Spatio-temporal heuristic method: a trajectory planning for automatic parking considering obstacle behavior

Nianfei Gan, Miaomiao Zhang, Bing Zhou and Tian Chai
HNU College of Mechanical and Vehicle Engineering, Hunan University, Changsha, China
Xiaojian Wu
School of Mechanical and Electrical Engineering, Nanchang University, Nanchang, China and
Intelligent and Connected Vehicles Research Institute, Jiangling Motors Corporation Ltd., Nanchang, China, and
Yougang Bian
HNU College of Mechanical and Vehicle Engineering, Hunan University, Changsha, China

Abstract

Purpose – The purpose of this paper is to develop a real-time trajectory planner with optimal maneuver for autonomous vehicles to deal with dynamic obstacles during parallel parking.

Design/methodology/approach – To deal with dynamic obstacles for autonomous vehicles during parking, a long- and short-term mixed trajectory planning algorithm is proposed in this paper. In long term, considering obstacle behavior, A-star algorithm was improved by RS curve and potential function via spatio-temporal map to obtain a safe and efficient initial trajectory. In short term, this paper proposes a nonlinear model predictive control trajectory optimizer to smooth and adjust the trajectory online based on the vehicle kinematic model. Moreover, the proposed method is simulated and verified in four common dynamic parking scenarios by ACADO Toolkit and QPOASE solver.

Findings – Compared with the spline optimization method, the results show that the proposed method can generate efficient obstacle avoidance strategies, safe parking trajectories and control parameters such as the front wheel angle and velocity in high-efficient central processing units.

Originality/value – It is aimed at improving the robustness of automatic parking system and providing a reference for decision-making in a dynamic environment.

Keywords Spatio-temporal heuristic method, Automatic parking, Trajectory planning

Paper type Research paper

1. Introduction

With the combination of big data cloud and artificial intelligence technology in the automobile industry, autonomous driving technology is penetrated in more application scenarios (Khayyam *et al.*, 2020). As a relatively comprehensive subsystem of advanced driver assistance systems, the automatic parking system is widely loved by users for its comfort and safety. To promote the implementation of autonomous vehicles (AVs) and reduce the burdens faced by novice drivers in complex parking environments, researchers have conducted a series of studies on how to improve the robustness and safety of trajectory planners. In a broad sense, current methods include the path-velocity decomposition (PVD) approach and the directly planning approach.

In terms of the PVD approach, these trajectory planners consist of two periods. One period is planning path in X-Y plane. For parking in narrow spaces, Zhou *et al.* (2020b) designed 24 kinds of parking paths, including straight lines and curves. To plan a collision-free path on unstructured road, the sample-and-search-based methods were explored based on grid map, such as RRT (Han *et al.*, 2011; Li *et al.*, 2019a, b), Dijkstra (Zhang and Zhao, 2014) and A-star (Cheng *et al.*, 2014). However, due to the limits of poor smoothness and higher solving precision, Tang *et al.* (2018) used the B-spline

This paper forms part of a special section "Intelligent Safety for Intelligent and Connected", guest edited by Jun Li.

This work was supported jointly by the National Natural Science Foundation of China (Nos. 51875184 and 52002163).

Received 14 January 2022
Revised 9 May 2022
19 May 2022
29 May 2022
16 June 2022
Accepted 17 June 2022

The current issue and full text archive of this journal is available on Emerald Insight at: **https://www.emerald.com/insight/2399-9802.htm**

Journal of Intelligent and Connected Vehicles
5/3 (2022) 177–187
Emerald Publishing Limited [ISSN 2399-9802]
[DOI 10.1108/JICV-01-2022-0002]

curve to smooth path while Jhang and Lian (2020) designed a fast exploration random tree (called Bi-RRT*) to plan a collision-free path rapidly.

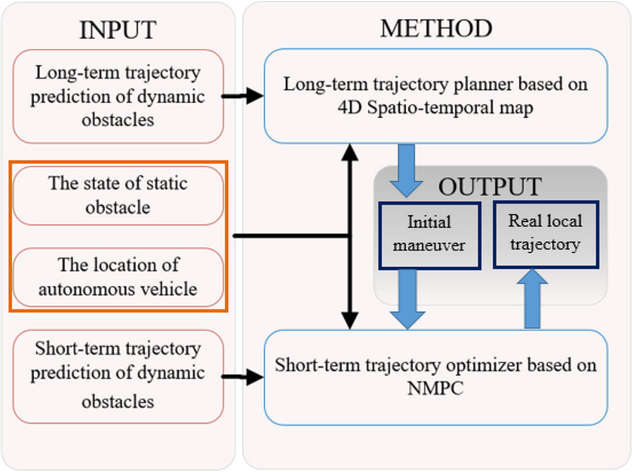
Velocity planning, the other period, refers to the pathplanning algorithm of Baidu Apollo, Zhou *et al.* (2020a) explored spline interpolation method to achieve a smooth velocity profile along the collision-free path planned in distance-time (S-T) space. Though the decomposition problem is easier than directly handling, the results of path planning and velocity planning are seldom optimal simultaneously, and the control quantity is not feasible in many narrow channels.

As to the directly planning approach, Qian *et al.* (2019, 2020) and Gen *et al.* (2019) adopted the pseudo-spectral method to optimize the autonomous vehicle's path, speed and front wheel angle by establishing starting point constraint, target point constraint, path constraint, kinematic differential equation and minimal time objective function. It is high accuracy but time-consuming. Thus, to shorten calculation time, Li *et al.* (2016, 2020) proposed the A-star algorithm and safe channel (STC) method to simplify solution regions and constraints before applying the simultaneous strategy. However, the second-based central processing unit calculation scale still fails to meet the real-time requirements in dynamic environment (Sheng *et al.*, 2021).

In practice, due to environmental uncertainty, automatic parking system is not absolutely static such as vehicle–vehicle interaction and pedestrian–vehicle interaction in some roadside parking and large parking lots. Based on safe distance requirements, Xie (2020) combined A-star and Reeds-Shepp to re-plan vehicle route but failed to take dynamic obstacles during parking process or near parking slot into consideration. Aiming to protect pedestrians from collisions in parking space, Hu *et al.* (2020) firstly combined constant velocity offset curves and sine curves to create the initial parking trajectory, and then adjusted parking speed on grounds of model predictive controller to keep a safe distance from pedestrians. Nevertheless, it was only suitable for large parking scenes or the situations in which dynamic obstacles appear behind the vehicles. Meanwhile, based on expert parking experience, obstacle orientation and target orientation information, Nakrani and Joshi (2021) presented a novel fuzzy-based obstacle avoidance controller to handle the obstacle avoidance task by changing the direction without losing sight of the primary goal of parallel parking. Although the parking task was highly completed and computation was also fast, the impact of obstacle intention on the parking task was ignored.

In conclusion, existing automatic parking trajectory planning studies seldom concentrate on dynamic obstacles or the link between path re-planning and waiting strategies. However, intelligent drivers can adjust the parking path and speed at the same time. Therefore, to balance the advantages of both strategies in a dynamic parking environment, this paper designs a novel trajectory planner, namely, a long- and short-term mixed trajectory planner. As shown in Figure 1, with a view to obstacles' long-term behavior, A-star algorithm is improved to search for a global optimal maneuver by establishing a spatio-temporal map, RS curve and potential function. Then, on the basis of short-term predictions of obstacles' trajectory, nonlinear model predictive control (NMPC) is applied to further optimize the initial maneuvers with vehicle kinematic constraints to get a safer, more feasible and efficient trajectory.

