



Online on-Road Motion Planning Based on Hybrid Potential Field Model for Car-Like Robot

Xiaohong Chen^{1,2} · Zhipeng Huang¹ · Yuanxi Sun^{1,2} · Yuanhong Zhong³ · Rui Gu¹ · Long Bai^{1,2} 

Received: 16 April 2021 / Accepted: 21 March 2022 / Published online: 21 April 2022
© The Author(s), under exclusive licence to Springer Nature B.V. 2022

Abstract

The application of Middle-sized Car-like Robots (MCRs) in indoor and outdoor road scenarios is becoming broader and broader. To achieve the goal of stable and efficient movement of the MCRs on the road, a motion planning algorithm based on the Hybrid Potential Field Model (HPFM) is proposed in this paper. Firstly, the artificial potential field model improved with the eye model is used to generate a safe and smooth initial path that meets the road constraints. Then, the path constraints such as curvatures and obstacle avoidance are converted into an unconstrained weighted objective function. The efficient least-squares & quasi-Newton fusion algorithm is used to optimize the initial path to obtain a smooth path curve suitable for the MCR. Finally, the speed constraints are converted into a weighted objective function based on the path curve to get the best speed profile. Numerical simulation and practical prototype experiments are carried out on different road scenes to verify the performance of the proposed algorithm. The results show that re-planned trajectories can satisfy the path constraints and speed constraints. The real-time re-planning period is 184 ms, which demonstrates the proposed approach's effectiveness and feasibility.

Keywords Car-like robot · Motion planning · Online trajectory optimization · Potential field collision avoidance

1 Introduction

Mobile robots can assist manual tasks in areas such as unmanned delivery and road cleaning. When mobile robots perform tasks such as delivery and disinfection (especially in special periods such as the outbreak of the COVID-19), they can significantly reduce the probability of contact between people. The MCRs have advantages in stability, load capacity, etc., which are often used as the primary mobile platform in the scenes described above. In addition, the MCRs run mainly on structured roads, which are characterized by limited lane

width and long lane look-ahead. The MCRs need to realize complex tasks in complex environments, which have extremely high requirements on the performance of navigation systems. The motion planning component plays a critical cardinal role in the navigation system, whose core function is to generate the trajectory based on the road map.

Because of MCRs' poor flexibility, they have high requirements on trajectory quality. Therefore, to improve the practicality of the motion planning model for the MCRs, the following five aspects need to be considered comprehensively:

- 1) **Path Curve Quality.** Under the constraints of road and robot kinematics, the path is required to have excellent smoothness, ample clearance, and long length.
- 2) **Speed Profile Quality.** Under the dynamic constraints, it is required to generate a speed profile with fast response, excellent tracking stability, and fine controllability.
- 3) **Computation Efficiency.** Under the premise of the trajectory quality, the model complexity needs to be reduced to guarantee real-time performance.
- 4) **Coupling Relationship.** The motion planning model is the bridge between the perception and control process in the navigation system, so it is necessary to consider the coupling relationship to enhance navigation performance.

Xiaohong Chen and Zhipeng Huang contributed equally to this work.

✉ Long Bai
bailong@cqu.edu.cn

¹ State Key Laboratory of Mechanical Transmission, College of Mechanical and Vehicle Engineering, Chongqing University, Chongqing 400044, China

² Chongqing Key Laboratory of Metal Additive Manufacturing (3D Printing), Chongqing University, Chongqing 400044, China

³ School of Microelectronics and Communication Engineering, Chongqing University, Chongqing 400044, China

- 5) **Modifiability.** The motion planning model should meet the requirements of easy adjustability, easy controllability, (sub-)optimal solution, good providentness, and re-planning performance.

Some algorithms have been proposed to solve the above problem, which can be divided into the following three categories:

