100081, China. ²School of Artificial Intelligence, Peking University, Beijing, China.

Received: 9 August 2020 Revised: 1 December 2021 Accepted: 6 January 2022

Published online: 12 February 2022

References

- [1] S W Loke. Cooperative automated vehicles: A review of opportunities and challenges in socially intelligent vehicles beyond networking. *IEEE Transactions on Intelligent Vehicles*, 2019, 4(4): 509–518.
- [2] S Wang, Y Zhang, Z Liao. Building domain-specific knowledge graph for unmanned combat vehicle decision making under uncertainty. *2019 Chinese Automation Congress (CAC)*. IEEE, 2019: 4718–4721.
- [3] F Lin, Y Zhang, Y Zhao, et al. Trajectory tracking of autonomous vehicle with the fusion of DYC and longitudinal-lateral control. *Chinese Journal of Mechanical Engineering*, 2019, 32(1): 1–16.
- [4] L Chen, X Hu, W Tian, et al. Parallel planning: A new motion planning framework for autonomous driving. *IEEE/CAA Journal of Automatica Sinica*, 2018, 6(1): 236–246.
- [5] L Claussmann, M Revilloud, D Gruyer, et al. A review of motion planning for highway autonomous driving. *IEEE Transactions on Intelligent Transportation Systems*, 2019, 21(5): 1826–1848.
- [6] C Katrakazas, M Quddus, W H Chen, et al. Real-time motion planning methods for autonomous on-road driving: State-of-the-art and future research directions. *Transportation Research Part C: Emerging Technologies*, 2015, 60: 416–442.
- [7] D Dolgov, S Thrun, M Montemerlo, et al. Path planning for autonomous vehicles in unknown semi-structured environments. *The International Journal of Robotics Research*, 2010, 29(5): 485–501.
- [8] B Wang, J Gong, H Chen. Motion primitives representation, extraction and connection for automated vehicle motion planning applications. *IEEE Transactions on Intelligent Transportation Systems*, 2020, 21(9): 3931–3945.
- [9] W Wang, W Han, X Na, et al. A probabilistic approach to measuring driving behavior similarity with driving primitives. *IEEE Transactions on Intelligent Vehicles*, 2019, 5(1): 127–138.
- [10] T Taniguchi, S Nagasaka, K Hitomi, et al. Sequence prediction of driving behavior using double articulation analyzer. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2015, 46(9): 1300–1313.
- [11] K Bergman, O Ljungqvist, D Axehill. Improved path planning by tightly combining lattice-based path planning and optimal control. *IEEE Transactions on Intelligent Vehicles*, 2020, 6(1): 57–66.
- [12] S Kammel, J Ziegler, B Pitzer, et al. Team AnnieWAY's autonomous system for the 2007 DARPA Urban Challenge. *Journal of Field Robotics*, 2008, 25(9): 615–639.
- [13] Y L Chen, V Sundareswaran, C Anderson, et al. Terramax™: Team oshkosh urban robot. *Journal of Field Robotics*, 2008, 25(10): 841–860.