Figure 1 Framework of long- and short-term mixed trajectory planner


INPUT
Long-term trajectory prediction of dynamic obstacles
The state of static obstacle
The location of autonomous vehicle
Short-term trajectory prediction of dynamic obstacles
METHOD
Long-term trajectory planner based on 4D Spatio-temporal map
OUTPUT
Initial maneuver
Real local trajectory
Short-term trajectory optimizer based on NMPC

The remainder of this paper is organized as follows. Obstacles and the ego vehicle are graphically described on spatio-temporal map in Section 2. By combining safe and efficient evaluation functions, a long-term trajectory planner based on improved A-star is specified in Section 3. A short-term trajectory planner based on NMPC to smooth or adjust initial trajectory is presented in Section 4. In Section 5, the performance of the proposed method is discussed by simulating in four common dynamic parking scenarios and comparing with spline local optimization. In the end, we close the paper with conclusion in Section 6.

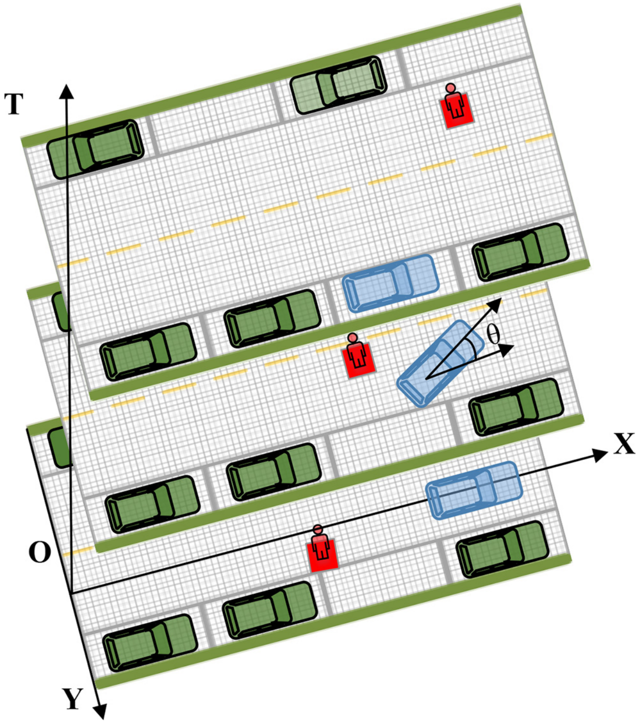
2. 4D Spatio-temporal map and vehicles model

There are many kinds of maps used to represent the driving environment of automobiles, such as topological map, geometric feature map and grid map. Among them, the grid map is widely used in navigation and the obstacle avoidance tasks. This description method uses grids to divide the planning space and increases the value of a specific grid to describe the threats posed by obstacles. Like the X-Y plane coordinate system in Figure 2, the 2D spatial grid map was set up by dividing driving environment into a finite number of identical squares. On the grid map, the resolution information is determined by the size of the square, and the position information is provided by the geometric center coordinates of each grid.

Assuming that the static obstacles' position is known before parking and that the trajectories of dynamic obstacles can be obtained by prediction module. The key of creating X-Y-T spatio-temporal map is to expand 2D spatial grid map along the time axis to describe the position and motion of the ego vehicle and obstacles (Xin, 2021). For example, the trajectories of static barriers, dynamic obstacles and the ego vehicle are represented by the green, red and blue parts, correspondingly in Figure 2.

Undeniably, the ego vehicle is not a particle. In order to receive a detailed trajectory, it is necessary to further depict the vehicle position. On the basis of the vehicle

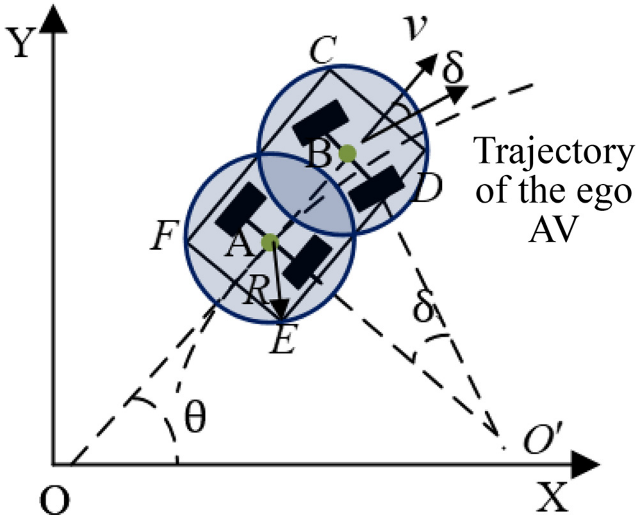
Figure 2 4D Spatio-temporal map



dimensions and Ackerman steering constraints, the rectangular vehicle CEDF can be equivalent to line segment AB (Li, 2021), as presented in Figure 3, where, v is the speed of the ego AV, δ is its front wheel angle and θ is its heading angle.

The blue part in Figure 3 is safe equivalent circles of automobile, which is used to describe the ego AV's shape with radius R. With the mid-point of rear wheel axis (x, y) as reference point, the relationship between the bounding box and safe equivalent circles can be expressed by the following equations:

Figure 3 Vehicle model



$$\begin{cases} R = \sqrt{((L + lf + lr)/4)^2 + (W/2)^2} \\ x_f = x + (0.75(L + lf + lr) - lr) \cdot \cos\theta \\ y_f = y + (0.75(L + lf + lr) - lr) \cdot \sin\theta \\ x_r = x + (0.25(L + lf + lr) - lr) \cdot \cos\theta \\ y_r = y + (0.25(L + lf + lr) - lr) \cdot \sin\theta \\ x_C = x + (L + lf) \cdot \cos\theta - (W/2) \cdot \sin\theta \\ y_C = y + (L + lf) \cdot \sin\theta + (W/2) \cdot \cos\theta \\ x_D = x + (L + lf) \cdot \cos\theta + (W/2) \cdot \sin\theta \\ y_D = y + (L + lf) \cdot \sin\theta - (W/2) \cdot \cos\theta \\ x_E = x\text{-}lr \cdot \cos\theta + (W/2) \cdot \sin\theta \\ y_E = y - lr \cdot \sin\theta - (W/2) \cdot \cos\theta \\ x_F = x\text{-}lr \cdot \cos\theta - (W/2) \cdot \sin\theta \\ y_F = y - lr \cdot \sin\theta + (W/2) \cdot \cos\theta \end{cases} \tag{1}$$

where x_f, y_f, x_r, y_r, x_C, y_C, x_D, y_D, x_E, y_E, x_F, y_F are coordinates of the line segment points A, B and the four corner points C, D, E, F along X- and Y-axes. L and W are the ego vehicle's wheelbase and width. lf and lr denote the length of its front overhang and rear overhang, respectively.

In X-Y coordinates, the ego AV moves continuously under vehicle kinematic constraints. Combined with Newton's second law, acceleration and angular velocity of the front wheel [a, w_δ] are set as control variables $U(t)$, while [x, y, θ, v, δ] are defined as state variables $X(t)$. Without regard to the impact of lateral acceleration or air resistance at low speed, kinematic model of a front-steering automobile can be developed through differential equation:

$$\dot{X}(t) = f(t, X(t), U(t)) = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{v} \\ \dot{\delta} \end{bmatrix} = \begin{bmatrix} v \cdot \cos\theta \\ v \cdot \sin\theta \\ v\tan\delta/L \\ a \\ w_\delta \end{bmatrix} \tag{2}$$

where both the state variables $X(t)$ and the control variables $U(t)$ are functions of time. Therefore, the ego AV's trajectory is composed of four dimensions: coordinates (x, y), heading θ and time t.

