IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 3 January 2023, accepted 19 January 2023, date of publication 23 January 2023, date of current version 27 January 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3239079

RESEARCH ARTICLE

Investigations on Speed Planning Algorithm and Trajectory Tracking Control of Intersection Scenarios Without Traffic Signs

WEI LIANG AND HAITAO XING

School of Mechanical and Automotive Engineering, Shanghai University of Engineering Science, Shanghai 201620, China

Corresponding author: Wei Liang (wei_liang222@126.com)

This work was supported by the National Natural Science Foundation of China under Grant 51505274.

ABSTRACT Efficient intersection planning is one of the most challenging tasks for an autonomous vehicle at present. The degree of response to other traffic participants and whether the path tracking effect is good are the key aspects that determine the performance of planning algorithm for an autonomous vehicle. In order to deal with these problems, in this paper, the frame structure of POMDP (Partially Observable Markov Decision Processes) and the time distance coordinate system are combined to carry out the speed planning for the self-propelled vehicle. The speed planning designs a novel POMDP model to ensure effective prediction of behavioral intentions and decision-making at intersections with no traffic signs. Specifically, the method is considered both the driving position and the time range of arrival of the vehicle at the intersection in real-world situations. At the same time, the algorithm is projected the state information into the time distance coordinate system in the form of probability, and used the value iteration method to solve the optimal solution, so as to improve the adaptability of the algorithm to real-world scenarios. Furthermore, MPC control is added on this basis. Rational use of dynamic MPC and kinematic MPC can effectively reduce the tracking error, improve the stability of tracking, and carry out finer control when the speed frequently fluctuates.

INDEX TERMS POMDP, MPC, distance time coordinate system.

I. INTRODUCTION

In the past few years, with the development of the automobile industry, more and more cars were equipped with the function of autonomous driving, a large number of algorithms were used to solve the problem of automobile path planning and tracking decision. However, most of the vehicles provided with autonomous driving function were running in some fixed scenes or some structured scenes. For complex scenes, the voice prompted the procedure to exit and require manual takeover. In the whole process from the driver entering the vehicle at the starting point to drive and leaving the vehicle at the destination, it was rare that the vehicle can fully drive autonomously. One of the key difficulties lied in the terrible traffic conditions, such as intersections without traffic signs.

The scene of intersection is one of the most commonplace scenes for vehicles on the road, and its complexity and dangerous degree also vary widely, requiring avoidance of pedestrians, vehicles and other static obstacles [1]. The avoidance of these objects is directly related to local path planning. Generally speaking, the decision and planning of local path are usually decouple into a position state and a velocity state. The position state is applied for the static obstacle, and the velocity state is for the dynamic obstacle. In the intersection scenario, it is also necessary to consider a tracking control state of the autonomous vehicle path, so that the vehicle can pass through this scenario safely and efficiently. Moreover, in the actual scene, due to noise, distance limitation and sensor occlusion, the perception of autonomous vehicles is uncertain. And considering the intentions of other traffic participants, their behavior cannot be accurately predicted.

Many researchers studied the intersection scenarios, such as the adaptive signal controlled scheme to reasonably

The associate editor coordinating the review of this manuscript and approving it for publication was Yang Tang.

allocate the green time and prevent vehicle congestion [2]. In order to help drivers pass the intersection safely, the pre-collision acceleration model of non-turning drivers in straight-line crossover path collision, the pre-collision acceleration model

of left-turning drivers in the left turn crossover path reverse direction collision and lateral collision were established [3]. Methods related to intersections are broadly divided into three categories.

1) The first category is non-learning based method. These schemes don't involve learning. For example, in the DARPA Urban Challenge, hierarchical state machines were applied to define the expected behavior of vehicles in the different scenarios, such as driving straight on urban roads, changing lanes, and turning left at intersections [4], [5]. However these methods ignore the uncertainty of the decision process. For the formation of the tracking path, a reaction-based trajectory generation method was proposed to avoid a collision on the road [6]. For the constraint of comfort, a new longitudinal driving strategy was proposed to carry out in a discrete planning, so that the driving vehicle can better recognize and respond obstacles [7]. But these planners tend to be too conservative. For the question of whether it can be stable during lane change, stable trajectory planning was achieved by establishing dynamic constraints on vehicles during lane change, with limiting time-dependent lane changing yaw velocity and lateral acceleration [8]. Nevertheless, surrounding vehicles were not considered.
2) The second category is supervised learning method. Such as, logistic regression model was used to deduce the probability of different intentions of other drivers, and a unified planning framework under uncertain probability was proposed to realize the decision of vehicles overtaking and yielding at intersections [9]. In addition to manually design driven decision, high performance driving strategies can also be obtained by collecting driving data and offline learning [10]. However, these methods are independent of the future movement of autonomous vehicles and lack of interactivity.
3) The third category is Reinforcement learning method. This can be divided into offline and online modes. Offline policies are powerful and can solve very complex situations: Mixed Observable Markov Decision Process (MOMDP) was used to describe the vehicle problem in the intersection scene as an intentions-aware planning decision problem, and the driving intentions of additional vehicles were considered in the discrete state space [11]. Considering that there may have obstruction in an uncertain environment, continuous POMDP was used to solve this problem, and a general method for decision-making in uncertain driving environment was proposed [12]. The disadvantage of these offline methods is that the diversity of scenarios and vehicle configurations can lead to policies that are unpredictable. Online approach requires trade-offs between solution quality and the complexity of problem formulation, depending on the size of the state space and planning horizon. For example, in order to adapt sensor noise in urban traffic, POMDP method was utilized to realize real-time decision when vehicle lane changes [13]. Because of the simplification of state space, this method can not be easily transferred to intersection planning. For the uncontrolled intersection scenario, the Hidden Markov Model (HMM) was used to model the behavior of other traffic participants, afterwards the POMDP decision-making process was established. The human-like strategy generation mechanism was used to generate candidate decisions to make them pass the intersection [14]. The intention estimation was assumed to be deterministic in the planning process. Therefore, no information-gathering operation was carried out. Aiming at the scene with complex pedestrians, an intention-aware online planning approach for autonomous driving amid many pedestrians was proposed, which can drive safely, efficiently and smoothly as near the pedestrians [15].

This paper studies the special movement scene of intersection scenarios without traffic signs. For such a low or middle speed scenario, the autonomous vehicle needs to predict the vehicle's behavior intention in order to prevent collisions, or to yield to other vehicles and pedestrians. In this scenario, in the absence of static obstacles, the moving destination target of both the ego-vehicle and the target vehicle is limited, and the corresponding path is restricted. No special planning is required for the position statement of the ego-vehicle. We focus on the planning of the speed and the tracking state of the ego-vehicle in the arc trajectory of intersection.

The contributions in this paper are two-folds:

1) POMDP and time distance coordinate system are used as the two methods for complementation. The advantage of POMDP method is a random process in accordance with the actual environment [16], [17], which can be an appropriate modeling calculation for complex scene, but it needs a lot of computing power. The method of time distance coordinates system is one of the methods based on the reaction formula, which has the advantages of simplicity, high efficiency and strong adaptability, but has the disadvantage of hysteresis. Furthermore, combining the characteristics of the two methods, we can achieve another speed planning method. This proposed method improves the interactivity and decision-making ability to the surrounding dynamic traffic participants while reducing the computational power. Under appropriate circumstances, it can also degenerate to a linear prediction.
2) For the position tracking control, the ego-vehicle itself is not stabilized when passing the arc trajectory. After adding the velocity planning to the ego-vehicle, it is

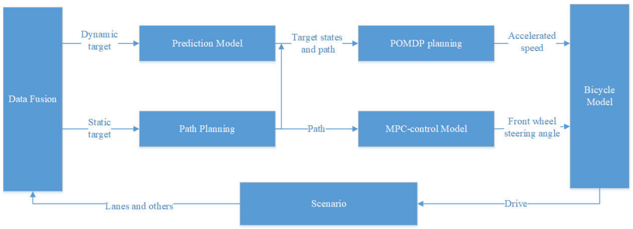

FIGURE 1. **Simulink simulation architecture.**

easier to cause the tracking error to become larger because of the frequent fluctuation of the velocity. Vehicle MPC control [18], [19] is the most commonly used method, which can be divided into the dynamics and kinematics according to the characteristics of modeling with different advantages. The combination of the two modeled MPC control methods can be used alternately to avoid their respective shortcomings. This control method is suitable for intersections with no traffic signs.

II. RELATED WORK

In this paper, we use MATLAB/Simulink simulation software to complete the whole closed-loop simulation experiment.

Fig.1 is the architecture diagram of the entire closed-loop simulation module. The vehicle adopts a bicycle model, and the simulation scene is set up by the Driving Scenario Designer APP of MATLAB. The data of the camera and radar in the data fusion module are used separately, and the camera is used to collect the lane line. Radar is devoted to detect the target vehicle. The prediction module carries out a separate setting for the scene, and will give the path of every conceivable target point of the target vehicle, as well as the probability of its driving intention. And it also simulates the output given by the actual prediction algorithm. For the path planning module of static obstacle avoidance, A* algorithm [20], [21] and ego-vehicle coordinate system are utilized to search the path. The obstacle avoidance of dynamic obstacles is based on the combination of POMDP and time-distance coordinate system. The overall process is shown in Fig.2. Before that, the static obstacle avoidance method of vehicle in road coordinate system is introduced. As shown in Fig.3.

