A Safe Motion Planning and Reliable Control Framework for Autonomous Vehicles

Huihui Pan, *Senior Member, IEEE*, Mao Luo, Jue Wang, Tenglong Huang, and Weichao Sun, *Senior Member, IEEE*

***Abstract*—Accurate trajectory tracking is unrealistic in real-world scenarios, however, which is commonly assumed to facilitate motion planning algorithm design. In this paper, a safe and reliable motion planning and control framework is proposed to handle the tracking errors caused by inaccurate tracking by coordinating the motion planning layer and controller. Specifically, motion space is divided into safe regions and risky regions by designing the movement restraint size dependent on tracking error to construct the repulsive potential field. The collision-free waypoint set can then be obtained by combining global search and the proposed waypoint set filtering method. The planned trajectory is fitted by an optimization-based approach which minimizes the acceleration of the reference trajectory. Then, the planned trajectory is checked and modified by the designed anti-collision modification to ensure safety. Using invertible transformation and adaptive compensation allows the transient trajectory tracking errors to be limited within the designed region even with actuator faults. Because tracking error is considered and margined at the planning level, safety and reliability can be guaranteed by the coordination between the planning and control levels under inaccurate tracking and actuator faults. The advantages and effectiveness of the proposed motion planning and control method are verified by simulation and experimental results.**

***Index Terms*—Inaccurate tracking, safe motion planning, actuator faults, reliable control, autonomous vehicles.**

I. Introduction

AUTONOMOUS vehicles [1], including autonomous robot vehicles [2], unmanned vehicles [3], and surface vehicles [4], performing a basic navigation task generally require perception, motion planning, and control to work together [3], [5]. The motion planning [6], [7], [8], [9] layer is taken as the brain, which processes the perception information obtained from sensors to generate a safe motion or trajectory without collisions. The control level [10] designs a suitable controller to track the desired motion or trajectory. To simplify motion planning algorithm design, it is commonly assumed that the planned trajectory can be tracked with complete accuracy [11]. Because tracking errors caused by inaccurate tracking are difficult to obtain in advance, it leads to collision risk caused by tracking errors in the planning layer that are hard to avoid. In this paper, it is expected to design a motion planning and control scheme that allows the system to be safe and collision-free under inaccurate tracking by coordinating motion control and planning algorithms in the presence of tracking errors.

The collision-free [12], [13] safe trajectory generation from the start point to the target point generally contains two stages: searching the global path and performing local smoothing [11]. Classical motion planning methods for global path search include Dijkstra [14], A^* [15], [16], [17] algorithms and artificial potential field (APF) approach [18], etc. The APF can model the motion space intuitively, however, the traditional APF tends to fall into local optimality. The heuristic search, such as Dijkstra and A^* algorithms, can search the optimal path. A^* algorithm is used to search for the original point set of the global path in this paper. The global path search algorithms are mature, and this part is not the focus of this paper. In contrast, many interesting algorithms have emerged for local trajectory smoothing. The challenge of how to ensure the safety of planned trajectories while performing local smoothing is an important and attractive one that needs further research. It is worth noting that Bezier-type [19] motion planning methods, such as the classical Bezier algorithm and the B-Spline algorithm [20], can significantly improve trajectory smoothness. Bezier algorithm can ensure that the generated trajectories are arbitrary-order differentiable. Polynomial-type trajectory planning methods, including cubic spline [21], [22] and quintic spline, can generate time-dependent motion trajectories based on the location of the start and target points, desired velocity, and desired acceleration. Further, optimization-based polynomial trajectory planning algorithms [23], including minimum velocity, mini acceleration, minimum jerk [24], and minimum snap algorithms, take the first to fourth-order derivatives of the trajectory as cost. It is worth noting that, to produce a sufficiently smooth reference trajectory, Bezier-type motion planning methods do not guarantee the generated planned trajectory passes through waypoints. Polynomial-type algorithms can constrain the reference trajectory to pass through the waypoints, and optimization-based polynomial planning algorithms can further optimize trajectory performance. However, by fitting the trajectory according to the original waypoints in safety areas, the generated

Manuscript received 14 January 2024; accepted 27 January 2024. Date of publication 31 January 2024; date of current version 12 June 2024. This work was supported in part by the National Natural Science Foundation of China under Grant 62173108, in part by the Heilongjiang Post Doctoral Science Foundation under Grant LBH-TZ2111, in part by the Major Scientific and Technological Special Project of Heilongjiang Province under Grant 2021ZX05A01, and in part by the Fundamental Research Funds for the Central Universities under Grant HIT.OCEF.2022012. *(Corresponding author: Huihui Pan.)*

Huihui Pan, Mao Luo, Tenglong Huang, and Weichao Sun are with the Research Institute of Intelligent Control and Systems, Harbin Institute of Technology, Harbin 150001, China (e-mail: huihuipan@hit.edu.cn; 21s004077@stu.hit.edu.cn; huangtenglong@hit.edu.cn; w.sun@hit.edu.cn).

Jue Wang is with the Ningbo Institute of Intelligent Equipment Technology Company Ltd., Ningbo 315200, China, and also with the Department of Automation, University of Science and Technology of China, Hefei 230027, China (e-mail: juewang@hit.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TIV.2024.3360418.

Digital Object Identifier 10.1109/TIV.2024.3360418

fitted trajectory is still at risk of collision with the obstacle. Meanwhile, the original global waypoints may lack useful path information. The expectation is that by streamlining the useless waypoints while adjusting the original risky waypoints to enable the generation of collision-free planned trajectories. Meanwhile, the collision risk caused by inaccurate tracking is also taken into account. Therefore, this paper proposes a safe motion planning method to obtain a collision-free and safe trajectory by waypoint fitting in the case of inaccurate tracking.

An interesting and novel perspective of ensuring system security is the control barrier function (CBF) technology [25], [26] which considers safety directly in the control layer, instead of coordinating the motion planning layer. The CBF approach can ensure system safety by directly dealing with safety constraints and potential control logic conflicts in the controller. The application in scenarios with multiple irregular obstacles can be further investigated. In contrast to the safety-oriented CBF approach, this paper constrains the tracking errors to prescribed regions by introducing invertible transformations, while designing planning algorithms to ensure safety according to the prescribed constraints [27]. It is worth noting that effective ideas, including the designed safety planner in [28], [29] and the motion planning method based on temporal logic [30], [31], provide interesting solutions for dealing with tracking errors. Coordinating the planning and control layers to ensure safety is more intuitive and provides more degrees of freedom to enhance safety.

In addition to system safety [32], the reliability of the system is also crucial [33]. Actuator faults [34] due to wear and tear, and aging of system actuator components, including loss-of-effectiveness fault and bias fault, can lead to system performance degradation and even system instability. Actuator faults can cause system states to deviate from the desired trajectory and lead to collisions. Further consideration of actuator faults to improve system reliability is therefore of great significance to system stability and safety. In summary, this paper proposes a safe motion planning and control framework to enhance the safety and reliability of autonomous vehicles under inaccurate tracking in the presence of tracking errors and actuator faults.

Specifically, the contributions of this paper are summarized as follows:

- In the motion planning layer, this paper designs a movement restraint size based on the prescribed constraints and geometric characteristics of autonomous vehicles to allow a margin for tracking error. The movement restraint size is introduced to construct the repulsive potential field, dividing the motion space into safe and risky regions.
- For the original point set generated based on the classical A^* algorithm, this paper designs a waypoint set filtering method, which refines and adjusts the original point set by the proposed selecting feature points, transforming corner points, and removing redundant points, to facilitate further smoothing.
- This paper introduces an optimization-based polynomial trajectory fitting method that minimizes trajectory acceleration. An anti-collision modification is presented by adjusting and adding auxiliary waypoints to ensure that the planned trajectory is safe and collision-free.
- A fault-tolerant controller is presented to constrain tracking errors to prescribed regions by employing transformation functions and the adaptive backstepping technique. Safety and reliability can be guaranteed by the coordination of the planning layer and controller under actuator faults and inaccurate tracking.

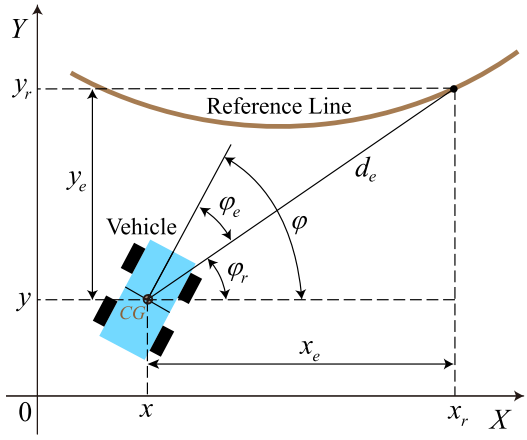


Fig. 1. Mathematical model for autonomous robot vehicles. CG denotes the center of mass.

II. Problem Statement

The primary objective of this paper is to develop a framework for safe motion planning and reliable tracking control of autonomous vehicles, even in the presence of actuator failures and trajectory tracking errors. The proposed framework addresses the critical issue of ensuring the safety and reliability of autonomous robot vehicles while performing complex tasks. Fig. 1 depicts the mathematical model of autonomous robot vehicles. The position of the robot in the Cartesian coordinate system is denoted by (x, y), while the yaw angle is represented by φ. The corresponding reference trajectory is indicated by (x_r, y_r), with the reference yaw angle denoted by φ.

Defining x_e and y_e as the tracking errors in the x and y directions, respectively, and d_e and φ as the distance error and yaw tracking error, as illustrated in Fig. 1, the corresponding errors [35] can be calculated by

$$x_e = x_r - x, y_e = y_r - y, d_e = \sqrt{x_e^2 + y_e^2}, \varphi_e = \varphi - \varphi_r \tag{1}$$

Referring to the kinematic relationship illustrated in Fig. 1, we can obtain

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\varphi} \end{bmatrix} = \begin{bmatrix} \cos(\varphi) & 0 \\ \sin(\varphi) & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \varpi \end{bmatrix} \tag{2}$$

where v and ϖ represent the linear velocity and angular velocity, respectively, while $u_1 = v$ and $u_2 = \varpi$ denote the control inputs. The possible fault [34] occurring in the i-th actuator can be modeled as follows

$$u_{if} = \alpha_{il} u_i + \alpha_{ib}, \ i = 1, 2 \tag{3}$$

where $0 < \alpha_{il} \leq 1$ denotes an unknown constant that indicates the loss-of-effective (LOE) fault occurring in the i-th actuator. To ensure controllability, $\alpha_{il} > 0$ is considered in this paper. $\alpha_{il} = 1$ signifies the absence of actuator LOE fault. Meanwhile, α_{ib} is another unknown constant that represents the actuator bias fault, with $\alpha_{ib} = 0$ indicating that the bias fault does not exist. Actuator faults can arise from various sources such as actuator wear, aging, and mechanical stress. It is worth noting

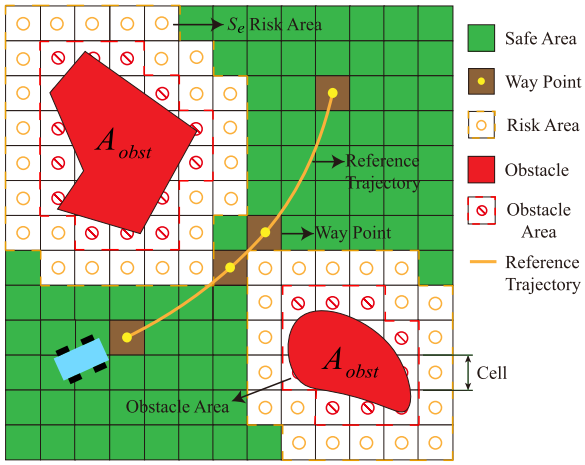


Fig. 2. Sensor data pre-processing to obtain grid map.

that multiple actuators may fail simultaneously, and multiple forms of faults may coexist, as considered in this paper.

Meanwhile, this paper considers the collision risks caused by inaccurate tracking at the motion planning level to ensure safety. In practice, reference trajectory accurate tracking is not realistic. Generally, it is assumed that accurate tracking can be achieved to simplify the challenges in motion planning.