3. Long-term maneuver generation

After creating the 4D spatio-temporal map, we proposed a long-term trajectory planning with the aim of searching for optimal maneuvers that determine whether the speed or steering angle of the ego AV should be adjusted. Compared with other search algorithms, the A-star algorithm is extended based on the idea of Dijkstra algorithm, and is popular in path planning for its heuristic search strategy and higher path search (Bai, 2020). However, in practical applications, search algorithms need to consider not only the calculation workload and accuracy, but also the feasibility and efficiency of planned trajectories. Consequently, this section mainly focuses on improving the A-star algorithm in terms of the configuration of expansion nodes and heuristic functions.

3.1 Child node configuration

The traditional A-star algorithm expands eight nodes or six nodes on 2D grid maps around the parent node. Given the ego AV's position [$x(t_i)$, $y(t_i)$, $\theta(t_i)$] at time t_i, if directly searching for trajectory on 4D spatio-temporal map mentioned above, problems including high dimensionality, excess child nodes, infeasible nodes and discontinuous nodes will appear.

Note that, set the velocity of front-wheel steer angle w_δ and the acceleration a as the action space, and take the discrete time Δt as a step size to configure the extended nodes of current grid as follows:

$$\begin{cases} t_{i+1} = t_i + \Delta t \\ v(t_{i+1}) = v(t_i) + a \cdot \Delta t \\ \delta(t_{i+1}) = \delta(t_i) + w_\delta \cdot \Delta t \\ \theta(t_{i+1}) = \theta(t_i) + v(t_{i+1}) \cdot \tan\left(\delta(t_{i+1})\right) \cdot \Delta t / L \\ x(t_{i+1}) = x(t_i) + v(t_{i+1}) \cdot \cos\left(\theta(t_{i+1})\right) \cdot \Delta t \\ y(t_{i+1}) = y(t_i) + v(t_{i+1}) \cdot \sin\left(\theta(t_{i+1})\right) \cdot \Delta t \\ v_{\min} \leq v(t_{i+1}) <= v_{\max} \\ \delta_{\min} \leq \delta(t_{i+1}) <= \delta_{\max} \end{cases} \tag{3}$$

where [$v_{\min}$, $v_{\max}$] and [$\delta_{\min}$, $\delta_{\max}$] are allowable range of vehicle velocity and the front-wheel steer angle, respectively.

Controlling the child nodes position based on equation (3) not only denotes the vehicle kinematic characteristics, but also reduces the expansion range and dimension. Besides, it also widens the requirements for the child nodes position to allow child nodes to be free from the grid center.

3.2 Evaluation function

The evaluation function of the A-star algorithm is composed of two parts (Bai, 2020). One is the accumulated cost that caused by moving from the initial point to the current node, namely, the weighted sum of three terms: length cost S, safety cost APF and comfort cost C. The other is the estimated cost of moving to parking slot, namely, the heuristic cost η. Here, we take the nth child node as an example, wherein, $n \in N^*$ & $n <= 7$, and the overall cost function F can be defined as follows:

$$F(n, t_i) = w_S S(n, t_i) + w_{APF} APF(n, t_i) + w_c C(n, t_i) + w_\eta \eta(n, t_i) \tag{4}$$

where w_S, w_η, w_C, w_{APF} are the weight coefficients corresponding to four indicators.

To avoid unnecessary detours, the length cost S represents the shortest distance in travelling. At time t_i, this cost of child node n is easy to be measured by Euclidean distance:

$$S(n, t_i) = \|x(n, t_i) - x(t_i), y(n, t_i) - y(t_i)\|_2 \tag{5}$$

where $x(t_i)$, $y(t_i)$, $x(n, t_i)$, $y(n, t_i)$ are the position information of current node and its nth child node.

Because most dynamic obstacles in parking scenarios are vulnerable road users, the ego vehicle should not only search for the shortest path but also need to consider the distance from these obstacles. The safety cost APF is virtual potential field force that guides the ego AV away from obstacles. For the same dynamic obstacle may has different energies under different circumstances, in the light of the potential field approach, taking the information such as the moving direction, velocity of dynamic obstacles into account (Want *et al.*, 2021), the driving safety field $P_j(x, y)$ of dynamic obstacle j can be established as:

$$\begin{cases} m_j = \left(x - x^j_{obs}, y - y^j_{obs}\right); j > 0, j \in N \\ U_j(x,y) = \left(-\dfrac{\left(x - x^j_{obs}\right)^2}{2K_j^2\left(v_{xj}^2 + \sigma_o\right)} - \dfrac{\left(y - y^j_{obs}\right)^2}{2K_j^2\left(v_{xj}^2 + \sigma_o\right)} \right) \\ P_j(x,y) = M_j U_j(x,y) \exp(\cos\theta_j) m_j / |m_j| \end{cases} \tag{6}$$

where, x^{j}_{obs}, y^{j}_{obs}, v_{xj}, v_{yj} are the position and velocity information of dynamic obstacle j; σ_o is an allowable distance between the ego vehicle and obstacles. K_j and M_j reflect the dynamic obstacle's size and type.

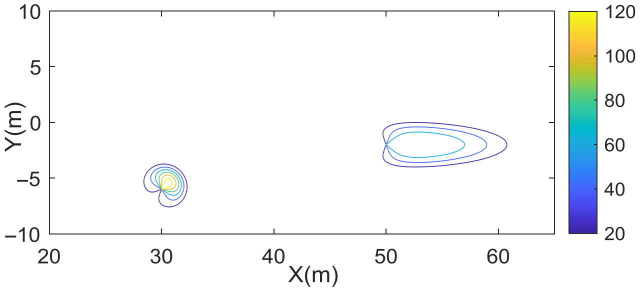
If there are two dynamic obstacles, one is located at (30 m, −6 m) with M_1 = 50, v_{x1} = 1 m/s, v_{y1} = 1 m/s, K_1 = 1. The other is at position (50 m, −2 m) with M_2 = 28, v_{x1} = 5 m/s, v_{y1} = 0 m/s, K_1 = 2. When σ_o = 1 m, the risk field can be assessed by Figure 4. Around the obstacle, the closer the ego AV gets to the obstacle's velocity direction and position, the more dangerous it will be.

Regarding the stationary state of the ego AV as a safe state, this article uses the driving safety field of dynamic obstacles to improve the safety cost function APF by:

$$\begin{cases} R^j_o = R + R^j_{obs} \\ sf(n, t_i) = \left(x_f(n, t_i) - x^j_{obs}(t_i + \Delta t), y_f(n, t_i) - y^j_{obs}(t_i + \Delta t)\right) \\ sr(n, t_i) = \left(x_f(n, t_i) - x^j_{obs}(t_i + \Delta t), y_f(n, t_i) - y^j_{obs}(t_i + \Delta t)\right) \\ dis(n, t_i) = \min\left(\|sf(n, t_i)\|_2, \|sr(n, t_i)\|_2\right) - R^j_o \\ APF(n, t_i) = \begin{cases} 0 \quad \text{if } dis(n, t_i) \geq 10 || v(n, t_i) = 0 \\ \sum_{j=0}^{NJ} P_j(x(n, t_i), y(n, t_i)) \text{ else} \end{cases} \\ 1 \leq n \leq 7, n \in N*; NJ \in N; j \geq 0, j \in N \end{cases} \tag{7}$$

where $\|sf(n, t_i)\|_2$ and $\|sr(n, t_i)\|_2$ denote the distances between obstacles and two safe equivalent circles, respectively. $dis(n, t_i)$ is the shortest one of them. R^j_{obs} is the radius of jth dynamic obstacle's equivalent circle. NJ is the number of dynamic obstacles. $t_i + \Delta t$ is the time of extended child nodes, when the parent node is at time t_i.