The coordinate system in Fig.3 adopts the Frenet coordinate system. The S axis is the center line of the road, which can be obtained from the lane line. The L axis is the direction perpendicular to the S axis. The red part in the Fig.3 is the projection of the static obstacle in the Frenet coordinate system. At this time, there are two ways for the vehicle to perform static obstacle avoidance: the first is to directly change the lane, using polynomial curves, Bezier curves, B-spline curves, sine-cosine curves or Dubins curves to connect to the center line of adjacent lanes; the second is to keep the lane unchanged and expand the driving range, which needs to evenly scatter points in Frenet coordinate system. The single point of each layer adds smoothing cost, distance obstacle cost and distance reference line cost as its final cost, and then the convex space is opened up by dynamic programming method to find a rough solution, such as the green curve part of the graph. Finally, the path of static obstacle avoidance is planned by smoothing the green curve.

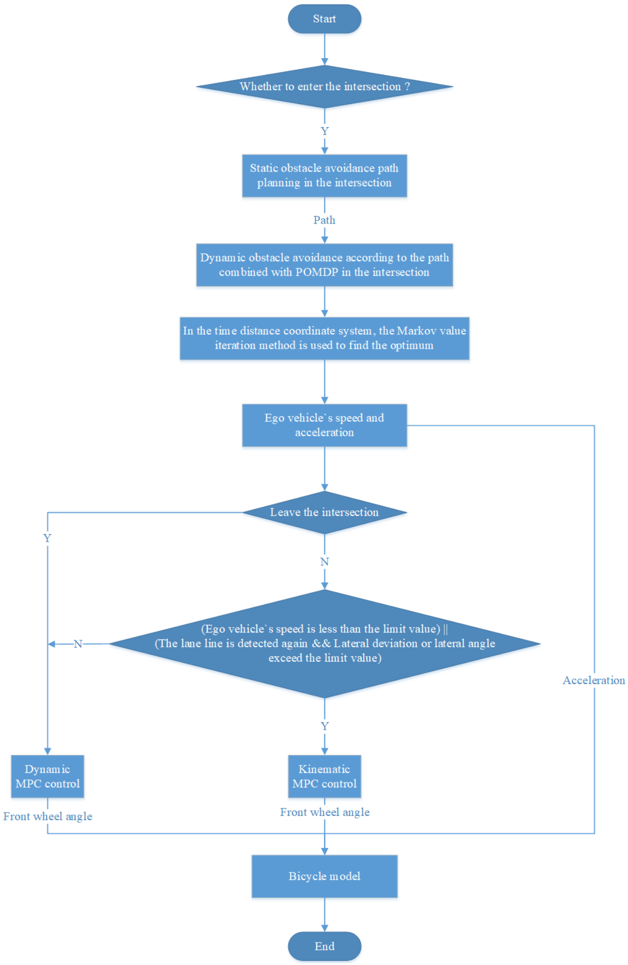
Start
Whether to enter the intersection ?
Y
Static obstacle avoidance path planning in the intersection
Path
Dynamic obstacle avoidance according to the path combined with POMDP in the intersection
In the time distance coordinate system, the Markov value iteration method is used to find the optimum
Ego vehicle`s speed and acceleration
Leave the intersection
Y
N
(Ego vehicle`s speed is less than the limit value) ||
(The lane line is detected again && Lateral deviation or lateral angle exceed the limit value)
N
Y
Acceleration
Dynamic MPC control
Kinematic MPC control
Front wheel angle
Front wheel angle
Bicycle model
End

FIGURE 2. **The overall implementation steps of method in a flowchart.**

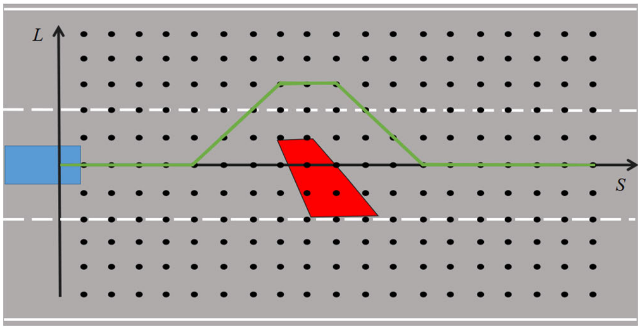

FIGURE 3. **Static obstacle avoidance in Frenet coordinate system.**

The static obstacle avoidance at the intersection is a little special. The entrance and exit of the intersection are structured roads, and only the middle connection is a large open area where all lane line information is not visible. In fact, all driving routes of vehicles passing through this area are shown in Fig.5. This means that open space at intersections

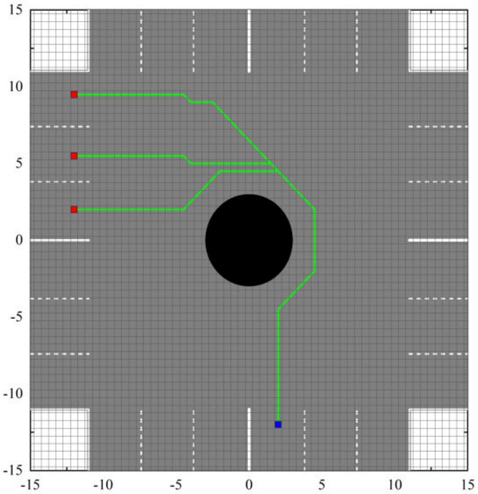

FIGURE 4. Paths searched using A * at roundabouts.

are actually unstructured areas affected by structured roads. When the unstructured area affected by the structured road is a regular shape, the driving path of the vehicle can be connected by an arc. When there are static obstacles, it is still possible to take the arc as the S-axis of the Frenet coordinate system and use the above second way to avoid static obstacles. But when the unstructured area affected by the structured road is irregularly shaped, similar to a roundabout, the directly connected curve is difficult to find, so a path needs to be searched out. A* search is used and the formula is as follows

$$f = g + h \tag{1}$$

where adopts the Euclidean distance. The effect of the search is shown in Fig.4. Blue is the starting point and red are the target points. When the vehicle turns left through this intersection, the path is one of the green curves, of course, the path needs further smoothing.

When the intersection shape is irregular and there are complex static obstacles, the vehicle must get rid of the influence of the structured road on the unstructured area if it wants to pass through the area safely, and the path of the static obstacle avoidance must be an irregular curve. As shown in Fig.5, if there are too many static obstacles in the scene, when the blue vehicle passes through the intersection, its path can pass through the area in front of the red vehicle or the green vehicle. Under normal circumstances, the path planned by the blue vehicle cannot pass through these areas under the influence of the structured road. At this time, it can consider Yi et al.'s improved P_RRT* algorithm [22], Xiang et al.'s improved A* algorithm [23] or Wang et al.'s PGI-RRT* algorithm [24]. These algorithms are effective algorithms in the face of complex static obstacles. In addition, when the vehicle passes through the scene of complex static obstacles, the speed is generally low, and there will be sufficient time to plan. In practice, it is very rare for intersections to have irregular shapes and complex static obstacles. By changing lanes, or keeping the lane unchanged and expand the driving range or simple A * search, most of the static obstacle avoidance path planning of the vehicle can be satisfied.

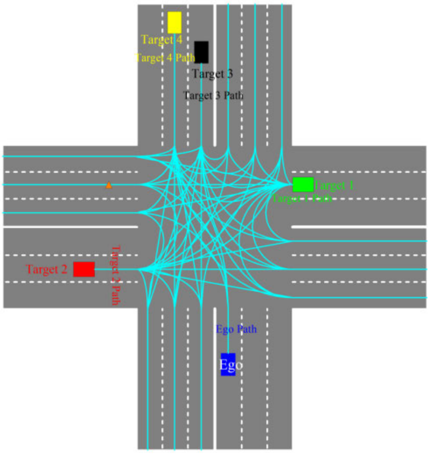

FIGURE 5. Reference path network of vehicles at the intersection.

The premise of the research is that the vehicle can perform static obstacle avoidance. And the speed planning module of POMDP and the control module of MPC are the main contents of this paper. The main validation is the feasibility of the POMDP model framework combined with time distance coordinate system, the hybrid control effect of dynamic MPC and kinematic MPC. Therefore, static obstacles are not installed in the intersection scenarios without traffic signs.

III. PROPOSED METHOD

A. POMDP MODEL COMBINED WITH DISTANCE TIME COORDINATE SYSTEM

POMDP decision process is a sequential decision model in a dynamic environment, which is partially known about the environment state. Based on MDP, POMDP considers the partial observability of the environment. That means the agent cannot obtain the environment state accurately. In unmanned driving, the vehicle itself can not fully obtain the driving path and intention of other vehicles.

A POMDP can be described by a seven-tuple array $[S,A,T,O,Z,R,\gamma]$. S is the set of states in the decision-making process. A is the set of actions in the decision-making process. T is the transition probability between states. O represents the observation state of the environment by the agent. Z is the observation function or the observation probability. R is the payoff after taking an action to reach the next state. γ is the discount factor for subsequent rewards.

The probability of a state transition depends on the action

$$P^a_{ss'} = p(s'|s,a) = p(S'_{t+1} = s'|S_t = s, A_t = a) \tag{2}$$

The observation probability of the agent is also related to the action

$$Z^a_{s'o} = p(o|s',a) = p(O_{t+1} = o|S'_{t+1} = s', A_t = a) \tag{3}$$

Compared with MDP, POMDP also has an initial probability distribution $b(s)$, which represents the probability distribution of the current state. The updated formula is

$$b'(s_{t+1}) = p(s_t|o, a, b) = \frac{p(o|s', a) \cdot \sum_s p(s'|s, a)b(s)}{p(o|a, b)} \tag{4}$$

$$p(o|a, b) = \sum\nolimits_{s'} p(o|s', a) \sum\nolimits_{s} p(s'|s, a)b(s) \tag{5}$$

The Bellman optimal equation of the reward function is

$$R_t^*(b) = \max_{a \in A}[\rho(b, a) + \gamma \sum_{o \in O} p(o|b, a)R_{t+1}^*(b')] \tag{6}$$

where $\rho(b, a) = \sum_{s'} b(s)V(s, a)$, *V(s,a)* represents the immediate reward for performing action A in state S.