This paper aims to develop a motion planning [36] and control framework that guarantees collision-free navigation, even in the presence of tracking inaccuracies. Specifically, the motion planning algorithm designed in this paper aims to achieve the following properties:

1) Safety and collision-free. Most motion planning algorithms assume accurate reference trajectory tracking to simplify the challenge; however, the motion planning proposed in this paper guarantees safety and collision avoidance even in the presence of trajectory tracking errors.
2) Reliability. In the presence of actuator faults, including bias and loss-of-effectiveness faults, the autonomous vehicle is still able to track reference values. The system is stable and can maintain safety and avoid collisions even in the event of actuator faults.
3) Satisfy transient and steady-state performance constraints. Users can design both transient and steady-state performance constraints [37] to meet practical requirements while ensuring safety and reliability. To ensure collision-free operation, it is necessary to restrict trajectory tracking errors within a predefined performance function.

III. Motion Planning

The motion planning algorithm presented in this paper takes into account the collision risks due to inaccurate tracking of the planned trajectory, ensuring safety. This differs from traditional planning algorithms, which assume that the planned trajectory can be tracked accurately.

A. Grid Map Pre-Processing

The sensor data is firstly pre-processed to generate a grid map, as plotted in Fig. 2. The construction of the grid map [38] can be facilitated by utilizing the given square grid width *Cell*. The obstacle area A_{obst} can be classified based on the locations and shapes of the obstacles. Fig. 2 demonstrates that a grid that intersects or is obstructed by an irregular obstacle is designated as a part of A_{obst}. Additionally, it is possible to account for the obstacle size and tracking error by generating a repulsive potential field. This potential field is utilized to partition the remaining region into safe and risky areas, which will be discussed further in the subsequent section. The next step is to search the waypoint set through path planning based on the previously established region divisions as depicted in Fig. 2. Then, a trajectory smoothing algorithm is presented and implemented to generate a collision-free reference trajectory, as described in detail below.

B. Movement Restraint Size

In practical scenarios, it is unrealistic for autonomous robot vehicles to track the planned reference trajectory with absolute accuracy. Trajectory tracking errors commonly occur, and it is significant to account for them when generating the desired trajectory to ensure collision-free navigation with obstacles, despite inaccurate tracking. The motion restraint size is determined not only by the physical size of the robot and map resolution but also by the potential magnitude of tracking errors. It is evident that the motion restraint size is proportional to the robot size and the magnitude of errors. The coarser map resolution requires a larger margin to ensure collision-free. The motion restraint size S_e presented in this paper can be calculated as follows:

$$S_e = w_1 * r_0 * (w_2 * A_e + w_3 * W) \tag{4}$$

where A_e denotes the magnitude of the tracking error, and W is the robot size. r_0 represents the width *Cell* of a single grid in the map. w_1, w_2, and w_3 serve as weighting factors. The motion restraint size S_e is calculated based on the selected weights, which serve as a foundation for collision-free motion planning.

From (4), it can be seen that the robot size W and tracking error A_e act on S_e, and the search region is divided by combining the repulsive potential field in the next subsection to generate the final planning trajectory. Different weighting parameters in (4) can regulate the role of different factors for S_e. A larger tracking error margin A_e or robot size W leads to a reduction of the safe region. Conversely, it leads to an increase in the safe region. This matches the actual situation, and we will further analyze it in Section V.

C. Repulsive Potential Field Generation

Using the constructed motion restraint size S_e, the repulsive potential field is created to obtain the feasible region. The repulsive potential field value U_{re} at $X_c(x, y)$ is given as

$$U_{re}(X_c) = \begin{cases} \eta_{re} - a_{re} \times \eta_{re} \times \rho^2, & \rho \le \rho_0 \\ 0, & \rho > \rho_0 \end{cases} \tag{5}$$

where $\eta_{re} > 0$ denotes the repulsive coefficient, $X_{obst}(x_0, y_0)$ represents the nearest obstacle, and $\rho = \sqrt{(x - x_o)^2 + (y - y_o)^2}$ is the shortest distance from $X_c(x, y)$ to obstacles. $\rho_0 > S_e > 0$ represents the maximum repulsive range, while $0 < a_{re} < \frac{1}{\rho_0^2}$ is the decay coefficient. The safety threshold $U_{re}(S_e)$ can be determined by $U_{re}(S_e) = \eta_{re} - a_{re} \times \eta_{re} \times S_e^2$. If $U_{re}(X_c) < U_{re}(S_e)$, the position $X_c(x, y)$ is considered safe; otherwise, it is considered risky. Fig. 3 illustrates how the repulsive potential field can be generated using (5). The constructed motion restraint size S_e

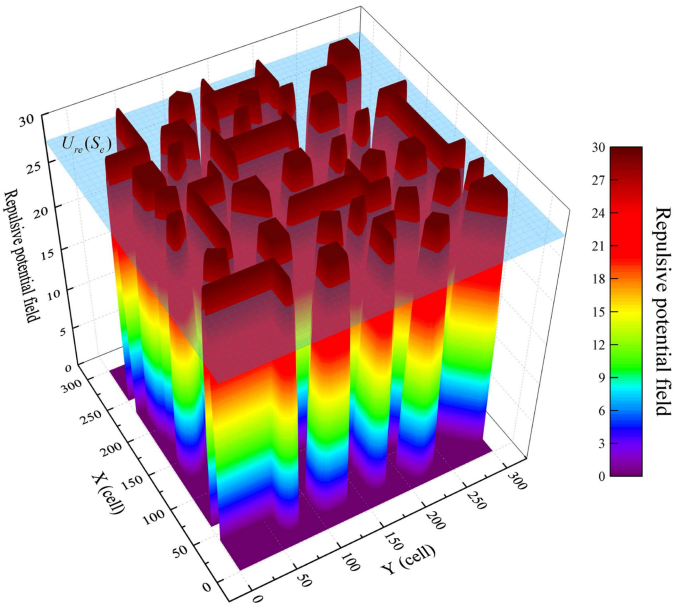


Fig. 3. Repulsive potential field.

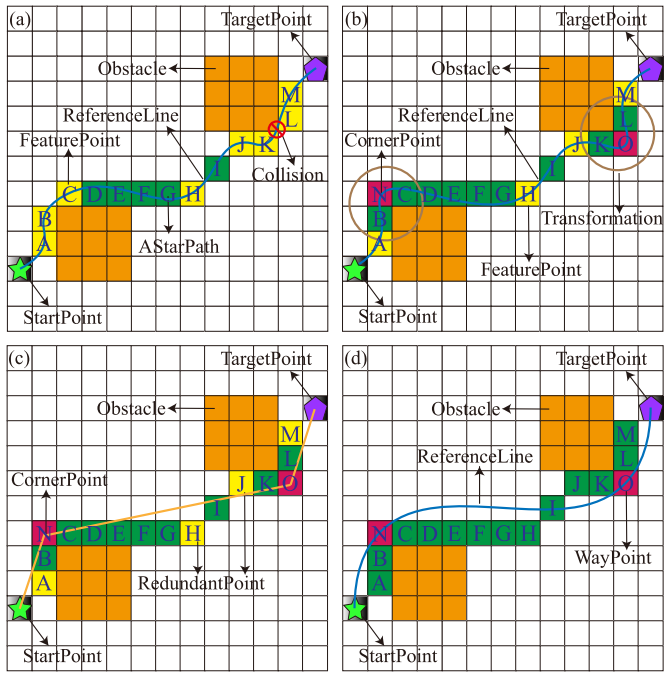


Fig. 4. Global path waypoint set Ω_f search process. The start point P_S is marked with a green pentagram grid, the target point P_T with a purple pentagon grid, and the obstacles with orange grids. Both green and yellow grids are initial waypoints and belong to the point set Ω_0. The yellow and red grids represent feature points and corner points, respectively.

facilitates the partitioning of the search space into the obstacle area A_{obst}, safe area A_s, and risky area A_r.

D. Waypoint Set Filtering

Waypoint set filtering and trajectory optimization are performed within the safety region A_s determined by the constructed motion restraint size S_e and repulsive potential field $U_{re}(X_c)$. Waypoint set filtering aims to reduce the path length, number of waypoints, and turns, thereby facilitating subsequent trajectory smoothing. As illustrated in Fig. 5, the A^* planning algorithm is utilized to generate the initial waypoint set, denoted as Ω_0. A heuristic distance is introduced into the search process as follows:

$$f(X_c) = g(X_c) + h(X_c) \tag{6}$$

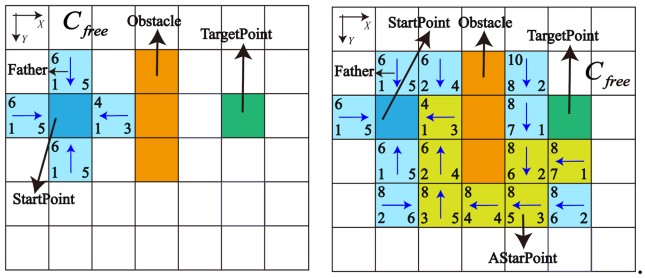
where $X_c = (x, y)$ indicates the current position, $f(X_c)$ denotes the search priority. X_{c_father} denotes the father point which is the same as the parent node in [16]. We can obtain

$$g(X_c) = g(X_{c\_father}) + dis(g(X_c), g(X_{c\_father})) \tag{7}$$

$$h(X_c) = |x - x_{target}| + |y - y_{target}| \tag{8}$$

where $dis(g(X_c), g(X_{c\_father}))$ is the Euclidean distance, $X_{target}(x_{target}, y_{target})$ denotes the target point. The one-step search process and the overall search process are illustrated in Fig. 5(a) and (b), respectively. The search principle is identical to [15], [16]. By using the classical A^* algorithm, Ω_0 is obtained.

Then, Ω_0 is refined by selecting feature points, transforming corner points, and removing redundant points to obtain the final waypoint set Ω_f. For clarity, the principle is illustrated with a simple scene search in Fig. 4. As depicted in Fig. 4(a), the A^* algorithm is used to search the initial waypoint set Ω_0, which includes the yellow and green grids. The substantial number of waypoints generated by the A^* algorithm can pose challenges for trajectory fitting and lead to poor tracking behavior. Therefore, it is crucial to meticulously optimize and filter the initial point set Ω_0 to guarantee that the resulting trajectory is both secure and well-suited for autonomous vehicles. The initial point



(a) One-step search process. (b) Overall search process.

Fig. 5. Implementation of A^* algorithm.

set $\Omega_0\{P_S, A, B, C, D, E, F, G, H, I, J, K, L, M, P_T\}$ filtering process is described as follows:

1) Selecting Feature Points (SFP): The initial point set Ω_0 contains numerous points that move in the same direction, providing limited trajectory information. Thus, the key feature points, specifically the waypoints where the direction of the path changes, are extracted. As shown in Fig. 4(a), vector $\overrightarrow{BC}$ has a different direction than vector $\overrightarrow{CD}$, indicating that point C is a feature point. Since vector $\overrightarrow{CD}$ and $\overrightarrow{DE}$ are in the same direction, point D is not considered a feature point. The waypoint set Ω_0 is simplified to $\Omega_1\{P_S, A, B, C, H, J, K, L, M, P_T\}$, resulting in a reduction of the number of waypoints by one-third while preserving the feature points that contain critical path information.

2) Transforming Corner Points (TCP): During the trajectory smoothing process, it is evident that corner points situated in close proximity to obstacles are prone to collision, such as

waypoints K and L in Fig. 4(a). As depicted in Fig. 4(b), the corner points are transformed into convex points, with corner points B and C being converted into convex point N, and points K and L being replaced by point O. First, find the feature point pairs like B, C among the feature points, namely, points B and C are diagonally adjacent and the other two points sandwiched in the middle of B, C one is collision-free and the other is the obstacle. Satisfying the above conditions, we find a generation condition for corner points, and we mark the corresponding collision-free point as a corner point. Fig. 4(b) demonstrates that the proposed transformation technique effectively avoids collisions, thereby facilitating the generation of safer smooth trajectories. In addition, the waypoint set Ω_1 undergoes a transformation to yield set $\Omega_2\{P_S, A, N, H, J, O, M, P_T\}$, resulting in a further reduction in the number of waypoints.