Figure 4 Risk assessment of dynamic obstacles



The comfort cost C plays an important role for the ego AV to avoid starting, stopping, braking and pivoting frequently. Given the velocity and steering angle of the parent node, the comfort loss can be shown as below:

$$C(n, t_i) = |v(n, t_i) - v(t_i)| + |\delta(n, t_i) - \delta(t_i)| \tag{8}$$

In addition, in early studies (Bai, 2020; Bui and Häggström, 2019), the Euclidean distance was commonly used as a heuristic function, which is suitable for robots to search for the shortest path, but fails to meet requirements of the automobile's shape and targeted posture. Recently, many scholars have combined the vehicle's non-holonomic and Reeds-Shepp (RS) path planning method (Reeds and Shepp, 1990) to improve the heuristic function for continuous-state data (Xie, 2020). By equating the ego vehicle to an oriented particle and connecting arcs and line segments, RS path planning method can quickly generate smooth and shortest path that meets maximum curvature constraint, the start and end positions constraints and heading attitude constraints. However, the evaluation based on the shortest path still fails to guarantee the parking efficiency.

In this work, combining with forward velocity integration method (Kapania *et al.*, 2016), we take the equidistantly sampled RS path as input, and propose an RS trajectory estimation method as equation (9) to estimate the passing time t_{rs} of each point in the RS path:

$$\begin{cases} v_{rs+1} = \begin{cases} \min\left(v_{\max}, \sqrt{2a_{\max}(S_{rs+1} - S_{rs}) + {v_{rs}}^2}\right) \text{accelerate} \\ \max\left(v_{\min}, \sqrt{2a_{\min}(S_{rs+1} - S_{rs}) + {v_{rs}}^2}\right) \text{decelerate} \\ v_{rs} \qquad \text{else} \end{cases} \\ t_{rs+1} = 2S_{rs+1}/(v_{rs} + v_{rs+1}) + t_{rs} \end{cases} \tag{9}$$

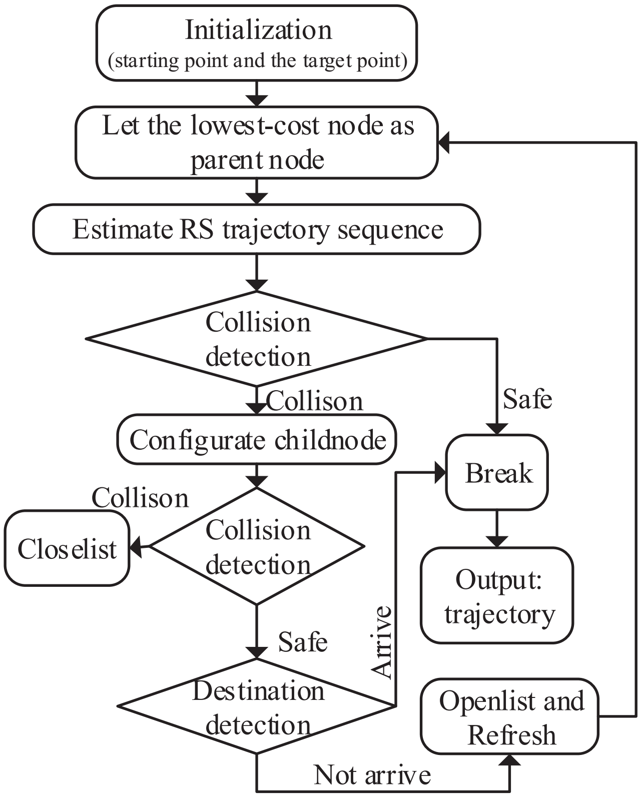
where rs is the number of sampling points in the RS path, $0 \le rs \le NRS - 1$. NRS is the target point's sequence number. $\{S_0, \ldots, S_{NRS}\}$ is a set of distances from the starting point to the sampling points. t_{rs} and t_{rs+1} are the passing time of sampling points rs and $rs + 1$, respectively.

Furthermore, the heuristic cost η can be regarded as the time required for the child node n at time t_i to reach the target point along the RS path by:

$$\eta(n, t_i) = t_{NRS}(n, t_i) \tag{10}$$

Figure 5 indicates the search flow chart of our improved A-star algorithm. Holding the evaluation function [equations (4)–(9)], in the search process, we put the occupied child nodes into the Closelist, let the child node with the lowest cost of the Openlist be the parent node. As the traditional A-star algorithm only concludes the search process after the target point has been found, which may lead to more computation. To simplify the total search process, if current cost is the least and current RS trajectory does not collide with the surrounding static or dynamic obstacles, we will use the sampled RS trajectory as the future trajectory, and terminate the search. In addition, we take the output points in X-Y-θ-T map as an initial trajectory for the ego automatic vehicle, and denote them as $X_r = [x(t), y(t), \theta(t), v(t), \delta(t)]$.

Figure 5 Flowchart of X-Y-θ-T 4D A-star algorithm



4. Short-term trajectory planning

The initial trajectory generated by the proposed long-term trajectory planner is a series of nodes at different times with coarse path information and rough velocity profile. In this case, tracking directly will result in great errors and even failure. Besides, the planning frequency of initial trajectory is so low that there are lots of uncertain prediction results of the surrounding obstacles. Hence, in this section, a multi-objective optimization technique, namely, the NMPC method (Taherian, 2021) is introduced to obtain a smoother and safer initial trajectory.

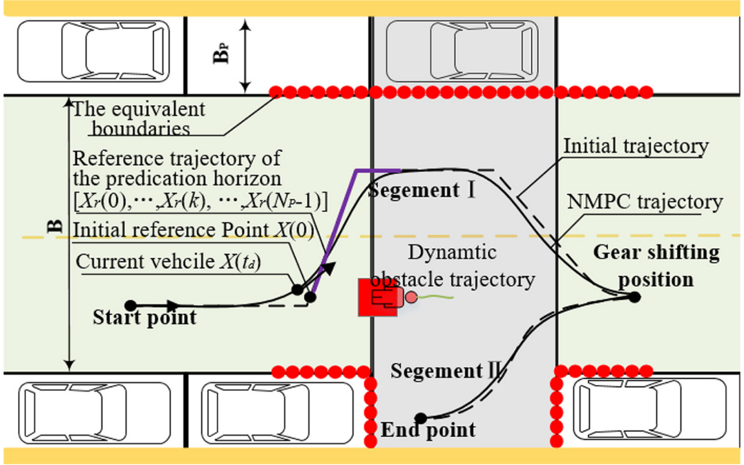
4.1 NMPC optimization method

The NMPC method consists of three important parts: discretization, prediction and rolling optimization. The key of this method is to transform time continuous optimal control problems of rolling horizons into structured nonlinear programs via discretization.

As presented in Figure 6, the dashed line is the reference trajectory, which is calculated using the improved A-star algorithm. The trajectory optimized by NMPC method is shown by the black bold line. In view of the multi-step parking phenomenon that may occur to the parking process, making the forward and backward reference trajectories separate at the gear shifting position.

In every path, choose the point that is the nearest to the actual position as the initial reference point. If there is more than one nearest point, use the one that is the closest in time as initial reference point. Supposing t_d is the current time instant, N_P is the number of look ahead prediction horizon steps.