Depending on the basic definition model of POMDP, the car passing through the intersection scenarios without traffic signs is modeled. It is assumed that the state of the autonomous vehicle is known, that is, the autonomous vehicle is completely observable in the process of moving. The coordinate system is the coordinate system of the car body. The trajectory and velocity of the ego-vehicle are decoupled and decomposed into a trajectory planning and a velocity planning. The intersection scenario without traffic signs is shown in Fig.5. The number of one-way lanes is 3, and the number of lanes has no essential impact on scene modeling. In the figure, cyan curves are the driving route connected by each lane, and the orange triangle is the driving target position of the ego vehicle. The red, green, black and yellow rectangles are the location of the target vehicle. There is no signal in this scene, that is, no traffic light sign, no lane restriction sign, and no headlight sign.

The forward direction of the autonomous vehicle at the intersection is determined according to the driving intention of the driver. After the vehicle's driving intention is determined, the target lane is also determined, which is the middle lane by default. The actual driving will also be affected by other vehicles to change the target lane, which is not considered in this paper. At this time, the trajectory that the self-propelled vehicle needs to track laterally has been completely determined, and the longitudinal speed planning needs to be completed with the POMDP model.

According to the prediction module, there are four kinds of driving intentions, namely [*turn left, turn right, straight on, turn round*], and each intention has different probability. Driving intention is the set of state S in POMDP and the set of observed state O of the vehicle, and the corresponding probability of driving intention is the initial probability distribution $b(s)$. The set of action A is [*accelerate, constant, decelerate*], with the difference between A_{Target} and $A_{Ego-Car}$. The intention state of the target vehicle will not change due to the change in the behavior of the ego vehicle, which means that the state transition matrix T is fixed. The observation probability depends on the camera and radar of the ego-vehicle. And the observation probability matrix Z is fixed. According to equations (2), (3), (4) and (5), the probability value of the target vehicle's driving intention at the next time can be obtained. The probability value of each intention can be divided according to the number of lanes of the target car. In a single lane, the probability of the target vehicle's driving intention is also the probability of driving in the lane. In the case of dual lanes, the weight of the inner lane is larger than that of the outer lane, and the probability of the target vehicle's driving intention is assigned according to the weight. In the case of three lanes, the middle lane has more weight and the two lanes have less weight. In order to simplify the model, the weights are evenly distributed. The probability of the target vehicle's driving intention was assigned to the driving path of each lane, and the driving path of the autonomous vehicle was used for collision detection to extract the target vehicle's driving path with intersecting trajectories. The speed planning operation was performed with time distance coordinate system, as shown in Fig.6.

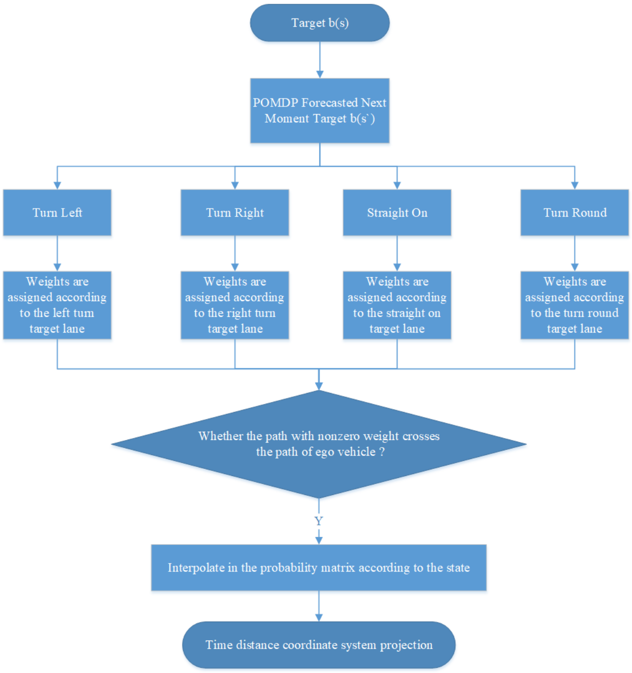
Target b(s)
POMDP Forecasted Next Moment Target b(s`)
Turn Left
Turn Right
Straight On
Turn Round
Weights are assigned according to the left turn target lane
Weights are assigned according to the right turn target lane
Weights are assigned according to the straight on target lane
Weights are assigned according to the turn round target lane
Whether the path with nonzero weight crosses the path of ego vehicle ?
Y
Interpolate in the probability matrix according to the state
Time distance coordinate system projection

FIGURE 6. Speed planning process of POMDP combined with distance time coordinate system.

Combined with the flow chart, each target vehicle path that intersects with the ego-vehicle path has its corresponding probability value, which is represented by $P_{intersect}$. When operating with the time distance coordinate system, it is necessary to extract other states of the target car except the driving intention. Once the intersection is determined, the distance from the ego-vehicle to the intersection and the distance from the target vehicle to the intersection can be calculated, which are represented by d_{Ego} and d_{Target} respectively. These two distances are corresponding to each other: use d_{Target} for target calculation, use the corresponding d_{Ego} for projection on the distance time coordinate system.

The state transition matrix of the target vehicle in the path is

$$\begin{pmatrix} d_{k+1} \\ v_{k+1} \end{pmatrix} = \begin{pmatrix} 1 & \Delta t \\ 0 & 1 \end{pmatrix} \begin{pmatrix} d_k \\ v_k \end{pmatrix} + \begin{pmatrix} \frac{\Delta t^2}{2} \\ \Delta t \end{pmatrix} a_k \tag{7}$$

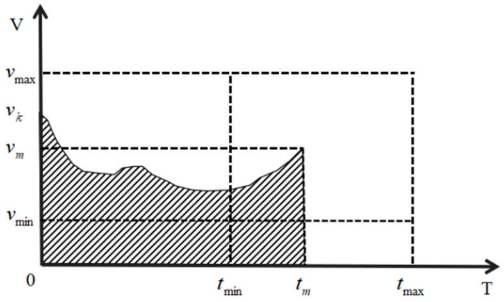

FIGURE 7. Curve of solving the equation.

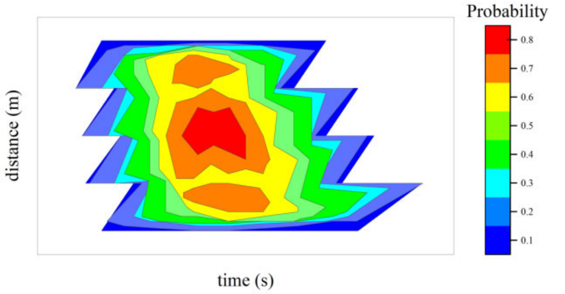

FIGURE 8. Probability distribution of a vehicle arriving at the collision point.

where $a_k \in A_{Target}$. Because the intersection point is solved, the d_k in the target transition matrix is limited by the maximum value, and the a_k of the target vehicle is also limited by the maximum value, namely $a_k \in(-a_{max}, a_{max})$. If the vehicle observes the position state d_k and speed state v_k of the target vehicle, the state quantities v_{min} and v_{max} of the two limit conditions reaching the intersection point can be calculated, and the corresponding limit time t_{min} and t_{max} can also be calculated. v_{max} is generally limited by the road speed. The limit state quantity of the target vehicle is discretized respectively, and the possibility of each combination is solved. In fact, it is to solve all curves satisfying the constraint conditions of formula (8) in Fig.7.

The horizontal axis is time, and the vertical axis is speed. The curve satisfies the formula

$$\begin{cases} V(0) = v_k \\ V(t_m) = v_m \\ V(t)_{\max} = v_{\max} \\ V(t)_{\min} = v_{\min} \\ V(t)'_{\max} = a_{\max} \\ V(t)'_{\min} = -a_{\max} \\ \int_0^{t_m} V(t) = d_{Target} \end{cases} \tag{8}$$

where $t_m \in (t_{min}, t_{max})$, $v_m \in (v_{min}, v_{max})$.

All curves satisfying the constraints of time and speed can be calculated, and the probability of the target vehicle reaching the state of the crossing point can be calculated according to the probability formula. Assuming m_{max}=n, the probability matrix is shown in Formula (9).

$$q = \begin{array}{cccc} \text{intersection} & t_{\min} & \rightarrow & t_{\max} \\ v_{\min} & p_{11} & \cdots & p_{1n} \\ \downarrow & \vdots & \ddots & \vdots \\ v_{\max} & p_{n1} & \cdots & p_{nn} \end{array} \tag{9}$$

As the distance and speed of the target vehicle are changed, its probability matrix will also change. d_{Target} is discretized, assuming it is discretized into k parts. Then the probability matrix is transformed into

$$\begin{array}{cccc} \text{target} & 0 & \rightarrow & v_{\max} \\ 0 & q_{11} & \cdots & q_{1n} \\ \downarrow & \vdots & \ddots & \vdots \\ d_{Target} & q_{k1} & \cdots & q_{kn} \end{array} \tag{10}$$

According to probability matrix (9) and probability matrix (10), the probability P obtained by interpolation is multiplied by $P_{intersect}$, which is the additional weight on the time distance coordinate system. The probability of a_k in the above solution process is an equal random process, and its action probability can be selected according to the driving style of the actual driver. Due to the existence of observation probability, the observation state needs to be discretized and then interpolated. After all the additional weights of a single target vehicle are added to the time distance coordinate system (Fig.8), it is a contour plane with different probabilities. The corresponding different colors are an additional weight loss when the ego-vehicle reaches this point during speed planning.