3) Removing Redundant Points (RRP): A^* algorithm searches only in the eight surrounding directions, leading to the generation of partially redundant waypoints. As illustrated in Fig. 4(c), according to waypoint set Ω_2, the trajectory must sequentially traverse P_S, A, N, H, J, O, M, and P_T. It is evident that the vector $\overrightarrow{NJ}$ does not intersect with obstacles. The intermediate point H is redundant for trajectory generation. By judging vector $\overrightarrow{NO}$ is collision-free, point J is marked as a redundant point. In a similar vein, the waypoints in point set Ω_2 are filtered to remove redundancies, ultimately yielding the final point set, Ω_f.

The final planned trajectory can be fitted using the point set Ω_f, as demonstrated in Fig. 4(d). Compared to the traditional A^* algorithm, the use of filtered waypoints leads to a reduction in the number of turns and promotes effective trajectory smoothing.

E. Safe Trajectory Smoothing

After obtaining the final point set Ω_f as described above, a smooth reference trajectory is generated using trajectory optimization in this section. Moreover, the reference trajectory is ensured to be safe and collision-free.

1) Optimization-Based Polynomial Planning Approach: An optimization-based polynomial approach is employed to achieve smooth trajectory generation. The planned reference trajectory can be represented as a quintic polynomial [36] with respect to τ in the x and y directions. Namely, the reference trajectory can be formulated as

$$\begin{cases} x_r(\tau) = \delta_0 + \delta_1\tau + \delta_2\tau^2 + \delta_3\tau^3 + \delta_4\tau^4 + \delta_5\tau^5 \\ y_r(\tau) = \mu_0 + \mu_1\tau + \mu_2\tau^2 + \mu_3\tau^3 + \mu_4\tau^4 + \mu_5\tau^5 \end{cases} \tag{9}$$

which can be rewritten in matrix form as follows:

$$\begin{cases} x_r(\tau) = \nabla_{(\tau)} \cdot \delta_r \\ y_r(\tau) = \nabla_{(\tau)} \cdot \mu_r \end{cases} \tag{10}$$

where $\delta_r = [\delta_0, \delta_1, \delta_2, \delta_3, \delta_4, \delta_5]^T$ and $\mu_r = [\mu_0, \mu_1, \mu_2, \mu_3, \mu_4, \mu_5]^T$ are coefficient vectors, $\nabla_{(*)} := [1, *, *^2, *^3, *^4, *^5]$ with its derivatives $\dot{\nabla}_{(\dot{*})} := [0, 1, 2*, 3*^2, 4*^3, 5*^4]$ and $\ddot{\nabla}_{(\ddot{*})} := [0, 0, 2, 6*, 12*^2, 20*^3]$ is a simplified vector expression. This paper considers a time-dependent reference trajectory, namely $\tau = t$. The corresponding velocity v and acceleration σ can be expressed as

$$\begin{cases} v_{xr}(\tau) = \dot{x}_r(\tau) = \dot{\nabla}_{(\dot{\tau})} \cdot \delta_r \\ v_{yr}(\tau) = \dot{y}_r(\tau) = \dot{\nabla}_{(\dot{\tau})} \cdot \mu_r \end{cases} \tag{11}$$

$$\begin{cases} \sigma_{xr}(\tau) = \ddot{x}_r(\tau) = \ddot{\nabla}_{(\ddot{\tau})} \cdot \delta_r \\ \sigma_{yr}(\tau) = \ddot{y}_r(\tau) = \ddot{\nabla}_{(\ddot{\tau})} \cdot \mu_r \end{cases} \tag{12}$$

The trajectory from the starting point P_S to the target point P_T comprises k sub-trajectories. The i-th, $i = 1, \ldots, k$ sub-trajectory can be written as

$$\begin{cases} x_{ir}(\tau) = \nabla_{(\tau)} \cdot \delta_{ir}, \tau_{i-1} \le \tau < \tau_i \\ y_{ir}(\tau) = \nabla_{(\tau)} \cdot \mu_{ir}, \tau_{i-1} \le \tau < \tau_i \end{cases}$$

where τ_{i-1} and τ_i denote the start and end moments of the i-th sub-trajectory, respectively. τ_0 refers to the initial moment, while τ_k denotes its final moment. $T = \tau_k - \tau_0$ is the total duration.

The coefficient matrices Θ_r and Λ_r, which describe the reference trajectory, can be defined as

$$\Theta_r = [\delta_{1r}^T, \delta_{2r}^T, \ldots, \delta_{kr}^T]^T,\ \Lambda_r = [\mu_{1r}^T, \mu_{2r}^T, \ldots, \mu_{kr}^T]^T \tag{13}$$

The trajectory smoothing problem can be reformulated as a search for Θ_r and Λ_r such that

- The trajectory is safe and collision-free;
- The trajectory exhibit continuity and possess second-order differentiability between two adjacent sub-trajectories;
- The trajectory intersects with the waypoints in the set Ω_f;
- To enhance comfort, the second-order derivative of the trajectory is minimized.

For clarity, the optimization problem is presented for illustrative purposes in the x-direction, and likewise in the y-direction. The trajectory optimization problem can be converted into a quadratic programming problem with equation constraints as

$$\min f(\Theta_r) = \min\left(\ddot{x}(\tau)\right)^2, s.t. \mathcal{A}_{(4k+2)\times 6k}\Theta_r = \xi \tag{14}$$

where the constraint determined by $\mathcal{A}_{(4k+2)\times 6k}$ and ξ is discussed below. By substituting (12), we can get

$$\begin{aligned} \min\left(\ddot{x}(\tau)\right)^2 &= \min \int_{\tau_0}^{\tau_k} \left(\ddot{x}(\tau)\right)^2 d\tau = \min \sum_{i=1}^{k} \int_{\tau_{i-1}}^{\tau_t} \left(\ddot{x}(\tau)\right)^2 d\tau \\ &= \min \sum_{i=1}^{k} {\Theta_{ir}}^T \mathcal{Q}_i \Theta_{ir} = \min {\Theta_r}^T \mathcal{Q} \Theta_r \end{aligned} \tag{15}$$

where $\mathcal{Q}$ =diag($[\mathcal{Q}_1, \mathcal{Q}_2, \ldots, \mathcal{Q}_k]$) and

$$\mathcal{Q}_i = \int_{\tau_{i-1}}^{\tau_i} \ddot{\nabla}_{(\ddot{\tau})}^T \times \ddot{\nabla}_{(\ddot{\tau})}\, \mathrm{d}\tau = \begin{bmatrix} 0_{2\times 2} & 0_{2\times 4} \\ 0_{4\times 2} & \mathcal{N}_{4\times 4} \end{bmatrix} \tag{16}$$

The $(\lambda+1)$-th row and $(\varrho+1)$-th column element in $\mathcal{Q}_i$ are equal to $\frac{\lambda(\lambda-1)\cdot\varrho(\varrho-1)}{(\lambda-2)+(\lambda-2)+1}\left(\tau_i^{(\lambda+\varrho-3)} - \tau_{i-1}^{(\lambda+\varrho-3)}\right)$. Subsequently, the matrix $\mathcal{N}_{4\times 4}$ can be obtained.

The constraints on position, velocity, and acceleration at P_S and P_T can be formulated as

$$[\nabla_{(\tau_0)}, 0_{6(k-1)}]\Theta_r = x_S,\ \ [0_{6(k-1)}, \nabla_{(\tau_k)}]\Theta_r = x_T \tag{17}$$

$$[\dot{\nabla}_{(\dot{\tau}_0)}, 0_{6(k-1)}]\Theta_r = v_{xS},\ [0_{6(k-1)}, \dot{\nabla}_{(\dot{\tau}_k)}]\Theta_r = v_{xT} \tag{18}$$

$$[\ddot{\nabla}_{(\ddot{\tau}_0)}, 0_{6(k-1)}]\Theta_r = a_{xS},\ [0_{6(k-1)}, \ddot{\nabla}_{(\ddot{\tau}_k)}]\Theta_r = a_{xT} \tag{19}$$

where x_S, v_{xS}, and a_{xS} represent the position, velocity, and acceleration, respectively, of P_S in the x direction, while x_T, v_{xT}, and a_{xT} are the corresponding variables of P_T. The user can assign these parameters based on their specific requirements. The trajectory needs to pass through the i-th waypoint (x_i, y_i)

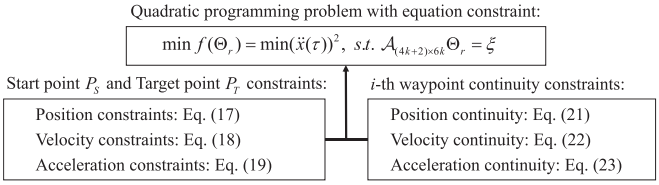


Fig. 6. Trajectory constraint.

in the set Ω_f, namely,

$$[0_{6(i-1)}, \nabla_{(\tau_i)}, 0_{6(k-i-1)}]\Theta_r = x_i \tag{20}$$

To ensure continuity of the reference trajectory at the i-th waypoint in the set Ω_f, we consider the following constraints

$$[0_{6(i-1)}, \nabla_{(\tau_i)}, -\nabla_{(\tau_i)}, 0_{6(k-i-1)}]\Theta_r = 0 \tag{21}$$

$$[0_{6(i-1)}, \dot{\nabla}_{(\dot{\tau}_i)}, -\dot{\nabla}_{(\dot{\tau}_i)}, 0_{6(k-i-1)}]\Theta_r = 0 \tag{22}$$

$$[0_{6(i-1)}, \ddot{\nabla}_{(\ddot{\tau}_i)}, -\ddot{\nabla}_{(\ddot{\tau}_i)}, 0_{6(k-i-1)}]\Theta_r = 0 \tag{23}$$

To clarify, the trajectory constraint considered and handled to generate the final planned trajectory from the obtained waypoints set Ω_f is illustrated in Fig. 6. In summary, we can obtain

$$\mathcal{A} = \begin{bmatrix} \nabla_{(\tau_0)}, 0_{6(k-1)} \\ \dot{\nabla}_{(\dot{\tau}_0)}, 0_{6(k-1)} \\ \ddot{\nabla}_{(\ddot{\tau}_0)}, 0_{6(k-1)} \\ \vdots \\ 0_{6(i-1)}, \nabla_{(\tau_i)}, 0_{6(k-i-1)} \\ \vdots \\ 0_{6(k-1)}, \nabla_{(\tau_k)} \\ 0_{6(k-1)}, \dot{\nabla}_{(\dot{\tau}_k)} \\ 0_{6(k-1)}, \ddot{\nabla}_{(\ddot{\tau}_k)} \\ 0_{6(i-1)}, \nabla_{(\tau_i)}, -\nabla_{(\tau_i)}, 0_{6(k-i-1)} \\ 0_{6(i-1)}, \dot{\nabla}_{(\dot{\tau}_i)}, -\dot{\nabla}_{(\dot{\tau}_i)}, 0_{6(k-i-1)} \\ 0_{6(i-1)}, \ddot{\nabla}_{(\ddot{\tau}_i)}, -\ddot{\nabla}_{(\ddot{\tau}_i)}, 0_{6(k-i-1)} \end{bmatrix}$$

$$\xi = [x_S, v_{xS}, a_{xS}, \ldots, x_i, \ldots, x_T, v_{xT}, a_{xT}, \ldots, 0]^T$$

Then, the planned trajectory can be obtained by solving the equation-constrained optimization problem (14).