- 1) Action-space sampling-based approaches. Typical algorithms are the dynamic window approach (DWA) [1], curvature-velocity method (CVM) [2], etc., which select the lowest cost trajectory. Although the computation efficiency of these approaches is incredibly high, the simple trajectory type results in divergence of candidate trajectories. For the indoor corridor scene, LCM [3] combines the speed space and the corridor direction space, thereby improving its providentness. However, if these action-space sampling-based approaches are applied to the MCRs, the practical providentness will be reduced due to the steering constraint. To solve this shortcoming, one means to integrate the global path planner, such as the integrated DWA and A* algorithm [4]. However, the global path planner will increase the computation cost significantly. In summary, the above action-space sampling-based motion planning approaches have excellent real-time performance, but the trajectory quality is average and relies heavily on the global planner.
- 2) State-space sampling-based approaches. This category of approaches utilizes the flexibility and diversity of polynomial curves to extend the trajectory length, improving providentness significantly. Werling et al. proposed the Frenet Planner (FP), the polynomial curves are used to connect adjacent states smoothly on the Frenet frame. Finally, the trajectory is selected with the lowest overall cost [5]. Similarly, the anticipatory kinodynamic motion planner (AKMP) proposed by Talamino et al. uses the path-speed decoupling method, and optimization parameters are reduced by the symmetrical characteristic of trajectory [6]. In addition, Xu et al. proposed a path-speed loop iterative optimization method to approximate the optimal trajectory in a limited time [7]. In summary, the above approaches can generate trajectory with good quality, trajectory expression with polynomials and path-speed decoupling is worthy of reference. However, they require complete probability assurance to decrease randomness. Besides, the computation cost increases significantly with the increase in sampling density.
- 3) Optimization-based approaches. Sattel et al. built an artificial potential field model to generate an initial path, and the Elastic Band (EB) model was applied to smooth the path [8]. Dolgov et al. used the conjugate gradient method

to optimize the initial path [9]. In the meantime, Rösman et al. proposed Timed Elastic Band (TEB) algorithm to describe the robot state with constraints in the sparse graph, and then used Levenberg-Marquardt solver to obtain a low-cost trajectory [10, 11]. Gu et al. proposed the decoupled space-time trajectory planning framework to reduce the optimization cost, using the improved EB model optimization to obtain a smooth path based on the initial path [12]. The Convex Elastic Smoothing (CES) algorithm proposed by Zhu et al., decomposing the trajectory optimization into the path and speed optimization. The best path is generated by the EB model. The iterative optimization model is repeated within a limited time to approximate the optimal trajectory [13]. In summary, optimization-based approaches can explicitly deal with various constraints, with the advantages of adjustability and controllability, and can generate good quality trajectories. However, the optimization cost and effect depend highly on the math model, resulting in unstable real-time trajectory planning. And it may fall into the local minimum.

The potential field method (PFM) is widely used for mobile robot navigation because of its simplicity and elegance compared to the high computational cost of high-density sampling of the sampling-based methods described above. However, the PFM has problems such as local minima, and many scholars have made improvements. Ge et al. proposed the new repulsive potential functions by taking the relative distance between the robot and the goal into consideration, which ensures that the goal position is the global minimum of the total potential [14]. Ren et al. adopted Modified Newton's method in continuous navigation functions to reduce the oscillation of the PFM in principle [15]. Ratliff et al. proposed the CHOMP (Covariant Hamilton Optimization Motion Planning) algorithm for the high-dimensional motion planning problem and introduced the use of the Hamilton Monte Carlo algorithm to apply perturbations to restart the optimization process when local minima are encountered [16]. Asadi and Atkins et al. adopted a potential field planning strategy to obtain trajectories from the motion primitives library to rapidly generate a safe landing trajectory for Damaged Airplane [17] and transformed the motion planning multi-objective optimization problem into a single-objective cost function based on the above basis, and proposed a novel approach to translate the subjective information provided by Pareto analysis into a weighted cost function using an entropy-based weight selection method [18].

At present, the part of the above methods that are used to solve the problem of on-road motion planning for MCRs do not balance the relationship well between the trajectory quality and the computation cost, etc., and the information contained in the road is not dug out fully.

To make a balanced trade-off, an online motion planning algorithm based on the hybrid potential field is proposed, which combines the improved artificial potential field model with optimization models to generate a high-quality trajectory in real-time. The main contribution of this paper is as follows:

- 1) The eye model is proposed to improve path smoothness effectively generated by artificial potential field model.
- 2) The path optimization efficiency is improved by the robot's geometric pose and the initial value of the optimization variable generated by the least-squares method.

2 Motion Planning

As shown in Fig. 1, the HPFM includes three parts: firstly, an improved artificial potential field method is designed to generate a safe initial path. Then, the constraints such as kinematics and obstacle avoidance are integrated into the path optimization model to generate the best path. Finally, the dynamic constraints are transformed into an objective function to get the analytical solution of the optimal speed profile. The best trajectory is converted into the motion command sequence.