When multiple target vehicles are detected, the additional weight of each vehicle will be projected onto a time distance coordinate system. If loss weight of different target vehicles is taken to the maximum value for each point in the coordinate system, it represents the possibility that each point is occupied by only one vehicle. When the vehicle with the largest weight occupies this point, other vehicles cannot occupy this position at the same time. In this case, no collision will occur by default. When the weight of each point in the coordinate system is accumulated, it represents the dangerous degree of each point at the corresponding time, which means the degree of collision of the car at this point at the corresponding time. Results will be calculated on the time distance coordinate system according to the loss weight after the final superposition. Since the state of the ego-vehicle is completely known, the classical value iteration algorithm of MDP can be used to solve the problem. However, the difference is that the complete action curve on the time distance coordinate system is not required, and only the discrete action $A_{Ego-Car}$ of the ego-vehicle at the next moment can be iteratively stable. Depending on the corresponding cost, the optimal is selected, which represents the action to be executed at the next moment.

B. KINEMATIC MODEL MPC

The kinematics of the vehicle is modeled. According to Fig.9, it is assumed that the state quantity deviation and control quantity deviation of the vehicle are as shown in

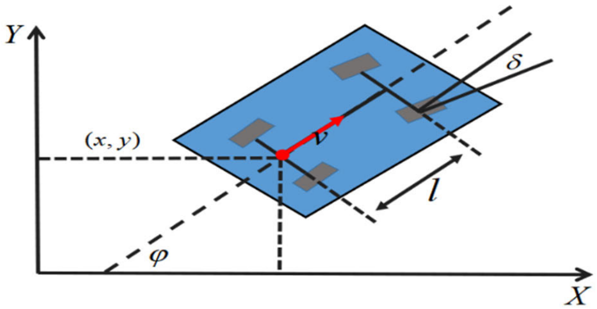

FIGURE 9. Kinematic model of vehicle.

Equation (11)

$$\hat{x}=\begin{bmatrix}\dot{x}-\dot{x}_r\\ \dot{y}-\dot{y}_r\\ \dot{\phi}-\dot{\phi}_r\end{bmatrix},\hat{u}=\begin{bmatrix}v-v_r\\ \delta-\delta_r\end{bmatrix}\tag{11}$$

The kinematics equation of the vehicle is

$$\begin{cases}\dot{x}=v_x=v\cos\phi\\ \dot{y}=v_y=v\sin\phi\\ \dot{\phi}=\frac{v\tan\delta}{l}\end{cases}\tag{12}$$

According to Fig.9 and Equations (11) and (12), the discrete state space equation of the vehicle kinematic model is as follows

$$\hat{x}(k+1)=\begin{bmatrix}1&0&-Tv_r\sin\phi_r\\ 0&1&Tv_r\cos\phi_r\\ 0&0&1\end{bmatrix}\hat{x}(k)+\begin{bmatrix}T\cos\phi_r&0\\ T\sin\phi_r&0\\ T\frac{tab\phi_r}{l}&T\frac{v_r}{l\cos^2\delta_r}\end{bmatrix}\hat{u}(k)=a\hat{x}(k)+b\hat{u}(k)$$

Define the output equation

$$y(k)=\begin{bmatrix}1&0&0\\ 0&1&0\\ 0&0&1\end{bmatrix}\hat{x}(k)=c\hat{x}(k)\tag{13}$$

Build a new state vector

$$\xi(k)=\begin{bmatrix}\hat{x}(k)\\ \hat{u}(k-1)\end{bmatrix}\tag{14}$$

Then the new state space expression is

$$\begin{aligned}\xi(k+1)&=\begin{bmatrix}\hat{x}(k+1)\\ \hat{u}(k)\end{bmatrix}\\ &=\begin{bmatrix}a\hat{x}(k)+b\hat{u}(k-1)+b\hat{u}(k)-b\hat{u}(k-1)\\ \hat{u}(k-1)+\hat{u}(k)-\hat{u}(k-1)\end{bmatrix}\\ &=A\xi(k)+B\Delta\hat{u}(k)\end{aligned}\tag{15}$$

where $A=\begin{bmatrix}a&b\\ 0&I_{N_u}\end{bmatrix}$, $B=\begin{bmatrix}b\\ I_{N_u}\end{bmatrix}$. Combined with equation (13), the new output equation is

$$\eta(k)=\begin{bmatrix}I_{N_x}&0\end{bmatrix}\cdot\begin{bmatrix}\hat{x}(k)\\ \hat{u}(k-1)\end{bmatrix}=C\xi(k)\tag{16}$$

Assuming that the prediction time domain is Np and the control time domain is Nc. Equation (15) is derived in multiple steps as, shown in the equation at the bottom of the next page.

In the formula, the sum of the exponent of A and the control step of u is Np+k-1. Similarly, formula (15) can be deduced in multiple steps as, shown in the equation at the bottom of the next page.

When

$$Y=\begin{bmatrix}\eta(k+1)\\ \eta(k+2)\\ \cdots\\ \eta(k+Nc)\\ \cdots\\ \eta(k+Np)\end{bmatrix},\Psi=\begin{bmatrix}CA\\ CA^2\\ \cdots\\ CA^{Nc}\\ \cdots\\ CA^{Np}\end{bmatrix},$$

$$\Delta U=\begin{bmatrix}\Delta\hat{u}(k)\\ \Delta\hat{u}(k+1)\\ \Delta\hat{u}(k+2)\\ \Delta\hat{u}(k+3)\\ \cdots\\ \Delta\hat{u}(k+Nc-1)\end{bmatrix}$$

$$\Theta=\begin{bmatrix}CB&0&0&\cdots&0\\ CAB&CB&0&\cdots&0\\ \vdots&\vdots&\vdots&\vdots&\vdots\\ CA^{Nc-1}B&CA^{Nc-2}B&CA^{Nc-3}B&\cdots&CA^0B\\ \vdots&\vdots&\vdots&\vdots&\vdots\\ CA^{Np-1}B&CA^{Np-2}B&CA^{Np-3}B&\cdots&CA^{Np-Nc}B\end{bmatrix}.$$

Then, the output equation is

$$Y=\Psi\xi(k)+\Theta\Delta U\tag{17}$$

The reference value of the output of the system is defined as

$$\begin{aligned}Y_r=&[\eta_r(k+1),\eta_r(k+2),\ldots,\eta_r(k+Nc),\\ &\ldots,\eta_r(k+Np)]^T=[0,0,\ldots,0]^T\end{aligned}\tag{18}$$

Since the state quantity is in the form of error, the reference value of the output quantity is 0. Suppose $E=\Psi\xi(k)$, $Q_Q=I_{Np}\otimes Q$, $R_R=I_{Np}\otimes R$, the defined optimization objective function is

$$\begin{aligned}J&=\hat{Y}^TQ_Q\hat{Y}+\Delta U^TR_R\Delta U\\ &=\Delta U^T(\Theta^TQ_Q\Theta+R_R)\Delta U+2E^TQ_Q\Theta\Delta U\\ &\quad+E^TQ_QE-Y_rQ_Q\Theta\Delta U+Y_r^TQ_QY-2Y_r^TQ_QE\end{aligned}\tag{19}$$

Let $H=\Theta^TQ_Q\Theta+R_R$, $g=E^TQ_Q\Theta$, and cancel the constant term of equation (17). The equation can be written as

$$\begin{aligned}\min_{\Delta U}J&=2\left(\frac{1}{2}\Delta U^TH\Delta U+g^T\Delta U\right)\\ \Leftrightarrow\min_{\Delta U}J&=\frac{1}{2}\Delta U^TH\Delta U+g^T\Delta U\end{aligned}\tag{20}$$

For the control quantity and control increment, it becomes the following recursive formula, as shown in the equation at the bottom of the next page.

When

$$U_t = \begin{bmatrix} \hat{u}(k-1) \\ \hat{u}(k-1) \\ \hat{u}(k-1) \\ \cdots \\ \hat{u}(k-1) \end{bmatrix}, \quad A_t = \begin{bmatrix} I_2 & 0 & 0 & \cdots & 0 \\ I_2 & I_2 & 0 & \cdots & 0 \\ I_2 & I_2 & I_2 & \cdots & 0 \\ \cdots & \cdots & \cdots & \ddots & 0 \\ I_2 & I_2 & I_2 & I_2 & I_2 \end{bmatrix},$$

$$\Delta U_t = \begin{bmatrix} \Delta\hat{u}(k) \\ \Delta\hat{u}(k+1) \\ \Delta\hat{u}(k+2) \\ \cdots \\ \Delta\hat{u}(k+Nc-1) \end{bmatrix}.$$

formula is rewritten as

$$U = \begin{bmatrix} \hat{u}(k) \\ \hat{u}(k+1) \\ \hat{u}(k+2) \\ \cdots \\ \hat{u}(k+Nc-1) \end{bmatrix} = U_t + A_t \Delta U \tag{21}$$

According to Equations (13) and (14), the constraints of the control quantity are

$$U_{\min} = \begin{bmatrix} \hat{u}_{\min} \\ \hat{u}_{\min} \\ \hat{u}_{\min} \\ \cdots \\ \hat{u}_{\min} \end{bmatrix} \leq \begin{bmatrix} \hat{u}(k) \\ \hat{u}(k+1) \\ \hat{u}(k+2) \\ \cdots \\ \hat{u}(k+Nc-1) \end{bmatrix} \leq \begin{bmatrix} \hat{u}_{\max} \\ \hat{u}_{\max} \\ \hat{u}_{\max} \\ \cdots \\ \hat{u}_{\max} \end{bmatrix} = U_{\max} \tag{22}$$

Combined with formula (18), it can be written as

$$\begin{aligned} U_{\min} &\leq U_t + A_t \Delta U \leq U_{\max} \\ &\Rightarrow \begin{cases} A_t \Delta U \leq U_{\max} - U \\ -A_t \Delta U \leq -U_{\min} + U_t \end{cases} \end{aligned} \tag{23}$$

In summary, the vehicle control model is transformed into a standard quadratic programming.