2) Anti-Collision Modification: The trajectory generated using the waypoint set Ω_f still poses a risk of collision, as shown in Fig. 7(a). An anti-collision modification approach is devised, which is delineated in this section. The procedures are graphically depicted in Fig. 7. Starting from P_S, the planned trajectory undergoes a thorough collision scan to detect any potential obstacles along its path. Upon collision, the trajectory undergoes a modification process. In this context, the collision location is denoted by point C, as illustrated in Fig. 7(b). The waypoints $P1$ and $P2$, within the set Ω_f on both sides of C, are located using the location information. Connect $P1$ and $P2$ to obtain the line segment marked as L. Draw a perpendicular line from point C to line L to find the perpendicular point refer as D. Let the length of the perpendicular $\overline{CD}$ be equal to $4r$. Consider a point E that lies in the opposite direction of $\overline{CD}$ and is situated such that the length of $\overline{DE}$ is equal to $2r$. It is necessary to check whether there exists any obstacle at point E. If no collision occurs, it indicates that the modification process

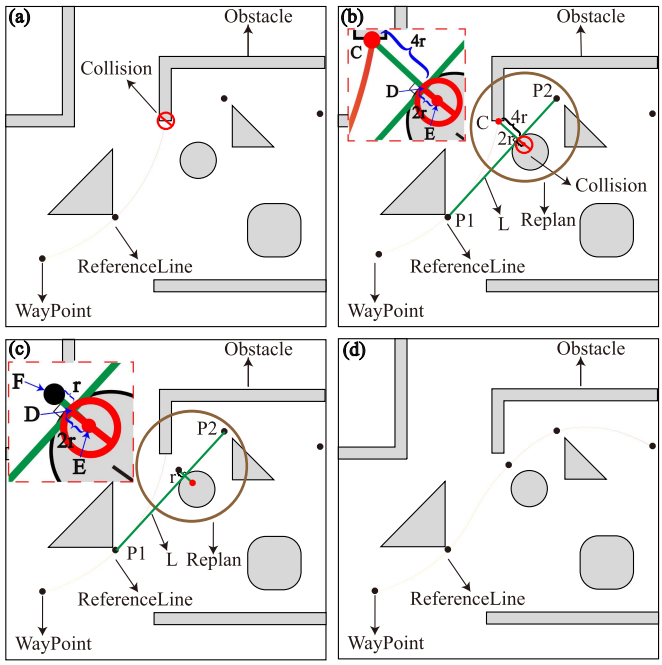


Fig. 7. Modification operation to ensure collision avoidance.

is complete. Point E can then be added between $P1$ and $P2$, to regenerate the planned trajectory. If a collision persists, point F can be found in the opposite direction of point E such that $\overline{FD} = r$, as plotted in Fig. 7(c). Continuing with the process, evaluate whether there is an obstacle at point F. Using a comparable procedure, modifications are executed for collision points within the generated trajectory until collision-free is achieved, as demonstrated in Fig. 7(d).

Remark 1: The proposed trajectory planning method can minimize the accelerations of the planned trajectory in the x and y directions, which contributes to enhancing longitudinal comfort. Moreover, the trajectory is safe and collision-free by employing anti-collision modification in the event of actuator faults and inaccurate tracking.

IV. Adaptive Reliable Controller Design

By analyzing the motion relationship shown in Fig. 1 and (1), we can derive that

$$\dot{d}_e = -\cos(\varphi_e)\, v + \left(\frac{x_e}{d_e}\dot{x}_r + \frac{y_e}{d_e}\dot{y}_r\right) \tag{24}$$

$$\dot{\varphi}_e = w + \frac{y_e}{d_e^2}\left(\dot{x}_r - v\cos(\varphi)\right) - \frac{x_e}{d_e^2}\left(\dot{y}_r - v\sin(\varphi)\right) \tag{25}$$

To ensure that the system remains collision-free even under inaccurate tracking, it is necessary to satisfy $d_e < A_e$, which is determined by the movement restraint size S_e. Meanwhile, it is essential to satisfy $d_e(t) > \varkappa$, $-\frac{\pi}{2} < \varphi_e(t) < \frac{\pi}{2}$ with $\varkappa \in \mathbb{R}^+$ to ensure the correct error definition. Therefore, we introduce invertible transformation functions as follows

$$\mathcal{G}(\zeta) = \frac{\upsilon_1 e^{(\varepsilon_1\zeta+\kappa_1)} + \varsigma_1 e^{-(\varepsilon_1\zeta+\kappa_1)}}{e^{(\varepsilon_1\zeta+\kappa_1)} + e^{-(\varepsilon_1\zeta+\kappa_1)}} \tag{26}$$

$$\mathcal{H}(\vartheta)=\frac{\upsilon_2 e^{(\varepsilon_2\vartheta+\kappa_2)}-\varsigma_2 e^{-(\varepsilon_2\vartheta+\kappa_2)}}{e^{(\varepsilon_2\vartheta+\kappa_2)}+e^{-(\varepsilon_2\vartheta+\kappa_2)}} \tag{27}$$

where $\kappa_1=\frac{1}{2}\ln\left(\frac{\theta}{\upsilon_1-\varsigma_1-\theta}\right)$, $\kappa_2=\frac{1}{2}\ln\frac{\varsigma_2}{\upsilon_2}$, $\varsigma_1,\varsigma_2,\upsilon_1,\upsilon_2,\varepsilon_1,\varepsilon_2$ and θ are positive real numbers subject to $\varsigma_1+\theta<\upsilon_1$.

Meanwhile, the following monotonic function is introduced

$$\psi(t)=(\psi_0-\psi_\infty)\,e^{-\iota t}+\psi_\infty \tag{28}$$

where $\iota>0$ and $\psi_0>\psi_\infty>0$ can be selected by the user.

Using the invertible transformation functions given in (26) and (27), we can express the transformed errors ζ and ϑ as

$$\begin{aligned}\zeta&=\mathcal{G}^{-1}(\omega_1(t))\\&=\frac{1}{2\varepsilon_1}\left(\ln\left(\omega_1(t)-\varsigma_1\right)+\ln\left(\upsilon_1-\varsigma_1-\theta\right)\right.\\&\quad\left.-\ln\left(\theta\upsilon_1-\theta\omega_1(t)\right)\right)\end{aligned} \tag{29}$$

$$\begin{aligned}\vartheta&=\mathcal{H}^{-1}\left(\omega_2(t)\right)\\&=\frac{1}{2\varepsilon_2}\left(\ln\left(\omega_2(t)\upsilon_2+\varsigma_2\upsilon_2\right)-\ln\left(\varsigma_2\upsilon_2-\omega_2(t)\varsigma_2\right)\right)\end{aligned} \tag{30}$$

where the superscript -1 denotes the inverse function, $\omega_1(t)=\frac{d_e(t)}{\psi(t)}$, and $\omega_2(t)=\frac{\varphi_e(t)}{\psi(t)}$.

Correspondingly, the derivatives of ζ and ϑ are written as

$$\dot{\zeta}=\epsilon_1(-\cos\left(\varphi_e\right)v+(\frac{x_e}{d_e}\dot{x}_r+\frac{y_e}{d_e}\dot{y}_r)-\frac{d_e\dot{\psi}}{\psi})$$

$$\dot{\vartheta}=\epsilon_2\left(w+\frac{y_e}{d_e^2}\left(\dot{x}_r-v\cos(\varphi)\right)-\frac{x_e}{d_e^2}(\dot{y}_r-v\sin(\varphi))-\frac{\varphi_e\dot{\psi}}{\psi}\right)$$

where $\epsilon_1=\frac{1}{2\varepsilon_1\psi}\left(\frac{1}{\omega_1(t)-\varsigma_1}-\frac{1}{\omega_1(t)-\upsilon_1}\right)$, $\epsilon_2=\frac{1}{2\varepsilon_2\psi}\left(\frac{1}{\omega_2(t)+\varsigma_2}-\frac{1}{\omega_2(t)-\upsilon_2}\right)$.

The following adaptive reliable controllers are constructed

$$v=\hat{\beta}_1\nu_1,\ \varpi=\hat{\beta}_2\nu_2 \tag{31}$$

$$\nu_1=-\frac{1}{\Gamma_1}\left(\hat{\beta}_{1_b}\Gamma_1+\Upsilon_1+m_1\zeta\right) \tag{32}$$

$$\nu_2=-\frac{1}{\varepsilon_2}\left(\Upsilon_2\hat{\beta}_{1_b}+\Upsilon_2\nu_1+\hat{\beta}_{2_b}\varepsilon_2+\Upsilon_3+m_2\vartheta\right) \tag{33}$$

where $\beta_1:=\frac{1}{\alpha_{1_l}}$, $\beta_2:=\frac{1}{\alpha_{2_l}}$, $\beta_{1_b}:=\alpha_{1_b}$, $\beta_{2_b}:=\alpha_{2_b}$, and $\hat{\circ}$ denotes the estimation of $\circ$. The adaptive laws are given by

$$\dot{\hat{\beta}}_1=-\frac{1}{k_1}\left(\zeta\Gamma_1\nu_1+\vartheta\Upsilon_2\nu_1\right)-k_{\sigma_1}\hat{\beta}_1 \tag{34}$$

$$\dot{\hat{\beta}}_{1_b}=\frac{1}{k_2}\left(\zeta\Gamma_1+\vartheta\Upsilon_2\right)-k_{\sigma_2}\hat{\beta}_{1_b} \tag{35}$$

$$\dot{\hat{\beta}}_2=-\frac{1}{k_3}\left(\vartheta\epsilon_2\nu_2\right)-k_{\sigma_3}\hat{\beta}_2 \tag{36}$$

$$\dot{\hat{\beta}}_{2_b}=\frac{1}{k_4}\left(\vartheta\epsilon_2\right)-k_{\sigma_4}\hat{\beta}_{2_b} \tag{37}$$

where $k_i\in\mathbb{R}^+, i=1,2,3,4$, and $\Upsilon_1=\epsilon_1(\frac{x_e}{d_e}\dot{x}_r+\frac{y_e}{d_e}\dot{y}_r)-\epsilon_1\frac{d_e\dot{\psi}}{\psi}$, $\Upsilon_2=-\epsilon_2\frac{y_e}{d_e^2}\cos(\varphi)+\epsilon_2\frac{x_e}{d_e^2}\sin(\varphi)$, $\Upsilon_3=\epsilon_2\frac{y_e}{d_e^2}\dot{x}_r-\epsilon_2\frac{x_e}{d_e^2}\dot{y}_r-\epsilon_2\frac{\varphi_e\dot{\psi}}{\psi}$, $\Gamma_1=-\epsilon_1\cos(\varphi_e)$.

Theorem 1: By utilizing the designed controller given in (31)-(33) and the adaptive laws described in (34)–(37), we can ensure the stability of the system even with actuator faults. And the tracking errors satisfy

$$\varsigma_1\psi(t)<d_e(t)<\upsilon_1\psi(t),\ -\varsigma_2\psi(t)<\varphi_e(t)<\upsilon_2\psi(t) \tag{38}$$

for any $t>0$. By choosing parameters ς_1, ς_2, υ_2, ψ_0 and ψ_∞ subject to $\varsigma_1\geq\frac{\Phi}{\psi_\infty}$, $\varsigma_2\leq\frac{\pi}{2\psi_\infty}$, $\upsilon_2\leq\frac{\pi}{2\psi_0}$, we can obtain

$$d_e(t)>\Phi,\ -\frac{\pi}{2}<\varphi_e(t)<\frac{\pi}{2} \tag{39}$$

where Φ is a positive constant. Meanwhile, the controller parameters are adjusted to satisfy $\upsilon_1\psi_0<A_e$, ensuring that the system remains collision-free even in the presence of actuator failures and tracking errors. In summary, the motion planning and control framework guarantees the safety and reliability of the system.

Proof: The following Lyapunov candidate function V is chosen as $V=\frac{\zeta^2}{2}+\frac{\vartheta^2}{2}+\frac{k_1}{2\beta_1}\tilde{\beta}_1^2+\frac{k_2}{2}\tilde{\beta}_{1_b}^2+\frac{k_3}{2\beta_2}\tilde{\beta}_2^2+\frac{k_4}{2}\tilde{\beta}_{2_b}^2$, where $\tilde{*}:=\hat{*}-*$ denotes the estimation error of $*$.