Figure 6 Diagram of NMPC


B_P
The equivalent boundaries
Reference trajectory of the predication horizon $[X_r(0),\cdots,X_r(k),\cdots,X_r(N_P\text{-}1)]$
B
Initial reference Point $X(0)$
Current vehcile $X(t_d)$
Start point
Segement Ⅰ
Initial trajectory
NMPC trajectory
Dynamtic obstacle trajectory
Gear shifting position
Segement Ⅱ
End point

Such as the purple part in Figure 6, the state sequence of prediction horizon $[X_r(0),\ldots, X_r(k),\ldots, X_r\,(N_P-1)]$ shifts forward at every time step T, wherein $k\in[0,\ N_P-1]$. Set $\overrightarrow{X}_{td}$ as the start point of NMPC and ensuring the start-point state of each sub-problem to be consistent with the solution of the previous. The local trajectory in each rolling window is then generated by solving the following optimization problem, which contains cost function $\mathcal{J}(t,\ X(t),\ U(t))$ and constraints, but only the first states and control inputs are applied to the system:

$$\left.\begin{aligned} &\min \mathcal{J}(t,X(t),U(t))\\ &s.t.\,\dot{X}(t)=f_N(t,X(t),U(t))\\ &\quad X(t_d)=\overrightarrow{X}_{t_d}\\ &\quad h(X(t),U(t))\le 0\end{aligned}\right\} k\in[0,N_P-1] \tag{11}$$

where $f_N(t,\ X(t),\ U(t))$ is the system differential equation. $h(X(t),\ U(t))$ is the convex constraint on this issue.

Because the vehicle differential equation [equation (2)] just ensures the front wheel angle profile and velocity profile continuous, there still will be some cusps in those profiles. For this reason, we set their second derivative $j_a=\ddot{v},\ j_\delta=\ddot{\delta}$ as the control variables, and define $[x,\ y,\ \theta,\ v,\ \delta,\ a,\ w_\delta]$ as the state variables of NMPC.

4.2 Cost function

From global optimality, our first targeted task of NMPC optimization is to reflect the advantage of the reference trajectory and seize the best time to accelerate or decelerate. To make the initial trajectory smooth and traceable, the states and control inputs errors between of the reference trajectory and local trajectory are taken as a penalty term via integral time t_P:

$$\mathcal{J}_F=\sum_{k=0}^{N_P-1}\|X(t_d+kT)-X_r(k)\|_Q^2+\|U(t_d+kT)\|_R^2 \tag{12}$$

where Q and R are the cost weighting matrices for the process states and the control inputs, respectively.

In practice, the condition of the obstacles may change at any time while driving on the road. Because of the precision of the predictor, a disparity between the projected and actual scenario is unavoidable, and it will grow larger as the prediction horizon lengthens. This indicates that the initial solution found through long-term trajectory planning cannot completely guarantee the vehicle's safety. Meanwhile, if local planning fails to predict the movement states of the obstacle in real time, and continues to solve the problem based on the initial state, parking accidents will occur. For that, holding the potential field function [equation (6)] and real-time predicted trajectories of obstacles, the objective function of the collision avoidance can be described as follows:

$$\mathcal{J}_O=w_O\sum_{k=0}^{N_P-1}\left(10/\exp(1+10\cdot dis(t_d+kT))\right) \tag{13}$$

where $dis(t_d+kT)$ is the shortest distance between obstacles and the ego vehicle in $k\cdot T$ seconds. w_O is weight of this objective.

Though, adjusting local trajectory ensures the safety of the vehicle, it also brings problem that the target is unreachable. To raise the success rate of parking in a dynamic environment, the terminal objective function $\mathcal{J}_E$ penalizes the deviation of the terminal states from the reference trajectory by:

$$\mathcal{J}_E=\|X(t_d+(N_P-1)T)-X_r(N_P-1)\|_E^2 \tag{14}$$

where E is a diagonal weighting matrix for the terminal states.

4.3 Constraints

Parking is a complex operation, which is integrated by steering control and speed control. In fact, in the process of parking, the speed is usually low and the steering angle is mostly at AVs' limit. With reference to the above conditions, the state variables and control variables are constrained to meet the physical requirements of the ego AV as:

$$\begin{bmatrix} v_{\min}\\ \delta_{\min}\\ a_{\min}\\ w_{\delta\min}\end{bmatrix}\le\begin{bmatrix} v(t)\\ \delta(t)\\ a(t)\\ w_\delta(t)\end{bmatrix}\le\begin{bmatrix} v_{\max}\\ \delta_{\max}\\ a_{\max}\\ w_{\delta\max}\end{bmatrix}\quad \begin{aligned}&k\in[0,N_P-1]\\ &t=t_d+kT\end{aligned} \tag{15}$$

where $v_{\min}$, $v_{\max}$, $\delta_{\min}$, $\delta_{\max}$ are the limited speed and front-wheel steering angle allowed in this case.

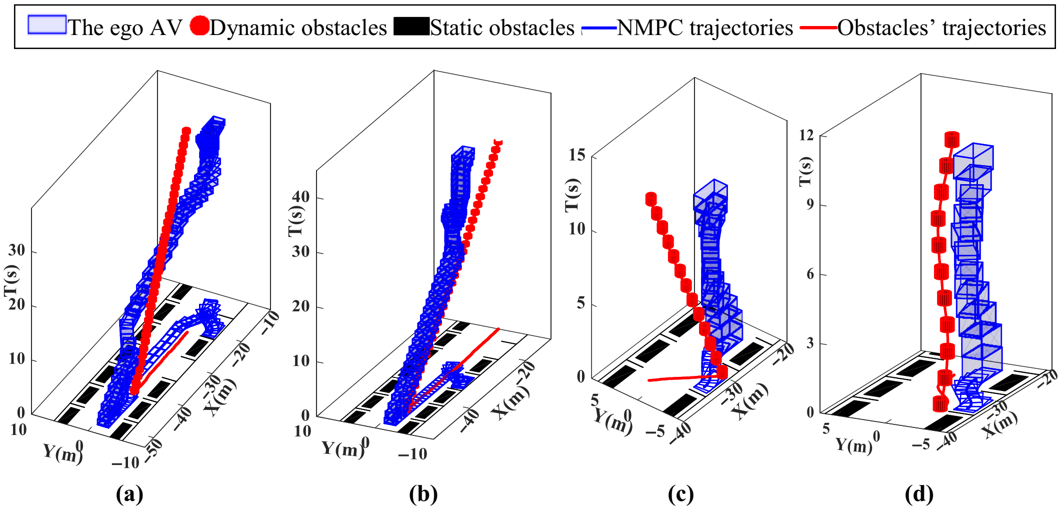
Near the parking slot, the direction of the vehicle needs to be adjusted continuously and the width of road changes suddenly. For reducing the search time of long-term trajectory planning and simplifying the constraints of short-term trajectory planning, we used the regional partition method to divide the road map into the channel region S and the complex region *D*, corresponding to the green and gray parts in Figure 6.