C. DYNAMIC MODEL MPC

The dynamics of the vehicle is modeled. According to Fig.10, the Y-axis of the vehicle can be obtained by applying Newton's second law

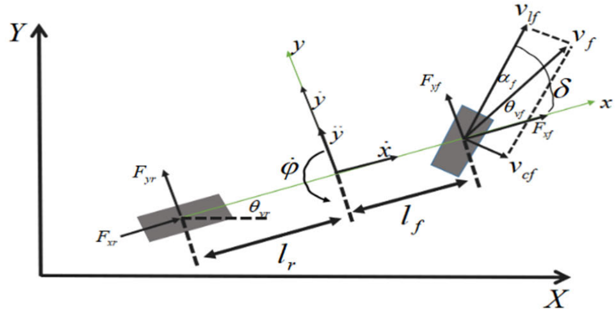

FIGURE 10. Dynamic model of vehicle.

$$ma_y = F_{yf} + F_{yr} \tag{24}$$

where a_y is the lateral acceleration at the center of mass of the vehicle, and F_{yf} and F_{yr} are the lateral forces exerted by the ground on the front and rear tires. The lateral acceleration is composed of two parts: the acceleration caused by the lateral movement of the vehicle along the Y-axis of the vehicle sound and the centripetal acceleration caused by the yaw movement of the body. Then formula (21) can be written as

$$m(\ddot{y} + v_x \dot{\phi}) = F_{yf} + F_{yr} \tag{25}$$

The torque balance equation of the vehicle around the Z-axis is

$$I_z \ddot{\phi} = F_{yf} l_f - F_{yr} l_r \tag{26}$$

For the two lateral forces on the tire, considering the assumption of small side deviation angle, the lateral force exerted by the ground on the tire has a linear relationship with the tire side deviation angle. And the front wheel side deviation angle is

$$\alpha_f = \delta - \theta_{vf} \tag{27}$$

where θ_{vf} is the angle between the speed of the front tire and the longitudinal axis of the body, and δ is the steering angle of the front wheel. The side angle of the rear wheel is

$$a_r = -\theta_{vr} \tag{28}$$

$$\begin{aligned} \xi(k+Nc) &= A^{Nc}\xi(k) + A^{Nc-1}B\Delta\hat{u}(k) + A^{Nc-2}B\Delta\hat{u}(k+1) + \ldots + A^0 B\Delta\hat{u}(k+Nc-1) \\ &\vdots \\ \xi(k+Np) &= A^{Np}\xi(k) + A^{Np-1}B\Delta\hat{u}(k) + A^{Np-2}B\Delta\hat{u}(k+1) + \ldots + A^0 B\Delta\hat{u}(k+Np-1) \end{aligned}$$

$$\begin{aligned} \eta(k+Nc) &= CA^{Nc}\xi(k) + CA^{Nc-1}B\Delta\hat{u}(k) + CA^{Nc-2}B\Delta\hat{u}(k+1) + \ldots + CA^0 B\Delta\hat{u}(k+Nc-1) \\ &\vdots \\ \eta(k+Np) &= CA^{Np}\xi(k) + CA^{Np-1}B\Delta\hat{u}(k) + CA^{Np-2}B\Delta\hat{u}(k+1) + \ldots + CA^0 B\Delta\hat{u}(k+Np-1) \end{aligned}$$

$$\begin{aligned} \hat{u}(k) &= \hat{u}(k-1) + \Delta\hat{u}(k) \\ &\cdots \\ \hat{u}(k+Nc-1) &= \hat{u}(k+Nc-2) + \Delta\hat{u}(k+Nc-1) = \hat{u}(k-1) + \Delta\hat{u}(k) + \Delta\hat{u}(k+1) + \ldots + \Delta\hat{u}(k+Nc-1) \end{aligned}$$

The lateral tire forces of the front and rear wheels are

$$F_{yf} = 2C_{af}(\delta - \theta_{vf}) \tag{29}$$

$$F_{yr} = -2C_{ar}\theta_{vr} \tag{30}$$

The center of mass of the vehicle to the tire is regarded as a rigid body, according to the rigid body kinematics

$$\tan\theta_{vf} = \frac{v_y + \dot{\phi} l_f}{v_x} \Rightarrow \theta_{vf} = \frac{v_y + \dot{\phi} l_f}{v_x} \tag{31}$$

$$\tan\theta_{vr} = \frac{v_y - \dot{\phi} l_r}{v_x} \Rightarrow \theta_{vr} = \frac{v_y - \dot{\phi} l_r}{v_x} \tag{32}$$

The expected yaw velocity and lateral acceleration of the vehicle are respectively

$$\phi_{des} = \frac{v_x}{r} \tag{33}$$

$$a_{y,des} = \frac{v_x^2}{r} = v_x \dot{\phi}_{des} \tag{34}$$

The change rate of yaw angle deviation and yaw angle deviation is

$$\hat{\phi} = \phi - \phi_{des} \tag{35}$$

$$\dot{\hat{\phi}} = \dot{\phi} - \dot{\phi}_{des} \tag{36}$$

Lateral acceleration deviation

$$\hat{a}_y = a_y - a_{y,des} = \ddot{y} + v_x \dot{\hat{\phi}} \tag{37}$$

When equation (35) is integrated, the transverse velocity deviation is

$$\hat{v}_y = \dot{\hat{y}} = \dot{y} + v_x \hat{\phi} \tag{38}$$

According to equations (21) - (32), the tracking error state equation of vehicle lateral dynamic model can be obtained as follows

$$\frac{\mathrm{d}}{dt}\begin{bmatrix} \hat{y} \\ \dot{\hat{y}} \\ \hat{\phi} \\ \dot{\hat{\phi}} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & -\frac{2C_{af}+2C_{ar}}{m\cdot v_x} & \frac{2C_{af}+2C_{ar}}{m} & \frac{2l_r C_{ar}-2l_f C_{af}}{m\cdot v_x} \\ 0 & 0 & 0 & 1 \\ 0 & \frac{2l_r C_{ar}-2l_f C_{af}}{I_z\cdot v_x} & \frac{2l_f C_{af}-2l_r C_{ar}}{I_z} & -\frac{2l_f^2 C_{af}-2l_r^2 C_{ar}}{I_z\cdot v_x} \end{bmatrix} \cdot \begin{bmatrix} \hat{y} \\ \dot{\hat{y}} \\ \hat{\phi} \\ \dot{\hat{\phi}} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{2C_{af}}{m} \\ 0 \\ \frac{2l_f\cdot C_{af}}{I_z} \end{bmatrix}\delta + \begin{bmatrix} 0 \\ \frac{2l_r C_{ar}-2l_f C_{af}}{m\cdot v_x} - v_x \\ 0 \\ -\frac{2l_f^2 C_{af}+2l_r^2 C_{ar}}{I_z\cdot v_x} \end{bmatrix}\dot{\phi}_{des}$$

The state space of equation is discretized

$$\hat{x}(\mathrm{k}+1) = As\cdot\hat{x}(k) + Bs\cdot\delta(k) + Gs\cdot\dot{\phi}_{des}(k) \tag{39}$$

where, as shown in the equation at the bottom of the next page.

Define the output equation

$$y = C\cdot\hat{x} \tag{40}$$

where

$$C = \begin{bmatrix} 1\,0\,0\,0 \\ 0\,1\,0\,0 \\ 0\,0\,1\,0 \\ 0\,0\,0\,1 \end{bmatrix}.$$

When the predicted step size is *Np* and the output step size is *Nc*, where the maximum value of *Nc* is *Np*-1, the output equation is transformed into, as in (41), shown at the bottom of the next page.

The theoretical yaw velocity cannot be obtained directly, therefore it is discretized as a coefficient and calculated in the matrix. According to the defined quadratic optimization objective function, it is reduced to

$$J = u^T Hu + fu \tag{42}$$

where $Q_Q = I_{Np}\otimes Q, R_R = I_{Np}\otimes R, R_R = I_{Np}\otimes R, H = \bar{B}Q_Q\bar{B}^T + R_R, f = 2E^T Q_Q\bar{B}, E = \bar{A}\cdot\hat{x} + \bar{G}\cdot EHR - y\_ref, R = \begin{bmatrix} 1\,0 \\ 0\,1 \end{bmatrix}$,

$$Q = \begin{bmatrix} qa & 0 & 0 & 0 \\ 0 & qb & 0 & 0 \\ 0 & 0 & qc & 0 \\ 0 & 0 & 0 & qd \end{bmatrix}, y\_ref = \begin{bmatrix} ref\cdot\hat{x} \\ ref^2\cdot\hat{x} \\ \vdots \\ ref^{Np}\cdot\hat{x} \end{bmatrix},$$

$$EHR = \begin{bmatrix} \frac{v_x}{r}\cdot 0.5^{1-1} \\ \frac{v_x}{r}\cdot 0.5^{2-1} \\ \vdots \\ \frac{v_x}{r}\cdot 0.5^{Nc-1} \end{bmatrix}.$$

where *qa*, *qb*, *qc* and *qd* are the expected reference coefficients of the corresponding terms respectively. *ref* is the weight of the theoretical deviation coordinate point, which is 0 in general, representing no deviation. In the actual control system, the constraint conditions of the system state quantity and the control quantity need to be satisfied. The formula for the constraint of the control quantity is

$$u_{\min}(t+k) \le u(t+k) \le u_{\max}(t+k), k = 0.1\ldots Nc - 1 \tag{43}$$

The first term after the quadratic programming solution is used as the actual control input for steering. When the vehicle runs on a curved road, especially the curve with large curvature, the dynamic MPC feedback control can not completely eliminate the tracking error, so the feedforward controller related to road curvature is introduced to help eliminate the tracking error

$$\delta_{ff} = \frac{l_f + l_r}{r} + \left[\frac{l_r\cdot m}{2C_{af}\cdot(l_f + l_r)} - \frac{l_f\cdot m}{2C_{ar}\cdot(l_f + l_r)}\right]\cdot\frac{v_x^2}{r} \tag{44}$$

Since then, the MPC modeling of the vehicle kinematics and dynamics model is completed. Then the vehicle control, according to the vehicle heading angle and speed, alternate uses different control methods.