The derivative of V can be derived as follows

$$\dot{V}=\zeta\dot{\zeta}+\frac{k_1\tilde{\beta}_1\dot{\hat{\beta}}_1}{\beta_1}+k_2\tilde{\beta}_{1_b}\dot{\hat{\beta}}_{1_b}+\vartheta\dot{\vartheta}+\frac{k_3\tilde{\beta}_2\dot{\hat{\beta}}_2}{\beta_2}+k_4\tilde{\beta}_{2_b}\dot{\hat{\beta}}_{2_b}$$

Substituting the designed controller (31)–(33) and the adaptive laws (34)–(37), we can obtain

$$\begin{aligned}\dot{V}\leq&-m_1\zeta^2-\frac{k_1k_{\sigma_1}}{2\beta_1}\tilde{\beta}_1^2-\frac{k_2k_{\sigma_2}}{2}\tilde{\beta}_{1_b}^2\\&-m_2\vartheta^2-\frac{k_3k_{\sigma_3}}{2\beta_2}\tilde{\beta}_2^2-\frac{k_4k_{\sigma_4}}{2}\tilde{\beta}_{2_b}^2+C_\sigma\\\leq&-m_\sigma V+C_\sigma\end{aligned} \tag{40}$$

where $m_\sigma=\min(2m_1,2m_2,k_{\sigma_1},k_{\sigma_2},k_{\sigma_3},k_{\sigma_4})$, $C_\sigma=\frac{k_1k_{\sigma_1}}{2}\beta_1+\frac{k_{\sigma_2}k_2}{2}\beta_{1_b}^2+\frac{k_{\sigma_3}k_3}{2}\beta_2+\frac{k_{\sigma_4}k_4}{2}\beta_{2_b}^2$.

Thus, the system is ultimately uniformly bounded [39], [40], and ζ and ϑ are bounded and accordingly, it is easy to obtain that $d_e(t)\in(\varsigma_1\psi(t),\upsilon_1\psi(t))$ and $\varphi_e(t)\in(-\varsigma_2\psi(t),\upsilon_2\psi(t))$. By choosing the parameters as in *Theorem 1*, we can obtain $d_e(t)>\Phi$ and $\varphi_e(t)\in(-\frac{\pi}{2},\frac{\pi}{2})$. Since we take into account the tracking error margin A_e in the movement restraint size S_e, the system can be made safe and collision-free by adjusting the parameters so that $\upsilon_1\psi_0<A_e$ holds. The proof of *Theorem 1* is complete.

Remark 2: The controller developed in this paper can enhance the system reliability by addressing actuator faults, including actuator loss-of-effectiveness faults and bias faults. By considering the tracking error caused by inaccurate tracking in the controller design, the dynamic response can be constrained to safe performance boundaries to ensure system safety. The designed control method can be easily applied to autonomous surface vehicles, mobile robots, etc. To enhance the proposed method, future work focuses on improving the system safety in

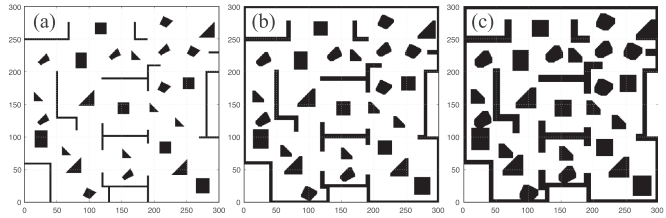


Fig. 8. Region division corresponds to different S_e.

TABLE I
MOTION PLANNING PARAMETER SETTINGS

Symbol	Definition	Value(Unit)
W	Robot size	2.8 cell
A_e	Tracking error margin	0.2 cell
w_1	Weighting factor	1
w_2	Weighting factor	1
w_3	Weighting factor	1
S_e	Motion restraint size	3 cell
η_{re}	Maximum repulsive coefficient	30
ρ_0	Maximum repulsive range	8 cell
a_{re}	Decay coefficient	0.003
T	Total duration	30 s

dynamic scenarios and reducing the traveling delay and energy consumption to further improve the system performance.

Remark 3: By employing the grid map pre-processing method in Section III-A, the map is pre-processed first. The corresponding repulsive potential field is generated with designed motion restraint size S_e, and then the generated grid map is divided into obstacle regions and safe collision-free regions. The initial collision-free candidate point set is then generated using the A^* algorithm, and the final point set Ω_f is obtained using the waypoint set filtering method proposed in Section III-D. Using the trajectory smoothing and anti-collision modification method in Section III-E, the planned smooth and safe trajectory can be obtained. The reliable controller designed in Section IV is used to achieve accurate tracking while enhancing the reliability of autonomous vehicles.

V. SIMULATION RESULTS

In this section, a complex indoor simulation scenario comprising multiple irregular obstacles is constructed to validate the effectiveness of the designed planning and control method. It is assumed that the map information is available through sensors.

To achieve collision-free motion planning, the map information is first pre-processed using the proposed map grid method. Combined with the repulsive potential field, the region division with different S_e is provided in Fig. 8. The S_e in Fig. 8(a), (b), and (c) are chosen as $S_e = 1$, $S_e = 3$, and $S_e = 5$, respectively. It can be seen that a larger S_e, caused by a larger tracking error margin or robot size W, results in a smaller safety zone, which is consistent with the actual situation. Then, the waypoint point set Ω_f can be generated using the presented waypoint set filtering strategy. For clarity, the parameter settings of motion planning are provided and summarized in Table I.

Specifically, the A^* algorithm is used to search the map space to find a feasible global path from P_S to the P_T. The points involved and searched are represented as *AStarExtensionPoints* in Fig. 9, and the final generated path Ω_0 is indicated as *AStarPoints*. Note that the *FeaturePoints* in Fig. 9 are also

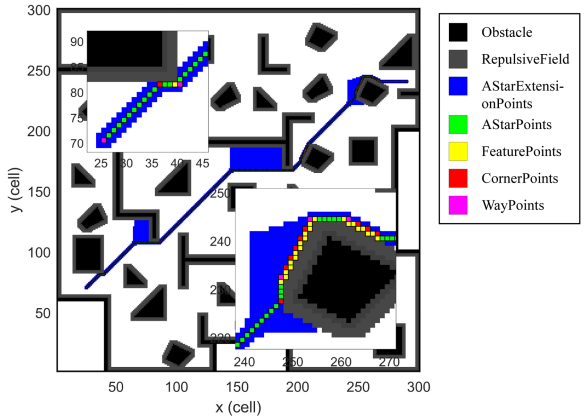
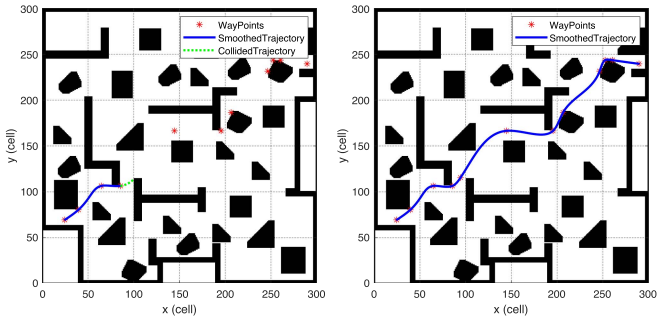


Fig. 9. Waypoint set Ω_f generation.

TABLE II
BENEFITS OF WAYPOINT SET FILTERING

Algorithm	Running time(s)	Path length(cell)	Number of nodes	Number of turns
Dijkstra	35.327834	332	263	39
A^*	1.346724	351	284	43
A^*2SFP	1.364578	321	45	43
A^*2TCP	2.213784	319	28	26
A^*2RRP	2.634227	303	11	9



(a) Collision phenomena. (b) Safe trajectory.

Fig. 10. Collision avoidance by the proposed anti-collision modification.

part of the original set of *AStarPoints*. By selecting feature points as described above, the set Ω_1 is generated, namely the *FeaturePoints* in Fig. 9. The points in set Ω_1 are then judged and transformed to produce the corner point set Ω_2, which are denoted by the *CornerPoints* in Fig. 9. The redundant points that do not contain valuable path information are removed to obtain the final set Ω_f, namely *WayPoints* in Fig. 9.

Meanwhile, quantitative results of Dijkstra, A^*, A^*2SFP (A^*+ SFP), A^*2TCP (A^*+ SFP + TCP), and A^* 2RRP (A^*+ SFP + TCP + RRP) are provided in Table II to evaluate the proposed waypoint set filtering method. From Table II, by combining with the designed waypoint set filtering strategy, the generated planned path obtains a shorter path length and fewer nodes. Moreover, the number of path turns is reduced, which contributes to generating and smoothing the planned trajectory. Based on the proposed safe trajectory smoothing method, the planned trajectory is obtained by fitting the waypoints in set Ω_f.

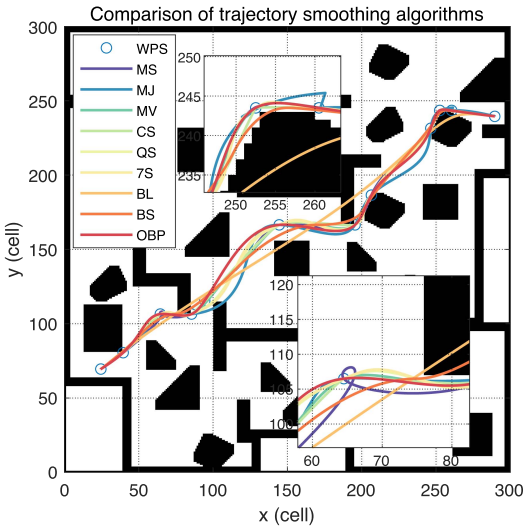


Fig. 11. Comparison with classical trajectory smoothing algorithms.

As shown in Fig. 10(a), collision phenomena occur when the planned trajectory is fitted directly using the final point set Ω_f. By modifying waypoint points using the proposed anti-collision modification, safe trajectory smoothing can enable collision-free trajectory fitting in complex environments, as in Fig. 10(b). The safety can be effectively enhanced by the proposed safe trajectory planning method.

Comparative simulation with the existing classical method is added to highlight the advantages of the presented method, as shown in Fig. 11. The classical Minimum Snap (MS) algorithm, the Minimum Jerk (MJ) algorithm, and the Minimum Velocity (MV) algorithm can minimize the 4th-order derivative, 3rd-order derivative, and 1st-order derivative of the planned trajectory, respectively. Trajectory generation is also performed and provided in Fig. 11 using the Cubic Spline (CS) algorithm, the Quintic Spline (QS) algorithm, the 7th Spline (7S) algorithm, the Bezier (BL) algorithm, the B-spline (BS) algorithm, and the proposed Optimization Based Polynomials(OBP) with anti-collision modification algorithm. And the WPS denote waypoints.

As depicted in Fig. 11, the collision occurred for the planned trajectory fitted by using the polynomial-like algorithms (CS, QS, 7S) based on the final point set Af, which means that the safety of the planned trajectory is difficult to guarantee for the complex scenario. MJ without anti-collision modification, BL and BS algorithms are more focused on the smoothness of the planned trajectory, making the trajectory appear frequent collisions with obstacles and making it may be unsafe. Optimization-based methods (MS, MV, OBP) with the proposed anti-collision modification can generate safe trajectories even in complex scenes.

Note that the MS and MJ optimize the higher order derivative of the planned trajectory than the OBP, with correspondingly increased constraints, making it easy to produce unreasonable trajectories in order to meet the stricter constraints. Trajectory spins of MS and MJ can be observed in the enlarged part, which is not reasonable for autonomous vehicle due to its geometric properties. In addition, except for the OBP, the MV algorithm also avoid the problems of the MS and MJ algorithms above because it minimizes the first-order derivative. Nonetheless, its planned trajectory may be subject to large accelerations which reduce comfort.