- [14] C Urmson, J Anhalt, D Bagnell, et al. Autonomous driving in urban environments: Boss and the urban challenge. *Journal of Field Robotics*, 2008, 25(8): 425–466.
- [15] D Ferguson, T M Howard, M Likhachev. Motion planning in urban environments. *Journal of Field Robotics*, 2008, 25(11–12): 939–960.
- [16] M McNaughton, C Urmson, J M Dolan, et al. Motion planning for autonomous driving with a conformal spatiotemporal lattice. *2011 IEEE International Conference on Robotics and Automation*. IEEE, 2011: 4889–4895.
- [17] T M Howard, C J Green, A Kelly, et al. State space sampling of feasible motions for high-performance mobile robot navigation in complex environments. *Journal of Field Robotics*, 2008, 25(6–7): 325–345.
- [18] M Montemerlo, J Becker, S Bhat, et al. Junior: The stanford entry in the urban challenge. *Journal of field Robotics*, 2008, 25(9): 569–597.
- [19] X Hu, L Chen, B Tang, et al. Dynamic path planning for autonomous driving on various roads with avoidance of static and moving obstacles. *Mechanical Systems and Signal Processing*, 2018, 100: 482–500.
- [20] Y Jiang, J Gong, G Xiong, et al. Research on differential constraints-based planning algorithm for autonomous-driving vehicles. *Acta Automatica Sinica*, 2013, 39(12): 2012–2020.
- [21] J Jiang, Q Wang, J Gong, et al. Research on temporal consistency and robustness in local planning of intelligent vehicles. *Acta Automatica Sinica*, 2015, 41(1): 518–527.
- [22] X Yang, Li-Ping L, Duan-Feng C, et al. Unified modeling of trajectory planning and tracking for unmanned vehicle. *Acta Automatica Sinica*, 2019, 45(4): 799–806.
- [23] L B Cremean, T B Foote, J H Gillula, et al. Alice: An information-rich autonomous vehicle for high-speed desert navigation. *Journal of Field Robotics*, 2006, 23(9): 777–810.
- [24] J Ziegler, P Bender, T Dang, et al. Trajectory planning for Bertha—A local, continuous method. *2014 IEEE Intelligent Vehicles Symposium Proceedings*. IEEE, 2014: 450–457.
- [25] A Desai, M Collins, N Michael. Efficient kinodynamic multi-robot replanning in known workspaces. *2019 International Conference on Robotics and Automation*. IEEE, 2019: 1021–1027.
- [26] C Chamzas, A Shrivastava, L E Kavraki. Using local experiences for global motion planning. *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019: 8606–8612.
- [27] S J Anderson, S B Karumanchi, K Iagnemma. Constraint-based planning and control for safe, semi-autonomous operation of vehicles. *2012 IEEE Intelligent Vehicles Symposium*. IEEE, 2012: 383–388.
- [28] Q Wang, C S Wiegardt, Y Jiang, et al. Generalized path corridor-based local path planning for nonholonomic mobile robot. *2015 IEEE 18th International Conference on Intelligent Transportation Systems*. IEEE, 2015: 1922–1927.
- [29] S Karaman, M R Walter, A Perez, et al. Anytime motion planning using the RRT. *2011 IEEE International Conference on Robotics and Automation*. IEEE, 2011: 1478–1483.
- [30] C Lu, F Hu, D Cao, et al. Transfer learning for driver model adaptation in lane-changing scenarios using manifold alignment. *IEEE Transactions on Intelligent Transportation Systems*, 2019, 21(8): 3281–3293.
- [31] H WANG, Z XIA, W CHEN. Lane departure assistance control based on extension combination of steering and braking systems considering human-machine coordination. *Journal of Mechanical Engineering*, 2019, 55(4): 135–147.
- [32] C Sentouh, A T Nguyen, J J Rath, et al. Human-machine shared control for vehicle lane keeping systems: A Lyapunov-based approach. *IET Intelligent Transport Systems*, 2019, 13(1): 63–71.
- [33] C Guo, K Kidono, M Ogawa. Learning-based trajectory generation for intelligent vehicles in urban environment. *2016 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2016: 1236–1241.
- [34] S Schnelle, J Wang, H Su, et al. A driver steering model with personalized desired path generation. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2016, 47(1): 111–120.
- [35] H Zhao, C Wang, Y Lin, et al. On-road vehicle trajectory collection and scene-based lane change analysis: Part I. *IEEE Transactions on Intelligent Transportation Systems*, 2016, 18(1): 192–205.
- [36] W Yao, Q Zeng, Y Lin, et al. On-road vehicle trajectory collection and scene-based lane change analysis: Part II. *IEEE Transactions on Intelligent Transportation Systems*, 2016, 18(1): 206–220.
- [37] D Xu, Z Ding, H Zhao, et al. Naturalistic lane change analysis for human-like trajectory generation. *2018 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2018: 1393–1399.
- [38] X He, D Xu, H Zhao, et al. A human-like trajectory planning method by learning from naturalistic driving data. *2018 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2018: 339–346.
- [39] A Wächter, L T Biegler. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming*, 2006, 106(1): 25–57.
- [40] R Serban, A C Hindmarsh. CVODES: An ODE solver with sensitivity analysis capabilities. Technical Report UCRL-JP-200039, Lawrence Livermore National Laboratory, 2003.

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► [springeropen.com](https://www.springeropen.com)