IV. SIMULATION AND RESULTS

After the model is established, the simulation will be carried out. The intersection scenario without traffic signs is generally a low or middle speed scene, thereby the optimal speed set during the simulation is 8 m/s. When there is no collision risk, the self-propelled vehicle performs P-controller acceleration. The algorithm starts to work when the target vehicle is 20 m away from the intersection, and the update time step is 0.1 s.

Different trajectory curves in Fig.11 and Fig.12 represent different restriction conditions, which have different degrees of influence on the autonomous vehicle. In Trajectory 1, the acceleration action is a ∈(-2,0,2). When the lateral deviation and heading angle deviation of the vehicle are too large, the kinematic MPC will be used to control the motion. Trajectory 2 adds P-controller on the basis of Trajectory 1 to carry out more detailed operations. Trajectory 3 is based on Trajectory 1, adding the operation of kinematic MPC control when the vehicle speed is less than 3 m/s. Trajectory 4 is to add P-controller on the basis of Trajectory 3. Trajectory 5 is adopted kinematic MPC when the speed threshold is less than 4 m/s based on Trajectory 4. Trajectory 6 is completely controlled by dynamic MPC based on Trajectory 2. Actually, it is found that the Trajectory 2 and Trajectory 4 coincide. That is, the speed of the car does not fall below 3 m/s under the restriction of Trajectory 4, which means that the speed does not play a role in the restriction of the kinematic MPC at this time.

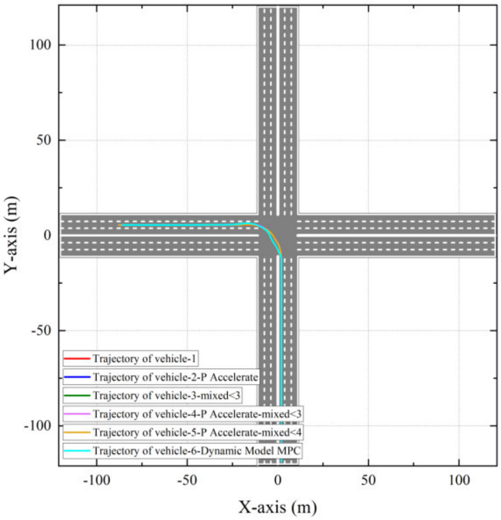

FIGURE 11. Global map driving trajectories.

Fig.11 is the global map coordinate diagram, which is the whole driving track of the vehicle in the simulation.

$$\hat{x}=\begin{bmatrix}\hat{y}\\ \dot{\hat{y}}\\ \hat{\phi}\\ \dot{\hat{\phi}}\end{bmatrix},As=\begin{bmatrix}1 & T & 0 & 0\\ 0 & 1-\frac{2C_{af}+2C_{ar}}{m\cdot v_x}T & \frac{2C_{af}+2C_{ar}}{m}T & \frac{2l_r\cdot C_{ar}-2l_f\cdot C_{af}}{m\cdot v_x}T\\ 0 & 0 & 1 & T\\ 0 & \frac{2l_r\cdot C_{ar}-2l_f\cdot C_{af}}{I_z\cdot v_x}T & \frac{2l_f\cdot C_{af}-2l_r\cdot C_{ar}}{I_z}T & 1-\frac{2l_f^2\cdot C_{af}+2l_r^2\cdot C_{ar}}{I_z\cdot v_x}T\end{bmatrix},$$

$$Bs=\begin{bmatrix}0\\ \frac{2C_{af}\cdot T}{m}\\ 0\\ \frac{2l_f\cdot C_{af}\cdot T}{I_z}\end{bmatrix},Gs=\begin{bmatrix}0\\ \frac{2l_r\cdot C_{ar}-2l_f\cdot C_{af}}{m\cdot v_x}T-v_x\cdot T\\ 0\\ -\frac{2l_f^2\cdot C_{af}+2l_r^2\cdot C_{ar}}{I_z\cdot v_x}T\end{bmatrix}.$$

$$\begin{aligned}&\mathrm{y}=\bar{A}\hat{x}+\bar{B}\delta+\bar{G}\\ &\bar{A}=\begin{bmatrix}C\cdot As\\ C\cdot As^2\\ \vdots\\ C\cdot As^{Np}\end{bmatrix},\\ &\bar{B}=\begin{bmatrix}C\cdot Bs & 0 & 0 & 0\\ C\cdot As\cdot Bs & C\cdot Bs & \cdots & 0\\ \vdots & \vdots & \vdots & \vdots\\ C\cdot As^{Np-1}\cdot Bs & C\cdot As^{Np-2}\cdot Bs & \cdots & C\cdot As^{Np-Nc}\cdot Bs\end{bmatrix},\\ &\bar{G}=\begin{bmatrix}C\cdot Gs & 0 & 0 & 0\\ C\cdot As\cdot Gs & C\cdot Gs & \cdots & 0\\ \vdots & \vdots & \vdots & \vdots\\ C\cdot As^{Np-1}\cdot Gs & C\cdot As^{Np-2}\cdot Gs & \cdots & C\cdot As^{Np-Nc}\cdot Gs\end{bmatrix}\end{aligned}\tag{41}$$

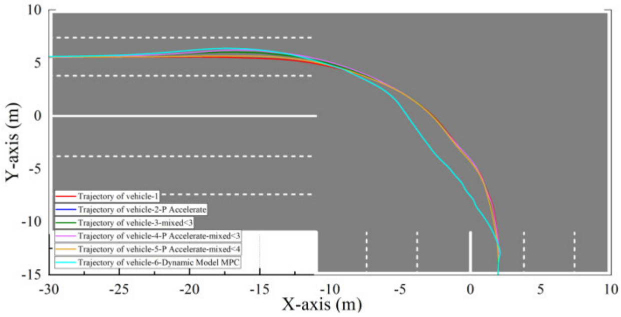

FIGURE 12. Enlarged view of the trajectories at the intersection.

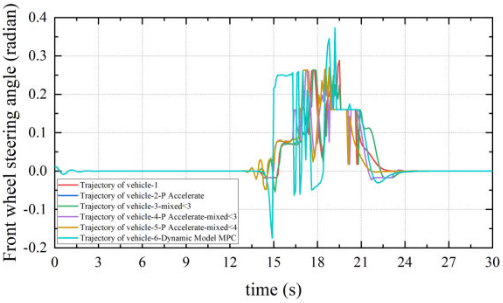

FIGURE 13. Front wheel steering angle.

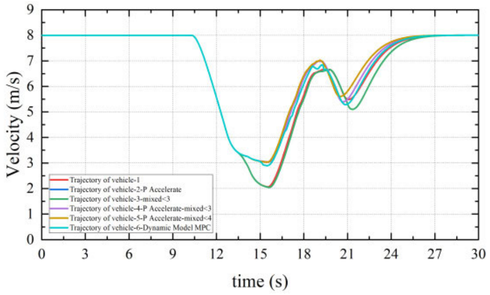

FIGURE 14. Velocity curves.

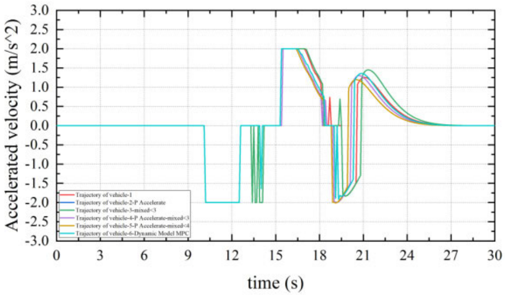

FIGURE 15. Acceleration curves.

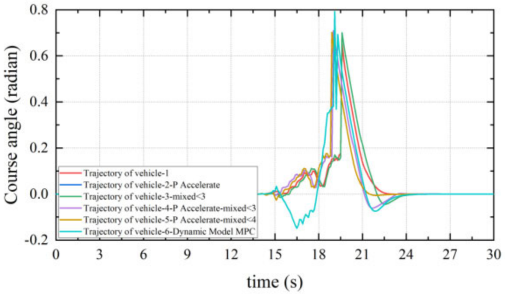

FIGURE 16. Heading angle deviation.

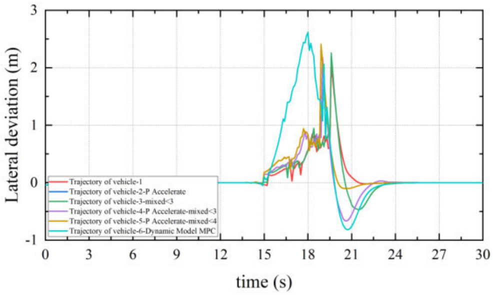

FIGURE 17. Lateral deviation.

Fig.12 is an enlarged view of the trajectory of the car at the intersection. In the figure, Trajectory 1 and Trajectory 5 are the smoothest curves, indicating that the actual process of the car is also the most stable. Compared with Trajectory 1, Trajectory 5 is restricted by P-controller and kinematic MPC, which improves the control of speed and trajectory tracking and make the vehicle experience more comfortable during driving. The corresponding front wheel angle, speed and acceleration curves are all reflected in Fig.13, Fig.14 and Fig.15. Trajectory 6 is the least smooth curve, which is manifested as the irregular curve of the front wheel angle, with serious fluctuation up and down, and the corresponding track tracking effect is not satisfactory, which is also reflected in Fig.16 and Fig.17. Other trajectory curves are more and more stable with the restriction of conditions, and the corresponding front wheel angle, lateral deviation and heading angle deviation are also appeared.