TABLE III
PLANNING ALGORITHMS EVALUATION

Algorithm	Safety	Number of modifications	Path length(cell)	CCR
MS	Yes	1	353.7	0.1158
MJ	No	No modification	389.8	0.0977
MV	Yes	0	347.5	0.0043
CS	No	No modification	345.3	0.0089
QS	No	No modification	351	0.0047
7S	No	No modification	353.6	0.0053
BL	No	No modification	320.3	0.0007
BS	No	No modification	333.4	0.0021
OBP	Yes	1	350.9	0.0039

TABLE IV
COMPARISON OF TRAJECTORY PERFORMANCE

Algorithm (Continuity)	x			y		
	v_{xr}	σ_{xr}	$jerk_{xr}$	v_{yr}	σ_{yr}	$jerk_{yr}$
MS(2-th)	9.93	6.59	11.3	7.76	5.1	7.74
MJ(2-th)	10.03	5.59	8.53	7.72	4.06	5.09
MV(2-th)	9.37	5.15	14.43	7.25	4.87	13.97
CS(1-th)	9.42	13.35	40.18	7.42	9.55	16.68
QS(2-th)	9.12	2.93	16.3	7.39	4.84	22.76
7S(3-th)	9.15	3.5	20.19	7.52	5.73	28.2
BL(Any-th)	9.18	0.73	0.22	6.24	0.58	0.18
BS(1-th)	8.92	2.12	0	6.62	2.45	0
OBP(2-th)	9.42	4.8	10.34	7.45	4.26	7.75

Further, we evaluate quantitatively the planning performance of the different algorithms, and the comparative results are given in Tables III and IV. As provided in Table III, the polynomial-like algorithms (CS, QS, 7S), BL algorithm, BS algorithm, and the MJ without anti-collision modification can not ensure safety and avoid collisions with obstacles. CCR denotes the root mean square (RMS) value of the rate of change of curvature. The planned trajectory generated by OBP is reduced in terms of path length and the rate of change of trajectory curvature compared with MS and MJ. More importantly, by incorporating the anti-collision treatment, the optimization-based (MS, MV, OBP) algorithm can guarantee collision-free and safe.

The performance of the planned trajectory is quantified and compared in the x and y directions as provided in Table IV. i-th, $i = 1, 2, 3$ means that the trajectory is i-order differentiable. And any-th means that it is differentiable to any order. All values in Table IV are the RMS values of the corresponding variables. To improve comfort, we introduced the acceleration of planned trajectory as cost into the optimization in OBP. It can be seen that the RMS of σ_{xr} the OBP algorithm is the smallest compared to MS and MV. Because the CS, QS, 7S, BL, and BS algorithms focus more on smoothness, smaller acceleration can be achieved. However, these algorithms can not guarantee the corresponding trajectories are safe. In summary, the designed OBP algorithm can improve the comfort of the trajectory, by reducing the acceleration of the reference trajectory, and more importantly, it can guarantee the safety of the reference trajectory.

Finally, the final planning trajectory with the upper and lower boundaries generated by S_e is shown in Fig. 12. It can be observed that, by introducing the designed motion constraint size S_e and repulsive potential field, the generated planned trajectory has a safety margin against inaccurate tracking, which can guarantee the safety of the planned trajectory.

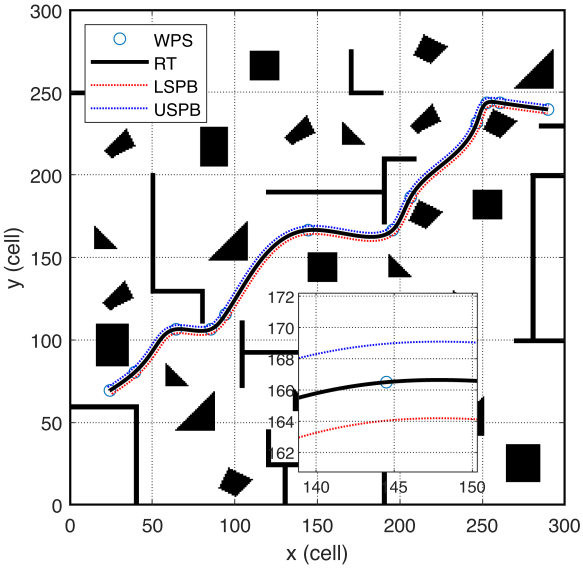


Fig. 12. Final trajectory with movement restraint size S_e. RT denotes the final reference trajectory. LSPB and USPB are the lower boundary and upper boundary dependent on S_e.

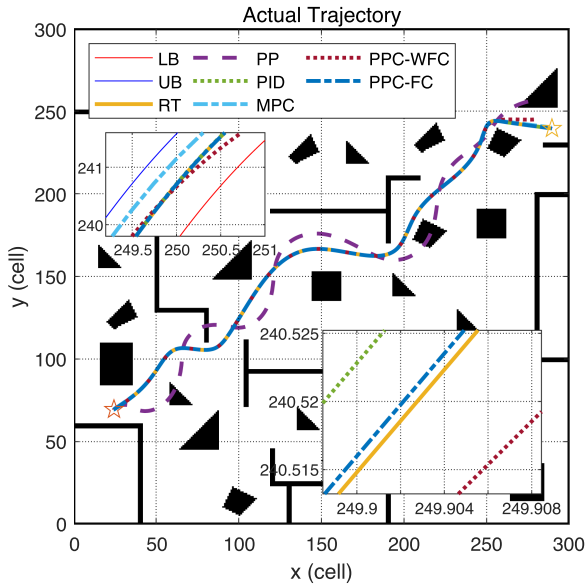


Fig. 13. Tracking performance with various control methods. LB and UB are the trajectory lower boundary and upper boundary. RT represents the reference trajectory.

To highlight and verify the tracking performance and reliability of the designed fault-tolerant controller, various control algorithms are implemented, including: 1) Pure tracking method (PP). 2) Proportional integral derivative (PID) controller. 3) Model predictive control (MPC) algorithm. 4) Prescribed performance-based control without fault tolerance (PPC-WFC). 5) Prescribed performance-based fault tolerant control (PPC-FC) with $\psi_0 = 0.2, \psi_\infty = 0.05, \iota = 2, \varsigma_1 = 0.1, \upsilon_1 = 1, \theta = 0.4, \quad \varsigma_2 = 5, \upsilon_2 = 5, \varepsilon_1 = 0.005, \varepsilon_2 = 0.005, k_1 = 100, k_2 = 100, k_3 = 100, k_4 = 100, m_1 = 100, m_2 = 0.1, k_{\sigma_1} = 0.01, k_{\sigma_2} = 0.01, k_{\sigma_3} = 0.01, k_{\sigma_4} = 0.01, \alpha_{1_l} = 0.8, \alpha_{1_b} = 0.1, \alpha_{2_l} = 0.8, \alpha_{2_b} = 0.1$. The actuator faults are set to when $t > 25$, $u_{if} = \alpha_{il} u_i + \alpha_{ib}$; otherwise $u_{if} = u_i, i = 1, 2$.

The corresponding tracking comparison results are given in Fig. 13. Note that the reference trajectory cannot be tracked completely accurately and tracking errors exist in practice. Although the planned trajectory is collision-free with the obstacle, there is still a risk of collision when the planned trajectory is tracked inaccurately, such as PP in Fig. 13. Most methods decouple motion planning from control and assume that the desired trajectory is accurately tracked to generate the planned trajectory, which in practice can still lead to unsafe collisions. This is the significance of this research.

To analyze the transient performance, the actual transient responses of controllers and the corresponding performance constraints are provided in Fig. 14. It is worth mentioning that the PP clearly does not satisfy the desired performance constraints. Meanwhile, PID, MPC, and PPC-WFC can achieve accurate tracking by tuning the parameters, however, the transient performance still may exceed the error constraints. Especially, when actuator faults occur, the negative effect of faults may bring the transient responses out of the error bound, such as MPC and PPC-WFC in Fig. 14.

In this case, the transient tracking error is possible to exceed A_e in the movement restraint size S_e, which indicates a potential collision risk. The safety of the corresponding system cannot

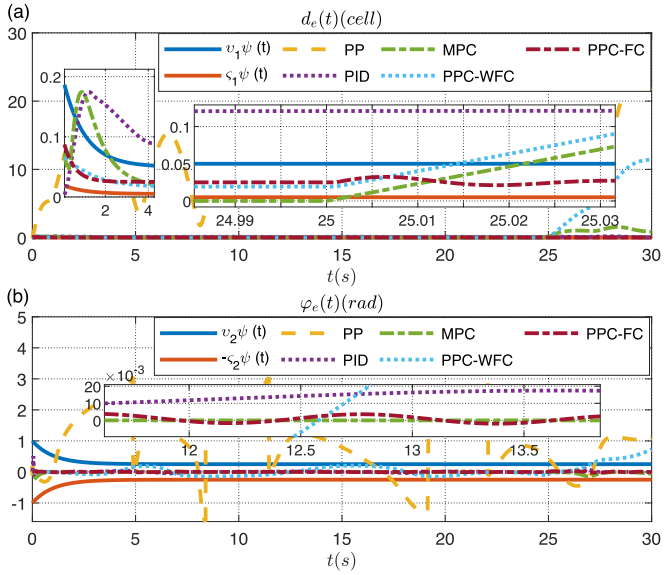


Fig. 14. Transient performance and prescribed performance constraints.

be guaranteed. The designed PPC-FC not only improves the reliability of the system but ensures its safety by combining the S_e in the planning layer.

To further visualize the tracking performance, comparative results of the tracking performance in the x and y directions are presented in Fig. 15(a) and (b), respectively. It can be observed that excellent trajectory tracking performance can be achieved for all controllers when no actuator faults occur. However, when actuator faults occur, the actual trajectories of MPC and PPC-WFC gradually deviate from the reference trajectory. In contrast, the fault-tolerant controller PPC-FC still maintains good tracking performance in case of failure, which significantly enhances the reliability of the system.

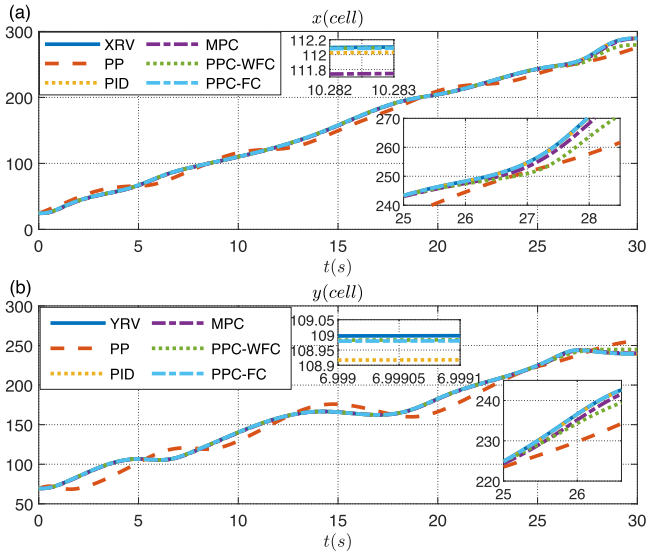


Fig. 15. Tracking performance in the x and y directions. XRV and YRV denote the reference values in the x and y directions, respectively.

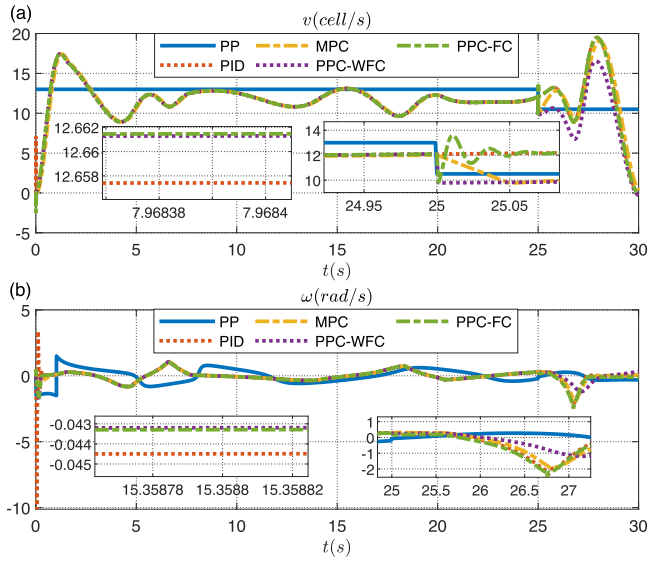


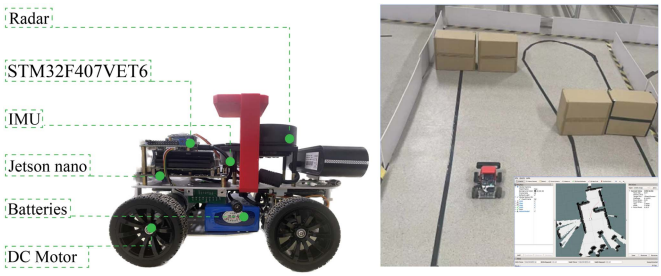
Fig. 16. Control inputs.