Given the location relationship between the prediction area *P* where the state sequence of reference trajectory $[X_r(0),\ldots, X_r(k),\ldots, X_r\ (N_P-1)]$ is located and the complex region *D* as the critical condition. If $P \cap D = \varnothing$, it is easy to express the corridor boundary constraints by limiting the ego vehicle's four corner points CDEF:

$$\begin{bmatrix} -B/2 \\ -B/2 \\ -B/2 \\ -B/2 \end{bmatrix} \leq \begin{bmatrix} y_C(t) \\ y_D(t) \\ y_E(t) \\ y_F(t) \end{bmatrix} \leq \begin{bmatrix} B/2 \\ B/2 \\ B/2 \\ B/2 \end{bmatrix} \tag{16}$$

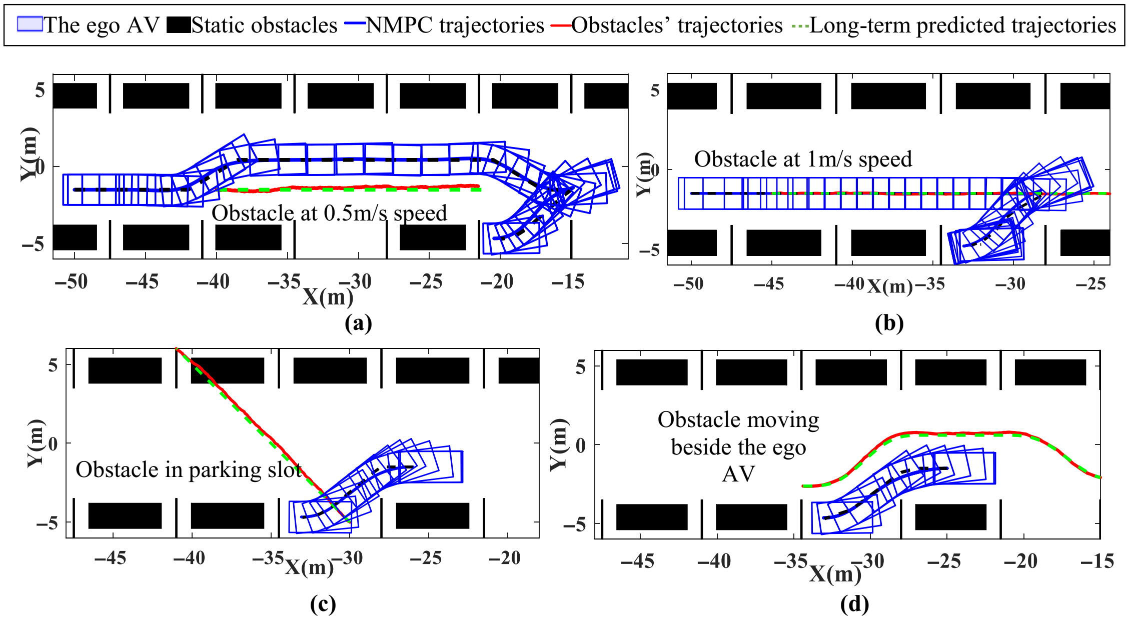
where $t = t_d + kT$, $k \in [0, N_P - 1]$, B is the width of road in the channel region.

Figure 7 Spatio-temporal trajectory



Notes: (a) Scene 1; (b) scene 2; (c) scene 3; (d) scene 4

Figure 8 Paths planned by our method



Notes: (a) Scene 1; (b) scene 2; (c) scene 3; (d) scene 4

When $P \cap D = \varnothing$, the complex road boundary is equivalent to static obstacles for processing, such as these red points in Figure 6. Here, the road boundaries are specifically constructed as:

$$\begin{cases} \begin{bmatrix} -B/2-B_P \\ -B/2-B_P \\ -B/2-B_P \\ -B/2-B_P \end{bmatrix} \le \begin{bmatrix} y_C(t) \\ y_D(t) \\ y_E(t) \\ y_F(t) \end{bmatrix} \le \begin{bmatrix} B/2+B_P \\ B/2+B_P \\ B/2+B_P \\ B/2+B_P \end{bmatrix} & \begin{matrix} t = t_d + kT \\ k \in [0, N_P - 1] \end{matrix} \\ \begin{bmatrix} \|x_f(t)-x_o(m), y_f(t)-y_o(m)\|_2 \\ \|x_r(t)-x_o(m), y_r(t)-y_o(m)\|_2 \end{bmatrix} >= \begin{bmatrix} R+r_o(m) \\ R+r_o(m) \end{bmatrix} & \end{cases} \tag{17}$$

where B_P is the width of parking slot, m is the number of static obstacles, and $x_o(m)$, $y_o(m)$, $r_o(m)$ represent the position and size of the mth static obstacle.

Similarly, the collision avoidance constraints of dynamic obstacle in the prediction horizon are given as:

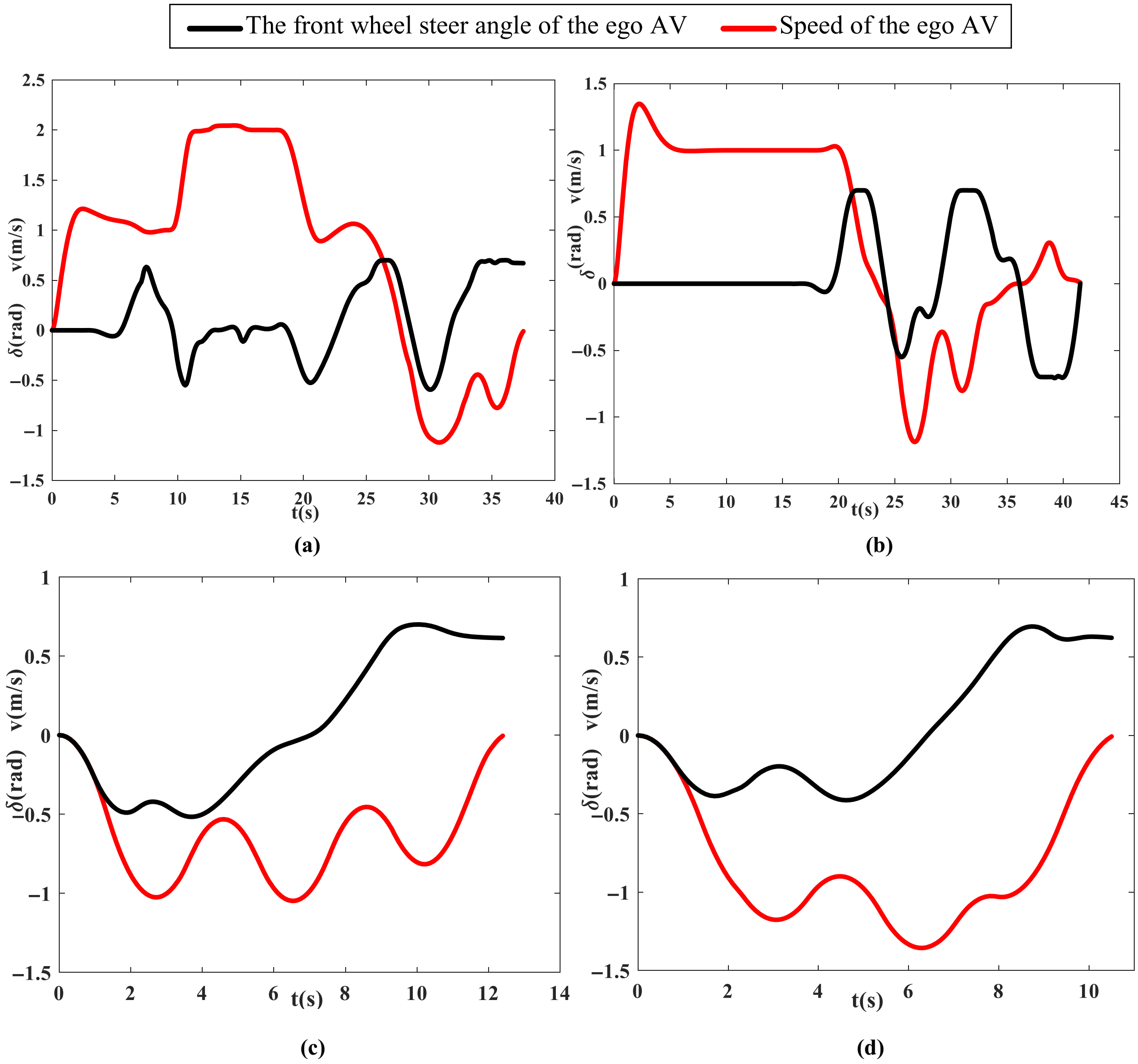
$$\begin{bmatrix} \|x_f(t)-x^j_{obs}(t), y_f(t)-y^j_{obs}(t)\|_2 \\ \|x_r(t)-x^j_{obs}(t), y_r(t)-y^j_{obs}(t)\|_2 \end{bmatrix} >= \begin{bmatrix} R+R^j_{obs} \\ R+R^j_{obs} \end{bmatrix}$$
$$k \in [0, N_P - 1], t = t_d + kT \tag{18}$$