As shown in the speed curve of Fig.14 and acceleration curve of Fig.15, the whole process is divided into four stages: 0-10 s stabilization stage, 10-15 s deceleration stage,15-19 s acceleration stage, 19-21 s deceleration stage, and 21-30 s final stage.

0-10 s, the car runs smoothly at the best speed, other parameters are 0.

10-15 s, at this time the car finds the right front intersection when the green vehicle is passing. According to the algorithm, there will be a heavy collision cost when the ego-vehicle continues to pass, while the cost is less as it slows down. Therefore, a deceleration operation is carried out at this time, which is also reflected in 10-15 s in Fig.15. In Fig.14, the curve appears to be separated after 13 s, which means that the car observed that the green vehicle finished passing and then found that the red vehicle in front of the left needed to pass. Depending on the algorithm, the cost of acceleration is higher than that of deceleration, hence the ego-vehicle decelerates for the second time. At this stage, the acceleration curves in Fig.15 are all shown as point braking. Since P-controller is added to Trajectory 2, Trajectory 4, Trajectory 5 and Trajectory 6, the point brake acceleration of the ego-vehicle is smaller than Trajectory 1 and Trajectory 3, and the control is more refined. Fig.14 shows that the velocity

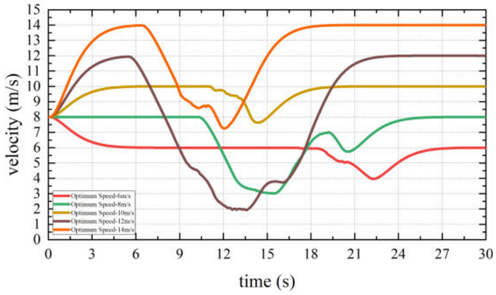

FIGURE 18. Different target velocities.

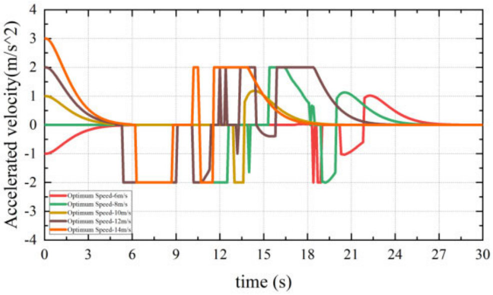

FIGURE 19. Acceleration of different target velocities.

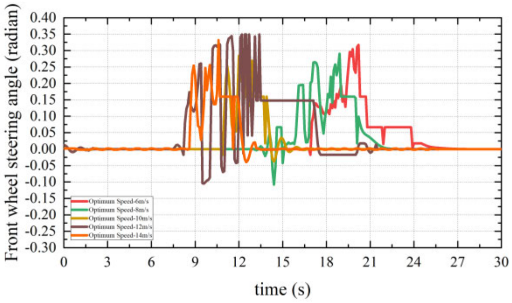

FIGURE 20. Front wheel angles at different target velocities.

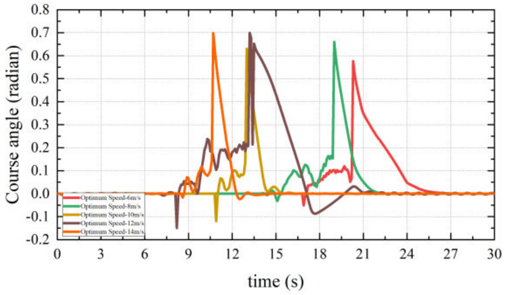

FIGURE 21. Heading angle deviation of different target velocities.

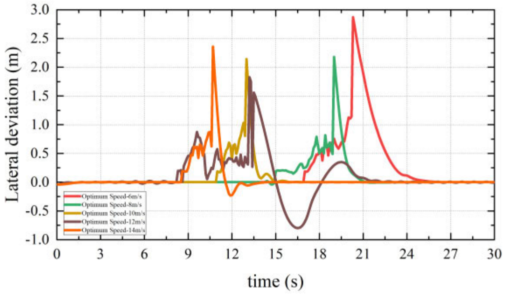

FIGURE 22. Lateral deviation of different target velocities.

curves of Trajectory 1 and Trajectory 3 are steeper than other curves, and the velocity decreases even lower.

15-19 s, at this time, the car finds that the black vehicle also would like to pass the intersection. According to the algorithm, the cost of acceleration is much smaller than that of deceleration, and the car starts to speed up. Trajectories 1 and 3 show a small fluctuation in Fig.15, which means that at this time, the vehicle has completely eliminated the risk of collision and starts to implement the acceleration of ego-vehicle's P-controller. However, other trajectories already have more refined operations when executing the algorithm, consequently there is no need to perform other operations when exiting the algorithm.

19-21 s, at this stage, the autonomous vehicle heading towards the target lane redetects the lane lines of the target lane and then abandons the reference line given by the prior condition. According to Fig.16 and Fig.17, the lateral deviation of the ego-vehicle from the lane center line of the target lane is about 2.5 m, and the deviation of the heading angle is about 0.7 radian. However, the vehicle is generally very close to the target road when it detects the lane line, hereby it needs to decelerate and correct in a very short time. The influence of this part can be well reduced by using kinematic MPC control. As shown in Fig.16 and Fig.17, the lateral deviation and heading angle deviation of the car are corrected within 1.5 seconds, and the lateral deviation is controlled within 0.5 meters and the heading angle deviation is controlled within 0.1 radian. After that, they quickly converge to 0.

After 21 s, the car completely leaves the intersection, and the road ahead is smooth. Start to execute the P accelerator, accelerate to the best speed, and drive out of the intersection.

The optimal target speed of the car is set to different values when making it through the same conditions of the scene. Fig.18 shows the velocity curve of the same scene, which has experienced at least one deceleration cycle, and the difference between the upper and lower limits is 10m/s when the fluctuation range is maximum. Fig.19 shows the corresponding acceleration curve.

Fig.20 shows the curve of the front wheel angle at different speeds through the intersection. The fluctuation of the front wheel angle is all within 0.35 rad, and the most severe curve fluctuation is at the optimal target speed of 12 m/s, its maximum speed is 12 m/s and minimum speed is 2 m/s. The drastic fluctuation of speed correspondingly causes the change of the front wheel angle, which is also reflected in the heading angle deviation in Fig.21. However, in lateral deviation in Fig.22, the maximum amplitude of the corresponding lateral deviation is the smallest, which means that the distance from the reference line is the smallest.

The driving target of the vehicle is changed to make the autonomous vehicle go straight through the intersection. Fig.23 is its speed curve, and Fig.24 is its acceleration curve. The curves of the dynamic MPC and kinematic MPC in the two figures are coincident, which means that when the vehicle is controlled in two ways, the response of the vehicle is

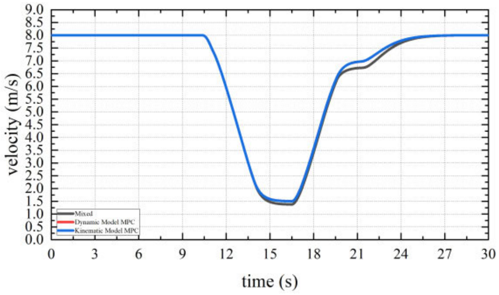

FIGURE 23. Velocity in straight motion.

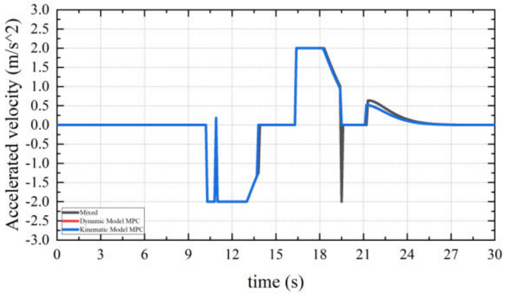

FIGURE 24. Acceleration in straight motion.

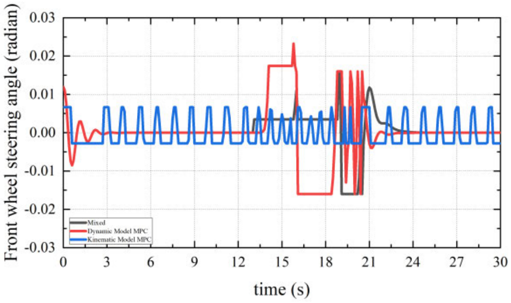

FIGURE 25. Front wheel angle in straight motion.

consistent. However, when the two control methods are used alternately, the curves appear slightly different, meaning that the control also is under a slight effect on the speed change.

When the car has been accelerated and decelerated, the performance will be distinct with different control algorithms. Fig.25 is the comparison of front wheel angles of vehicles using different control algorithms as going through the intersection. Dynamic MPC control will have an obvious jitter when the speed changes greatly, and this change starts from the low speed of the car. When the speed climbs, the front wheel angle quickly tends to be stable. The control algorithm of kinematic MPC fluctuates back and forth in the form of harmonics throughout the whole process, whereas it is maintained in a small range. When the vehicle speed decreases, the range of fluctuation becomes smaller. After alternating the two control algorithms, it is obviously that the front wheel rotation angle has no jitter of dynamic MPC at low speed and no harmonic fluctuation of kinematic MPC at high speed, which combines the advantages of the two algorithms. The deviation of heading angle is also reflected in Fig.26.