The corresponding control inputs are shown in Fig. 16. Note that the proposed prescribed performance fault-tolerant control method can reduce the negative effects caused by faults through adaptive adjustment. PPC-WFC cannot compensate for actuator failures, which impairs the system reliability and thus makes the overall performance degrade, as shown in Figs. 14 and 15, where the tracking performance deteriorates significantly.

The RMS values of tracking errors are given in Table V. It can be seen that PPC-FC still achieves excellent tracking performance in the presence of an actuator failure. Meanwhile, MPC needs to solve the optimization problem, which leads to larger computational requirements than PPC-FC. Overall, the designed controller improves reliability and achieves good tracking performance even in the event of actuator faults. Moreover, the computation requirements are greatly reduced because the optimization problem is avoided.

TABLE V
RUNNING TIME AND TRACKING ERRORS

Algorithm	Time(s)	x_e(cell)	y_e(cell)	d_e(cell)	φ_e(rad)
PP	2.256	6.242	9.593	11.445	1.508
PID	0.704	0.896	0.658	1.111	0.041
MPC	11.3	0.371	0.261	0.453	0.028
PPC-WFC	0.889	2.595	1.39	2.944	0.184
PPC-FC	0.532	0.021	0.017	0.027	0.008



(a) Autonomous vehicle. (b) Experiment site.

Fig. 17. Experimental setup.

VI. EXPERIMENTAL RESULTS

The autonomous vehicle hardware platform and the experimental site are shown in Fig. 17(a) and (b), respectively. The proposed algorithm is implemented in Jetson nano 4 GB with Ubuntu 18.04 and ROS Melodic system and deployed on the micro-autonomous vehicle illustrated in Fig. 17(a). The micro-vehicle platform [41] with the same kinematic properties as actual autonomous vehicles can be used to validate the effectiveness and advantages of the designed motion planning and control framework. By employing SLAMTEC RPLIDAR A1 and IMU sensors, the map information can be obtained using simultaneous localization and mapping (SLAM), and real-time localization information of the autonomous vehicle can be collected. Based on the acquired perception information, the proposed motion planning algorithm can generate a safe collision-free planned trajectory. The control inputs are calculated by using the developed control algorithm and sent via communication to the lower computer STM32F103VET6 to enable tracking of the planned reference trajectory. The sensors and hardware configuration of the autonomous vehicle are provided in Fig. 17(a). The experimental scenario and the corresponding map information are shown in Fig. 17(b).

Similar to the simulation scenario, PP, PID, PP-WFC, and PP-FC controllers are implemented to track the planned trajectory and analyze the system performance, where the parameters are set to $\psi_0 = 1, \psi_\infty = 0.1, \iota = 0.1, \varsigma_1 = 0.001, \upsilon_1 = 0.1, \theta = 0.08, \varsigma_2 = 1.1, \upsilon_2 = 1.1, \varepsilon_1 = 0.01, \varepsilon_2 = 0.01, k_1 = 100, k_2 = 100, k_3 = 100, k_4 = 100, m_1 = 0.01, m_2 = 0.01, k_{\sigma_1} = 0.01, k_{\sigma_2} = 0.01, k_{\sigma_3} = 0.01, k_{\sigma_4} = 0.01, \alpha_{1_l} = 0.9, \alpha_{1_b} = 0.021, \alpha_{2_l} = 0.9, \alpha_{2_b} = 0.021$.

The actuator faults are set to when $t > 2.5, u_{if} = \alpha_{il} u_i + \alpha_{ib}$; otherwise $u_{if} = u_i, i = 1, 2$. The comparative results of the trajectory tracking performance obtained by different controllers are plotted in Fig. 18. It can be observed that the reliability of PPC-WFC and PP decreases significantly as the occurrence of actuator faults (3), gradually exceeds the desired constraint boundaries. Compared to PPC-WFC, PPC-FC can adaptively

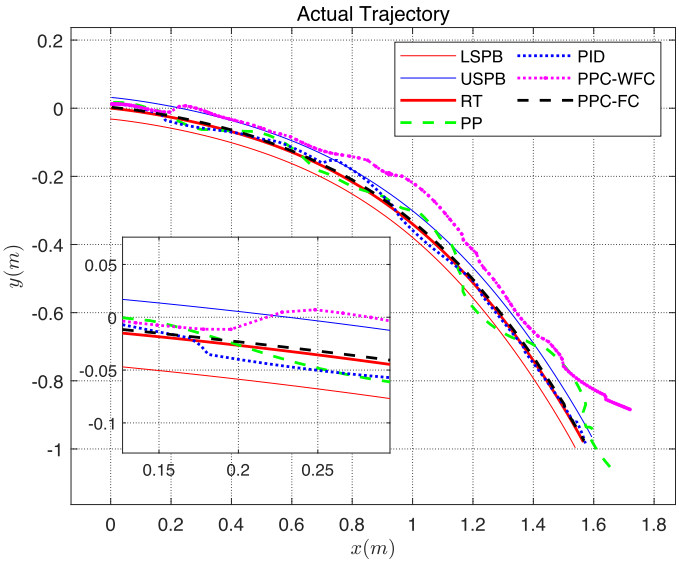


Fig. 18. Tracking performance comparative results. LB and UB are the trajectory lower boundary and upper boundary. RT represents the reference trajectory.

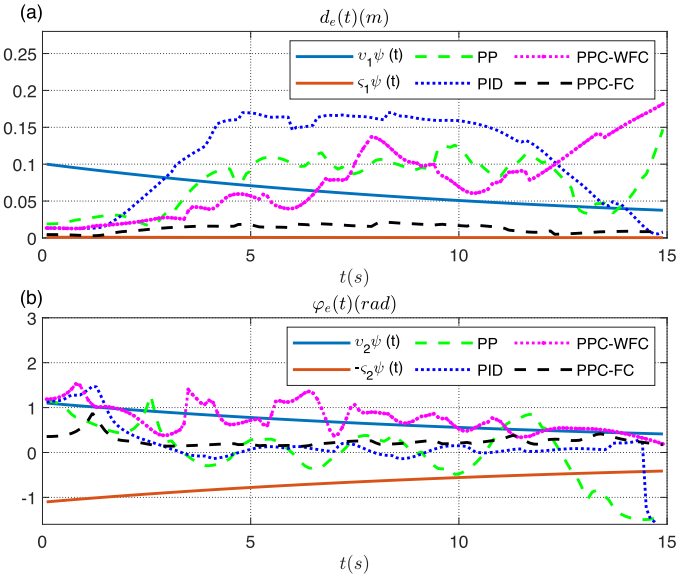


Fig. 19. Transient responses and performance constraints.

address actuator faults which significantly improves the reliability and tracking performance.

The transient responses of d_e, φ_e, and the prescribed performance constraints are illustrated in Fig. 19. PP, PPC-WFC, and PID without actuator faults can obtain satisfactory tracking performance by tuning controller parameters. However, after actuator faults occur, the system gradually deviates from the desired trajectory and the tracking performance and reliability of the system degrades significantly. It is clear that the PPC-FC designed in this paper can ensure the safety and reliability of the autonomous vehicle under inaccurate tracking, and in practical applications, users can select specific performance parameters to achieve the desired planning and control performance.

Meanwhile, to further analyze the tracking performance, comparative results in the x and y directions are shown in Fig. 20(a) and (b), respectively. After actuator failures occur, the trajectory tracking capability of the PPC-WFC decreases considerably. It

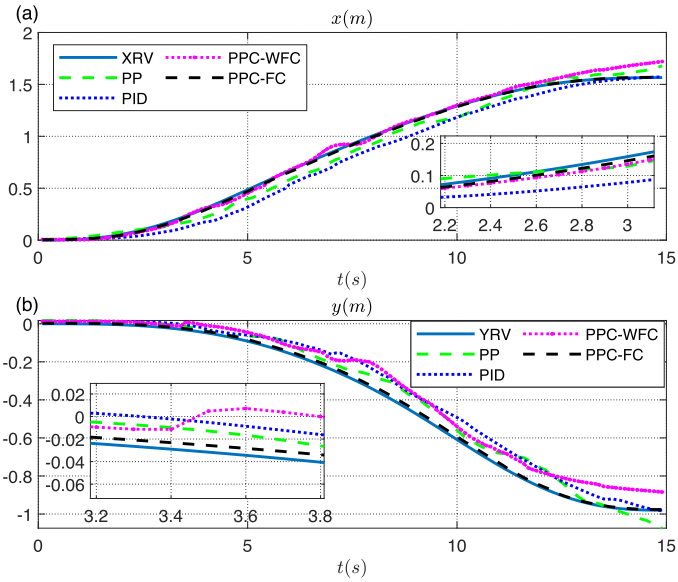


Fig. 20. Tracking performance in the x and y directions. XRV and YRV denote the reference values in the x and y directions, respectively.

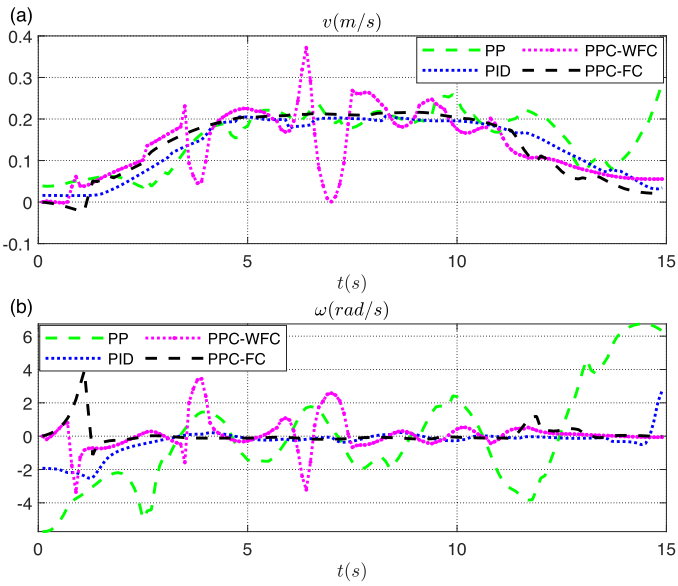


Fig. 21. Control inputs.

TABLE VI
EXPERIMENTAL RUNNING TIME AND TRACKING ERRORS

Algorithm	Time(s)	x_e(m)	y_e(m)	d_e(m)	φ_e(rad)
PP	19.894	0.0653	0.0521	0.0836	0.6487
PID	18.168	0.1015	0.0699	0.1232	0.5183
PPC-WFC	18.648	0.0474	0.0738	0.0877	0.8113
PPC-FC	18.592	0.009	0.010	0.0137	0.2816

is clear that the PPC-FC considering actuator fault-tolerance can significantly improve the reliability and tracking performance of the system.

The corresponding control inputs and quantitative results are provided in Fig. 21 and Table VI. By using the control inputs as in Fig. 21, it can be observed from Fig. 18 that the tracking performance of the controller is significantly improved with adaptive regulation. Compared with PID, PP, and PPC-WFC, it can be

seen that the transient responses are within the performance constraints, which is theoretically guaranteed and verified by the experimental results in Fig. 19. This means that not only is the tracking performance and reliability improved as in Table VI, but the safety of the system is ensured combined with motion restraint size S_e in the planning level.

VII. Conclusion

This paper presents a safe and reliable motion planning and control framework that can ensure the safety and reliability of autonomous vehicles in the presence of inaccurate tracking and actuator faults by coordinating the planning and control layers. The motion restraint size is introduced into the planning layer to leave a safe margin for inaccurate tracking, and the potential field based on S_e is constructed to divide safe regions. The original point set generated using A^* algorithm in the safe region is refined and modified by the proposed waypoint set filtering operation. Then safe planned trajectory is obtained by combining the optimization-based polynomial planning method and anti-collision modification.

Comparative results with classical planning and control algorithms show that the proposed planning algorithm can generate a smooth trajectory that satisfies the given constraints, including position, velocity, and acceleration constraints at P_S and P_T. Satisfactory tracking performance even under actuator faults can be achieved by employing the proposed adaptive fault-tolerant controller. In future work, we focus on further improving the proposed method to ensure the safety of autonomous vehicles in dynamic scenarios [18]. Meanwhile, reducing travel delay and energy consumption by using advanced techniques [42], such as reinforcement learning-based multi-objective optimization, to further improve energy efficiency while ensuring safety is an interesting topic.