5. Numerical simulation

5.1 Simulations in four common scenes

In long-term trajectory planning, the 4D spatio-temporal map with discrete resolution of 0.1 is used to improve A-star algorithm and search for optimal maneuver. Meanwhile, the ACADO Toolkit is applied to transform the NMPC optimization problem [equation (11)] into a small and sparse

Figure 9 Control profiles planned by our method



Notes: (a) Scene 1; (b) scene 2; (c) scene 3; (d) scene 4 quadratic program by using multiple shooting discretization, Real-time iterations scheme and Gaussian–Newton, as well as efficient C programming language (Rathai, 2019). Presuming that the prediction information is known, simulations are presented to verify the effectiveness of the proposed method in this section.

In this paper, the proposed method is tested in four common dynamic parking scenes from daily life. Scene 1 and Scene 2 focus on the situation where automatic vehicle drives to the parking spot with obstacles of different speeds. Scene 3, Scene 4 are set with obstacles of circuitous and straight path in AV's backward direction. Combined with QPOASES solver, the short-term planning runs every 0.1 s and derives the simulation results as follows:

The planned spatio-temporal trajectory is presented in Figure 7, where the blue lines are planned trajectories by NMPC optimizer, while the red lines represent obstacles' trajectory. It is demonstrated that the vehicle can complete the parking task quickly and safely by overtaking, waiting, speed regulation or other schemes automatically itself when facing obstacles in different situations. And there is no need for an additional decision module because the efficiency and safety can be balanced in searching.

In Figures 8 and 9, it can be seen that the planned path, speed and front-wheel steering angle profiles of these scenarios are safe (without collision), continuous, smooth and feasible (within the range of 2 m/s and 0.7 rad), which meets the physical performance constraints demands.

From computing power, the calculation time of the local planning mentioned above is recorded. As described in Figure 10, based on ACADO and OASEQP, the required time for the calculation in every rolling window is within 7 ms in most cases. Besides, the longest calculation time is 15 ms, which is much less than the sampling time of 100 ms and satisfies the real-time requirement of calculation.

5.2 Comparison with spline optimization method

Taking Scenario 4 as an example, the spline planning used in the study of Zhou *et al.* (2020a, 2020b) is compared with NMPC put forward in this paper. Constrained by static obstacles avoidance target and minimum curvature limitation, the path and speed profiles planned by spline method are shown in Figure 11.

As shown in Figure 11(a) and (b), the results of our method are similar to the spline results, both of which have high smoothness. Meanwhile, the path curvature is less than $0.34\,\mathrm{m}^{-1}$, which meets the maximum front wheel angle requirements. However, as we all know that using fmincon solver to acquire the spline coefficients of a nonlinear optimization problem takes a long time.

In the case of the control profiles in Figure 11(c), the spline method is applied to solve the speed profile without collision in a S-T map. On the whole, both planned velocity profiles are smoother. In view of safety, the velocity result of our method reaches its maximum value during the departure of the obstacle, while the result of the spline method reaches the maximum value of 2 m/s when the obstacle is approaching,

Figure 10 Computation performance

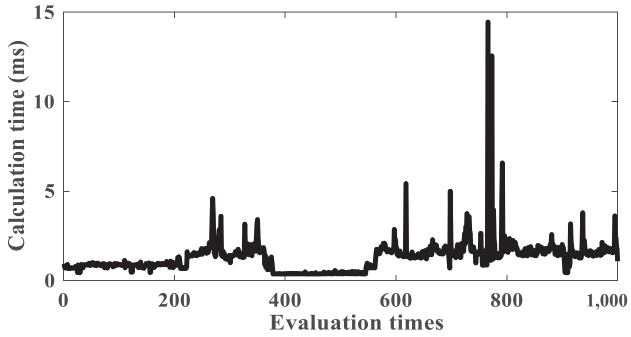
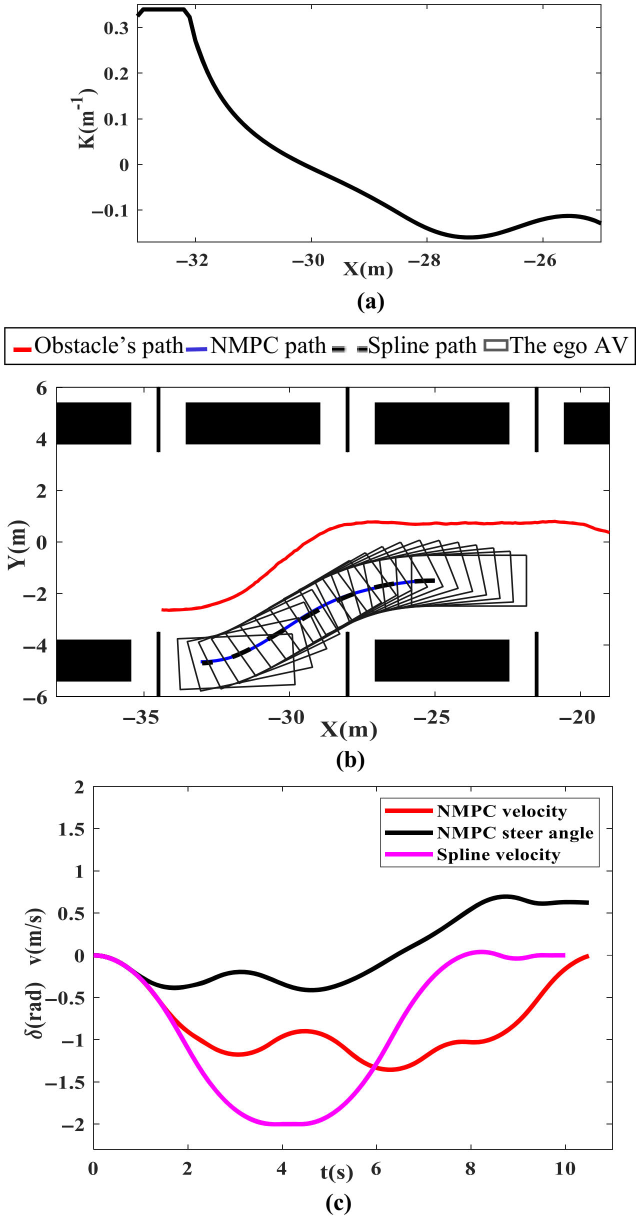


Figure 11 Trajectories planned by different methods



Notes: (a) the path curvature planned by spline method; (b) paths planned by different methods; (c) control profiles which is more dangerous. Furthermore, by contrast, NMPC optimizer can coordinate the ego AV's speed and its front-wheel steering angle simultaneously, which is in accord with the parking and collision avoidance demands, acquires more detailed trajectory's information and simplifies the following tracking and control tasks.