The heading angle of kinematic MPC decreases significantly at 12 s, and the deviation tends to 0 obviously in the whole low-speed driving interval. However, after 18 s, when the vehicle speed increases, the heading angle deviation becomes markedly and violently dithered, and then tends to be harmonically stable. On the contrary, the dynamic MPC approaches 0 when the heading angle deviation is at high speed. But after 12 s, when the vehicle speed decreases, the heading angle deviation becomes significantly and rapidly larger, and finally becomes stable when the vehicle speed increases. When the two methods are used alternately, the heading angle deviation of the vehicle at high speed is significantly more stable than the kinematic MPC. The heading angle deviation increases slowly at low speed, and the trend is markedly lower than that of MPC alone. When the vehicle speed increases again, the change of heading angle deviation is also more stable than when it is used alone. Fig.27 shows the comparison of their lateral deviations.

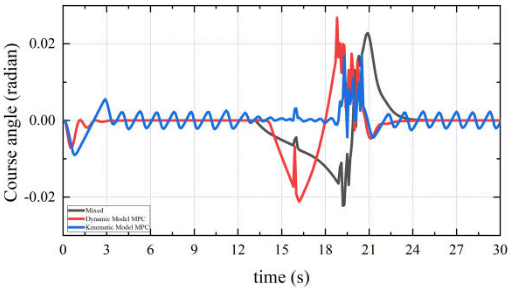

FIGURE 26. Heading angle deviation in straight motion.

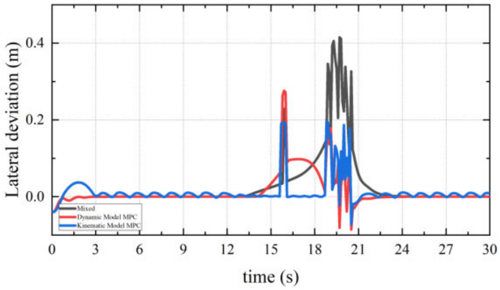

FIGURE 27. Lateral deviation in straight motion.

Although the lateral deviations become larger when they are mixed, they are still within the acceptable range of 0.4 m.

V. CONCLUSION

This paper studies the scenario of cars passing through intersection scenarios without traffic signs. POMDP and time distance coordinate system are used to carry out speed planning to avoid the problem of high computational force consumption when solving POMDP model. The optimal acceleration action of the vehicle can be solved within the specified time, so that the vehicle can effectively avoid other vehicles and pass at a higher speed, in order to improve the efficiency of traffic. Compared with using dynamic MPC or kinematic MPC alone, using dynamic MPC and kinematic MPC alternately in control can effectively reduce the tracking error, improve the stationarity of tracking, and carry out finer control under the condition of frequent velocity fluctuation.

According to the graph of all simulation results, it can be observed that the time period of autonomous vehicles passing through the entire intersection is concentrated within 8-24 s, and the whole process lasts about 16 s. In this process, three vehicles that might be in collision are prevented, which has high timeliness. At the same time, in the control process of turning, the fluctuation range of the front wheel angle is always controlled within 0.35 rad, without drastic fluctuation. This method may also have implications in the treatment of other traffic problems.

REFERENCES

[1] J. Yuan, M. A. Abdel-Aty, L. Yue, and Q. Cai, "Modeling real-time cycle-level crash risk at signalized intersections based on high-resolution event-based data," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 11, pp. 6700–6715, Nov. 2021.

[2] Y. Ren, Y. Wang, G. Yu, H. Liu, and L. Xiao, "An adaptive signal control scheme to prevent intersection traffic blockage," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 6, pp. 1519–1528, Jun. 2017.

[3] J. M. Scanlon, R. Sherony, and H. C. Gabler, "Models of driver acceleration behavior prior to real-world intersection crashes," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 3, pp. 774–786, Mar. 2018.

[4] M. Montemerlo, J. Becker, S. Bhat, H. Dahlkamp, D. Dolgov, S. Ettinger, D. Haehnel, T. Hilden, G. Hoffmann, B. Huhnke, and D. Johnston, "Junior: The Stanford entry in the urban challenge," *J. Field Robot.*, vol. 25, no. 9, pp. 569–597, Sep. 2008.

[5] S. Kammel, J. Ziegler, B. Pitzer, M. Werling, T. Gindele, D. Jagzent, J. Schroder, M. Thuy, M. Goebl, F. V. Hundelshausen, and O. Pink, "Team AnnieWAY's autonomous system for the 2007 DARPA urban challenge," *J. Field Robot.*, vol. 25, no. 9, pp. 615–639, 2008.

[6] M. Werling, J. Ziegler, S. Kammel, and S. Thrun, "Optimal trajectory generation for dynamic street scenarios in a Frenét frame," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA)*, May 2010, pp. 987–993.

[7] C. Hubmann, M. Aeberhard, and C. Stiller, "A generic driving strategy for urban environments," in *Proc. IEEE 19th Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2016, pp. 1010–1016.

[8] L. Xu, G. Yin, G. Li, A. Hanif, and C. Bian, "Stable trajectory planning and energy-efficience control allocation of lane change maneuver for autonomous electric vehicle," *J. Intell. Connected Vehicles*, vol. 1, no. 2, pp. 55–65, Dec. 2018.

[9] W. Zhan, C. Liu, C.-Y. Chan, and M. Tomizuka, "A non-conservatively defensive strategy for urban autonomous driving," in *Proc. IEEE 19th Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2016, pp. 459–464.

[10] J. Chen, B. Yuan, and M. Tomizuka, "Deep imitation learning for autonomous driving in generic urban scenarios with enhanced safety," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Nov. 2019, pp. 2884–2890.

[11] V. Sezer, T. Bandyopadhyay, D. Rus, E. Frazzoli, and D. Hsu, "Towards autonomous navigation of unsignalized intersections under uncertainty of human driver intent," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2015, pp. 3578–3585.

[12] S. Brechtel, T. Gindele, and R. Dillmann, "Probabilistic decision-making under uncertainty for autonomous driving using continuous POMDPs," in *Proc. 17th Int. IEEE Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2014, pp. 392–399.

[13] S. Ulbrich and M. Maurer, "Probabilistic online POMDP decision making for lane changes in fully automated driving," in *Proc. 16th Int. IEEE Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2013, pp. 2063–2067.

[14] W. L. Song, G. M. Xiong, and H. Y. Chen, "Intention-aware autonomous driving decision-making in an uncontrolled intersection," *Math. Problems Eng.*, vol. 2016, Apr. 2016, Art. no. 1025349.

[15] H. Bai, S. Cai, N. Ye, D. Hsu, and W. S. Lee, "Intention-aware online POMDP planning for autonomous driving in a crowd," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2015, pp. 454–460.

[16] S. Ross, J. Pineau, S. Paquet, and B. Chaib-draa, "Online planning algorithms for POMDPs," *J. Artif. Intell. Res.*, vol. 32, pp. 663–704, Jul. 2008.

[17] T. Lee and Y. J. Kim, "Massively parallel motion planning algorithms under uncertainty using POMDP," *Int. J. Robot. Res.*, vol. 35, no. 8, pp. 928–942, Jul. 2016.

[18] R. Verschueren, S. De Bruyne, M. Zanon, J. V. Frasch, and M. Diehl, "Towards time-optimal race car driving using nonlinear MPC in real-time," in *Proc. 53rd IEEE Conf. Decis. Control*, Dec. 2014, pp. 2505–2510.

[19] H. Wang, B. Liu, X. Ping, and Q. An, "Path tracking control for autonomous vehicles based on an improved MPC," *IEEE Access*, vol. 7, pp. 161064–161073, 2019.

[20] D. Harabor and A. Grastien, "Online graph pruning for pathfinding on grid maps," in *Proc. AAAI Conf. Artif. Intell.*, Aug. 2011, vol. 25, no. 1, pp. 1114–1119.

[21] J. Chen, R. C. Holte, S. Zilles, and N. R. Sturtevant, "Front-to-end bidirectional heuristic search with near-optimal node expansions," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, Aug. 2017, pp. 489–495.

[22] J. Yi, R. Sun and H. Bai, "Path planning of a manipulator based on an improved P_RRT* algorithm," *Complex Intell. Syst.*, 2022, vol. 8, no. 3, pp. 2227–2245.

[23] D. Xiang, H. Lin, J. Ouyang, and D. Huang, "Combined improved A* and greedy algorithm for path planning of multi-objective mobile robot," *Sci. Rep.*, vol. 12, no. 1, p. 13273, Aug. 2022.

[24] Y. Wang, D. Liu, H. Zhao, Y. Li, W. Song, M. Liu, L. Tian, and X. Yan, "Rapid citrus harvesting motion planning with pre-harvesting point and quad-tree," *Comput. Electron. Agricult.*, vol. 202, Nov. 2022, Art. no. 107348.

WEI LIANG received the B.S. degree in electronic information engineering from the University of Shanghai for Science and Technology, Shanghai, China, in 2007, the M.S. degree in sensor technology from the Coburg University of Applied Sciences, Coburg, Germany, in 2010, and the Ph.D. degree in materials science from the University of Bayreuth, Bayreuth, Germany, in 2014.

From 2010 to 2014, she was a Research Assistant at the Institute of Sensor and Actuator Technology, HS Coburg, Coburg. Since 2015, she has been an Associate Professor with the School of Mechanical and Automotive Engineering, Shanghai University of Engineering Science. Her research interests include the development of droplet propulsion, trajectory tracking, and path planning.

Dr. Liang received awards and honors, including the Young Eastern Scholar of Shanghai and the Morning Plan of Shanghai.

HAITAO XING was born in Xiangyang, Hubei, China, in 1997. He received the B.S. degree in energy and power engineering form the Wuhan Institute of Technology, Wuhan, in 2020. He is currently pursuing the M.S. degree in mechanical engineering with the Shanghai University of Engineering Science, Shanghai.

Since 2020, his research direction is intelligent connected vehicles.