References

[1] S. Park and S. Hashimoto, "Autonomous mobile robot navigation using passive RFID in indoor environment," *IEEE Trans. Ind. Electron.*, vol. 56, no. 7, pp. 2366–2373, Jul. 2009.

[2] D. Dolgov, S. Thrun, M. Montemerlo, and J. Diebel, "Path planning for autonomous vehicles in unknown semi-structured environments," *Int. J. Robot. Res.*, vol. 29, no. 5, pp. 485–501, 2010.

[3] D. Cao et al., "Future directions of intelligent vehicles: Potentials, possibilities, and perspectives," *IEEE Trans. Intell. Veh.*, vol. 7, no. 1, pp. 7–10, Mar. 2022.

[4] D. K. M. Kufoalor, T. A. Johansen, E. F. Brekke, A. Hepsø, and K. Trnka, "Autonomous maritime collision avoidance: Field verification of autonomous surface vehicle behavior in challenging scenarios," *J. Field Robot.*, vol. 37, no. 3, pp. 387–403, 2020.

[5] F.-Y. Wang et al., "Verification and validation of intelligent vehicles: Objectives and efforts from China," *IEEE Trans. Intell. Veh.*, vol. 7, no. 2, pp. 164–169, Jun. 2022.

[6] F.-Y. Wang, N.-N. Zheng, D. Cao, C. M. Martinez, L. Li, and T. Liu, "Parallel driving in CPSS: A unified approach for transport automation and vehicle intelligence," *IEEE/CAA J. Automatica Sinica*, vol. 4, no. 4, pp. 577–587, Oct. 2017.

[7] E. A. Sisbot, L. F. Marin-Urias, R. Alami, and T. Simeon, "A human aware mobile robot motion planner," *IEEE Trans. Robot.*, vol. 23, no. 5, pp. 874–883, Oct. 2007.

[8] T. Huang, J. Wang, and H. Pan, "Adaptive bioinspired preview suspension control with constrained velocity planning for autonomous vehicles," *IEEE Trans. Intell. Veh.*, vol. 8, no. 7, pp. 3925–3935, Jul. 2023.

[9] H. Pan, Y. Hong, W. Sun, and Y. Jia, "Deep dual-resolution networks for real-time and accurate semantic segmentation of traffic scenes," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 3, pp. 3448–3460, Mar. 2023.

[10] F.-Y. Wang, "Parallel control and management for intelligent transportation systems: Concepts, architectures, and applications," *IEEE Trans. Intell. Transp. Syst.*, vol. 11, no. 3, pp. 630–638, Sep. 2010.

[11] L. Chen et al., "Milestones in autonomous driving and intelligent vehicles: Survey of surveys," *IEEE Trans. Intell. Veh.*, vol. 8, no. 2, pp. 1046–1056, Feb. 2023.

[12] B. Li et al., "From formula one to autonomous one: History, achievements, and future perspectives," *IEEE Trans. Intell. Veh.*, vol. 8, no. 5, pp. 3217–3223, May 2023.

[13] S. Teng, L. Chen, Y. Ai, Y. Zhou, Z. Xuanyuan, and X. Hu, "Hierarchical interpretable imitation learning for end-to-end autonomous driving," *IEEE Trans. Intell. Veh.*, vol. 8, no. 1, pp. 673–683, Jan. 2023.

[14] A. J. Rimmer and D. Cebon, "Planning collision-free trajectories for reversing multiply-articulated vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 7, pp. 1998–2007, Jul. 2016.

[15] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Trans. Syst. Sci. Cybern.*, vol. 4, no. 2, pp. 100–107, Jul. 1968.

[16] H. Pan, X. Chang, and W. Sun, "Multitask knowledge distillation guides end-to-end lane detection," *IEEE Trans. Ind. Inform.*, vol. 19, no. 9, pp. 9703–9712, Sep. 2023.

[17] S. M. Persson and I. Sharf, "Sampling-based A* algorithm for robot path-planning," *Int. J. Robot. Res.*, vol. 33, no. 13, pp. 1683–1708, 2014.

[18] T. Huang, H. Pan, W. Sun, and H. Gao, "Sine resistance network-based motion planning approach for autonomous electric vehicles in dynamic environments," *IEEE Trans. Transport. Electrific.*, vol. 8, no. 2, pp. 2862–2873, Jun. 2022.

[19] J. Faigl and P. Váňa, "Surveillance planning with bézier curves," *IEEE Robot. Automat. Lett.*, vol. 3, no. 2, pp. 750–757, Apr. 2018.

[20] C.-C. Tsai, H.-C. Huang, and C.-K. Chan, "Parallel elite genetic algorithm and its application to global path planning for autonomous robot navigation," *IEEE Trans. Ind. Electron.*, vol. 58, no. 10, pp. 4813–4821, Oct. 2011.

[21] A. M. Lekkas and T. I. Fossen, "Integral LOS path following for curved paths based on a monotone cubic hermite spline parametrization," *IEEE Trans. Control Syst. Technol.*, vol. 22, no. 6, pp. 2287–2301, Nov. 2014.

[22] J. Wittmann and D. J. Rixen, "Time-optimization of trajectories using zero-clamped cubic splines and their analytical gradients," *IEEE Robot. Automat. Lett.*, vol. 7, no. 2, pp. 4528–4534, Apr. 2022.

[23] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2011, pp. 2520–2525.

[24] A. Piazzi and A. Visioli, "Global minimum-jerk trajectory planning of robot manipulators," *IEEE Trans. Ind. Electron.*, vol. 47, no. 1, pp. 140–149, Feb. 2000.

[25] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs for safety critical systems," *IEEE Trans. Autom. Control*, vol. 62, no. 8, pp. 3861–3876, Aug. 2017.

[26] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control barrier functions: Theory and applications," in *Proc. IEEE 18th Eur. Control Conf.*, 2019, pp. 3420–3431.

[27] H. Pan, D. Zhang, W. Sun, and X. Yu, "Event-triggered adaptive asymptotic tracking control of uncertain MIMO nonlinear systems with actuator faults," *IEEE Trans. Cybern.*, vol. 52, no. 9, pp. 8655–8667, Sep. 2022.

[28] S. L. Herbert, M. Chen, S. Han, S. Bansal, J. F. Fisac, and C. J. Tomlin, "FaSTrack: A modular framework for fast and guaranteed safe motion planning," in *Proc. IEEE 56th Annu. Conf. Decis. Control*, 2017, pp. 1517–1522.

[29] M. Chen, S. Herbert, and C. J. Tomlin, "Exact and efficient Hamilton-Jacobi guaranteed safety analysis via system decomposition," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2017, pp. 87–92.

[30] G. E. Fainekos, A. Girard, H. Kress-Gazit, and G. J. Pappas, "Temporal logic motion planning for dynamic robots," *Automatica*, vol. 45, no. 2, pp. 343–352, 2009.

[31] H. Kress-Gazit, G. E. Fainekos, and G. J. Pappas, "Temporal-logic-based reactive mission and motion planning," *IEEE Trans. Robot.*, vol. 25, no. 6, pp. 1370–1381, Dec. 2009.

[32] W. Chung et al., "Safe navigation of a mobile robot considering visibility of environment," *IEEE Trans. Ind. Electron.*, vol. 56, no. 10, pp. 3941–3950, Oct. 2009.

[33] C. J. Ostafew, A. P. Schoellig, and T. D. Barfoot, "Robust constrained learning-based NMPC enabling reliable mobile robot path tracking," *Int. J. Robot. Res.*, vol. 35, no. 13, pp. 1547–1563, 2016.

[34] H. Pan, C. Zhang, and W. Sun, "Fault-tolerant multiplayer tracking control for autonomous vehicle via model-free adaptive dynamic programming," *IEEE Trans. Rel.*, vol. 72, no. 4, pp. 1395–1406, Dec. 2023.

[35] W. Wang, J. Huang, and C. Wen, "Prescribed performance bound-based adaptive path-following control of uncertain nonholonomic mobile robots," *Int. J. Adaptive Control Signal Process.*, vol. 31, no. 5, pp. 805–822, 2017.
[36] S. Teng et al., "Motion planning for autonomous driving: The state of the art and future perspectives," *IEEE Trans. Intell. Veh.*, vol. 8, no. 6, pp. 3692–3711, Jun. 2023.
[37] C. P. Bechlioulis and G. A. Rovithakis, "Robust adaptive control of feedback linearizable MIMO nonlinear systems with prescribed performance," *IEEE Trans. Autom. Control*, vol. 53, no. 9, pp. 2090–2099, Oct. 2008.
[38] X. Tao, N. Lang, H. Li, and D. Xu, "Path planning in uncertain environment with moving obstacles using warm start cross entropy," *IEEE/ASME Trans. Mechatron.*, vol. 27, no. 2, pp. 800–810, Apr. 2022.
[39] N. Wang, S. Lv, M. J. Er, and W.-H. Chen, "Fast and accurate trajectory tracking control of an autonomous surface vehicle with unmodeled dynamics and disturbances," *IEEE Trans. Intell. Veh.*, vol. 1, no. 3, pp. 230–243, Sep. 2016.
[40] P. Shi, X. Yu, X. Yang, J. J. Rodrguez-Andina, W. Sun, and H. Gao, "Composite adaptive synchronous control of dual-drive gantry stage with load movement," *IEEE Open J. Ind. Electron. Soc.*, vol. 4, pp. 63–74, 2023.
[41] J. Hu, Y. Zhang, and S. Rakheja, "Adaptive lane change trajectory planning scheme for autonomous vehicles under various road frictions and vehicle speeds," *IEEE Trans. Intell. Veh.*, vol. 8, no. 2, pp. 1252–1265, Feb. 2023.
[42] S. Lim, H. Balakrishnan, D. Gifford, S. Madden, and D. Rus, "Stochastic motion planning and applications to traffic," *Int. J. Robot. Res.*, vol. 30, no. 6, pp. 699–712, 2011.

Huihui Pan (Senior Member, IEEE) received the Ph.D. degree in control science and engineering from the Harbin Institute of Technology, Harbin, China, in 2017, and the Ph.D. degree in mechanical engineering from the Hong Kong Polytechnic University, Hong Kong, in 2018. He is currently a Professor with the Research Institute of Intelligent Control and Systems, Harbin Institute of Technology. His research interests include nonlinear control, vehicle dynamic control, and intelligent vehicles. He is an Associate Editor for IEEE TRANSACTIONS ON INTELLIGENT VEHICLES, IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS, and *Mechatronics*.

Mao Luo received the B.S. degree in automation and the M.S. degree in control science and engineering from the Harbin Institute of Technology, Harbin, China, in 2021 and 2023, respectively. His research interests include motion planning and intelligent vehicles.

Jue Wang received the B.S. degree in automation and the M.S. degree in pattern recognition and intelligent systems from Huaqiao University, Xiamen, China, in 2016 and 2019, respectively, and the Ph.D. degree from the Harbin Institute of Technology, Harbin, China, in 2023. She is currently a Postdoctoral Fellow with the Ningbo Institute of Intelligent Equipment Technology Company Ltd., Ningbo, China, and the University of Science and Technology of China, Hefei, China. Her research interests include adaptive control, fault-tolerant control, mechatronics, and intelligent vehicles.

Tenglong Huang (Graduate Student Member, IEEE) received the B.E. degree in automation from the Henan University of Technology, Zhengzhou, China, in 2019. He is currently working toward the Ph.D. degree with the Research Institute of Intelligent Control and Systems, Harbin Institute of Technology, Harbin, China. His research interests include motion planning, vehicle dynamics control, adaptive control, fault-tolerant control, and intelligent vehicles.

Weichao Sun (Senior Member, IEEE) received the Ph.D. degree in control science and engineering from the Harbin Institute of Technology, Harbin, China, in 2013. He is currently a Professor with the Research Institute of Intelligent Control Systems, Harbin Institute of Technology. His research interests include adaptive robust control, mechatronics, robotics, and autonomous vehicles. He is an Associate Editor for IEEE/ASME TRANSACTIONS ON MECHATRONICS, IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS: SYSTEMS, and *Journal of Dynamic Systems, Measurement and Control*.