6. Conclusion

To summarize, the proposed trajectory planning frame based on spatio-temporal heuristic method used for automatic parking to avoid obstacles has the following advantages:

- In terms of long-term planning, a 4D spatio-temporal map, vehicle kinematics constraints and RS curve are used to improve A-star algorithm with the shortest time as the heuristic function. It not only finds the initial solution of parking trajectory but also resolves a contradiction between the parking maneuver and the obstacle avoidance maneuver in dynamic scenes.
- In terms of short-term planning, a NMPC trajectory optimizer has been introduced based on vehicle kinematics constraints, minimum deviation target and the obstacle avoidance target. Besides, applying ACADO Toolkit and QPOASES solver can make the trajectory feasible, safe and efficient.

This study provides an effective approach to trajectory planning and decision-making in dynamic environment, which can be extended to other complex applications, such as highways and unstructured roads.

However, as the parking process is short and the obstacles' long-term trajectories are pre-assigned instead of designing a prediction module, the planning frequency of the long-term planner is not set. To solve above limitations and improve the performance of automatic parking products, designing an obstacles' trajectories prediction module for parking like the previous (Gan, 2021) would be our future work.

References

Bai, S. (2020), "Research on obstacle avoidance method of indoor mobile robot based on trajectory prediction", Harbin Institute of Technology.

Bui, L. and Häggström, M. (2019), "Implementation of autonomous parking with two path planning algorithms".

Cheng, L., Liu, C. and Yan, B. (2014), "Improved hierarchical A-star algorithm for optimal parking path planning of the large parking lot", *2014 IEEE International Conference on Information and Automation (ICIA)*. IEEE, pp. 695-698.

Gen, N., Zhao, K., Pei, F., Guo, Q. and Liu, S. (2019), "Automatic parking path planning based on pseudo-spectral splicing method", *Journal of Chongqing University of Technology(Natural Science)*, Vol. 33 No. 4, pp. 26-31.

Gan, N., Jiang, Z., Zhou, B., Chai, T. and He, Z. (2021), "Research on vehicle trajectory prediction method for intersections without signal lights", *SAE International Journal of Connected and Automated Vehicles*, Vol. 4 No. 3, (12-04-03-0021).

Han, L., Do, Q.H. and Mita, S. (2011), "Unified path planner for parking an autonomous vehicle based on RRT", *2011 IEEE International Conference on Robotics and Automation*, IEEE, pp. 5622-5627.

Hu, Y., He, P. and Liu, X. (2020), "Parallel parking control method based on model predictive control", *Journal of Chongqing University of Technology (Natural Science)*, Vol. 34 No. 10, pp. 1-8.

Jhang, J.H. and Lian, F.L. (2020), "An autonomous parking system of optimally integrating bidirectional rapidly-exploring random trees* and parking-oriented model predictive control", *IEEE Access*, Vol. 8 No. 99, pp. 1-1.

Kapania, N.R., Subosits, J. and Christian Gerdes, J. (2016), "A sequential two-step algorithm for fast generation of vehicle racing trajectories", *Journal of Dynamic Systems, Measurement, and Control*, Vol. 138 No. 9.

Khayyam, H., Javadi, B., Jalili, M. and Jazar, R.N. (2020), "Artificial intelligence and internet of things for autonomous vehicles", *Nonlinear Approaches in Engineering Applications*, Springer, Cham, pp. 39-68.

Li, X. (2021), "Research on automatic parking path planning and tracking control in narrow parking spaces", Dalian University of Technology.

Li, B., Wang, K. and Shao, Z. (2016), "Time-optimal maneuver planning in automatic parallel parking using a simultaneous dynamic optimization approach", *IEEE Transactions on Intelligent Transportation Systems*, Vol. 17 No. 11, pp. 3263-3274.

Li, Z., Ma, H., Zhang, X. and Fei, Q. (2019a), "Path planning of the dual-arm robot based on VT-RRT algorithm", *2019 Chinese Control Conference (CCC)*, IEEE, pp. 4359-4364.

Li, B., Acarman, T., Peng, X., Zhang, Y. and Kong, Q. (2019b), "Automatic parking path planning based on pseudo spectral splicing method", *Journal of Chongqing University of Technology (Natural Science)*, Vol. 33 No. 4, pp. 26-31.

Li, B., Acarman, T., Peng, X., Zhang, Y. and Kong, Q. (2020), "Maneuver planning for automatic parking with safe travel corridors: a numerical optimal control approach", *2020 European Control Conference (ECC)*, Saint Petersburg, pp. 1993-1998.

Nakrani, N.M. and Joshi, M.M. (2021), "A human-like decision intelligence for obstacle avoidance in autonomous vehicle parking", *Applied Intelligence*, Vol. 52 No. 4, pp. 1-20.

Qian, L., Wu, B., Qiu, D. and Hu, W. (2019), "Path planning for autonomous parallel parking based on piecewise gauss pseudospectral method", *Automotive Engineering*, Vol. 41 No. 12, pp. 1401-1409.

Qian, L., Wu, B., Qiu, D. and Hu, W. (2020), "Path planning of autonomous parking based on HP-adaptive pseudospectral method", *Journal of Mechanical Engineering*, Vol. 56 No. 4, pp. 125-134.

Rathai, K.M.M., Sename, O. and Alamir, M. (2019), "GPU-based parameterized NMPC scheme for control of half car vehicle with semi-active suspension system", *IEEE Control Systems Letters*, Vol. 3 No. 3, pp. 631-636.

Reeds, J. and Shepp, L. (1990), "Optimal paths for a car that goes both forwards and backwards", *Pacific Journal of Mathematics*, Vol. 145 No. 2, pp. 367-393.

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

Taherian, S., Halder, K., Dixit, S. and Fallah, S. (2021), "Autonomous collision avoidance using MPC with LQR-based weight transformation", *Sensors*, Vol. 21 No. 13, p. 4296.

Tang, W., Yang, M., Le, F., Yuan, W., Wang, B. and Wang, C. (2018), "Micro-vehicle-based automatic parking path planning", *2018 IEEE International Conference on Real-time Computing and Robotics (RCAR)*, IEEE, pp. 160-165.

Wang, M., Wang, Z. and Zhang, L. (2021), "Local path planning for intelligent vehicles based on collision risk evaluation", *Journal of Mechanical Engineering*, Vol. 57 No. 10, pp. 28-41.

Xie, F. (2020), "Research on key algorithms of autonomous parking in dynamic and complex scenarios", Jilin University.

Xin, L., Kong, Y., Li, S.E. and Chen, J. (2021), "Enable faster and smoother spatio-temporal trajectory planning for autonomous vehicles in constrained dynamic environment", *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, Vol. 235 No. 4, pp. 1101-1112.

Zhang, Z. and Zhao, Z. (2014), "A multiple mobile robots path planning algorithm based on A-star and Dijkstra algorithm", *International Journal of Smart Home*, Vol. 8 No. 3, pp. 75-86.

Zhou, R.F., Liu, X.F. and Cai, G.P. (2020b), "A new geometry-based secondary path planning for automatic parking", *International Journal of Advanced Robotic Systems*, Vol. 17 No. 3, pp. 1729881420930575.

Zhou, J., He, R., Wang, Y., Jiang, S. and Luo, Q. (2020a), "Autonomous driving trajectory optimization with dual-loop iterative anchoring path smoothing and piecewise-jerk speed optimization", *IEEE Robotics and Automation Letters*, Vol. 6 No. 2.

Corresponding author

Bing Zhou can be contacted at: zhou_bingo@163.com