2.1 Initial Path Generation

To obtain a safe and adjustable initial path, an improved artificial potential field model is proposed by optimizing the obstacles model and combining the environmental constraints.

2.1.1 Analysis and Modeling of the Original Motion Scene

Figure 2a shows the scene of the i -th motion planning. By making a vertical line segment of the road centerline through the geometric center point o_{ct} (yellow dot) of the robot in the current state, the road coordinate system $o_{r_i}x_{r_i}y_{r_i}$ is established with their intersection o_{r_i} as the origin, y_{r_i} points to the front (the longitudinal direction of the road), x_{r_i} points to the right. The starting point of the initial path sequence is the same as the starting point of the state sequence.

In the initial path planning model, the MCR is treated as a point, so the obstacles are viewed as inflated. And the motion scene model is composed of boundary obstacles (**OL**) and road obstacles (**OC**), **OL** range is:

$$OL = \{(x_{ol}, y_{ol}) \mid |x_{ol}| \geq D_2, x_{ol} \in (-D_1, D_1), y_{ol} \in \mathbb{R}\} \quad (1)$$

Here, (x_{ob}, y_{ob}) is the point in the **OL** area in $o_{r_i}x_{r_i}y_{r_i}$; $D_1 = w_{rd}/2$, where w_{rd} is the road width; $D_2 = w_{rd}/2 - r_{ex}$, where r_{ex} is the inflation radius.

The dark gray circles represent original road obstacles, while light gray areas indicate expanded areas in Fig. 2a, so the j -th road obstacle **oc_j** is expressed as:

$$oc_j = \{(x_{oc}^j, y_{oc}^j) \mid |P_{oc}^j - P_c^j| \leq D_3\} \quad (2)$$

Here, $P_{oc}^j = (x_{oc}^j, y_{oc}^j)$ is the point in the **oc_j** area in $o_{r_i}x_{r_i}y_{r_i}$; $P_c^j = (x_c^j, y_c^j)$ is the center of the obstacle; $D_3 = r_c^j + r_{ex}$, where r_c^j is the furthest distance from P_c^j to the edge.

The **OC** can be expressed as:

$$OC = \{oc_j \mid j = 1, 2, \dots, m\} \quad (3)$$

2.1.2 Eye Model of Road Obstacle

Because the artificial potential field is sensitive to the outline shape of the obstacles in the discrete road scene, the eye model is designed instead of the circle outline in the initial path generation model to improve the smoothness of the initial path.

Figure 2b shows the eye model, the local coordinate system $o_{c_j}x_{c_j}y_{c_j}$ of eye model in Fig. 2a is established with P_c^j as the origin, where the x_{c_j} direction is the same direction as y_{r_i} , the y_{c_j} direction is opposite to x_{r_i} . The contour of the eye model is an axisymmetric quartic polynomial curve, and polynomial curve shape can be adjusted adaptively with the change of the radius of the road obstacle, taking the upper boundary of the eye model in $o_{c_j}x_{c_j}y_{c_j}$ as an example, the derivatives of the quartic polynomial curve at the vertices and side endpoints are all 0, so the objective equation with constraints are expressed as

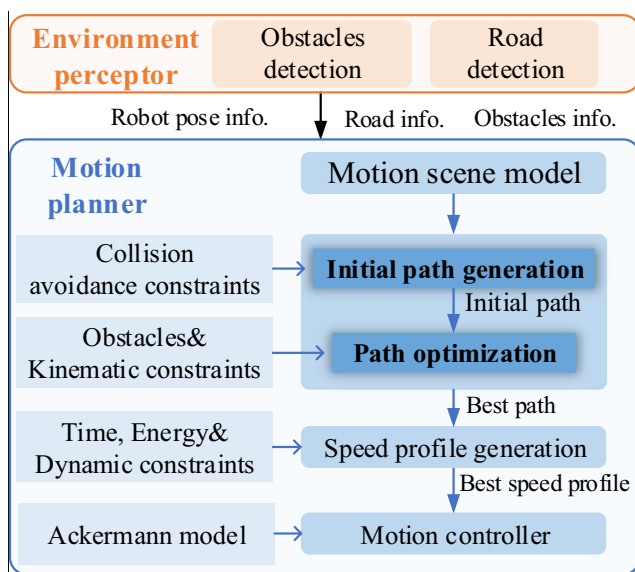
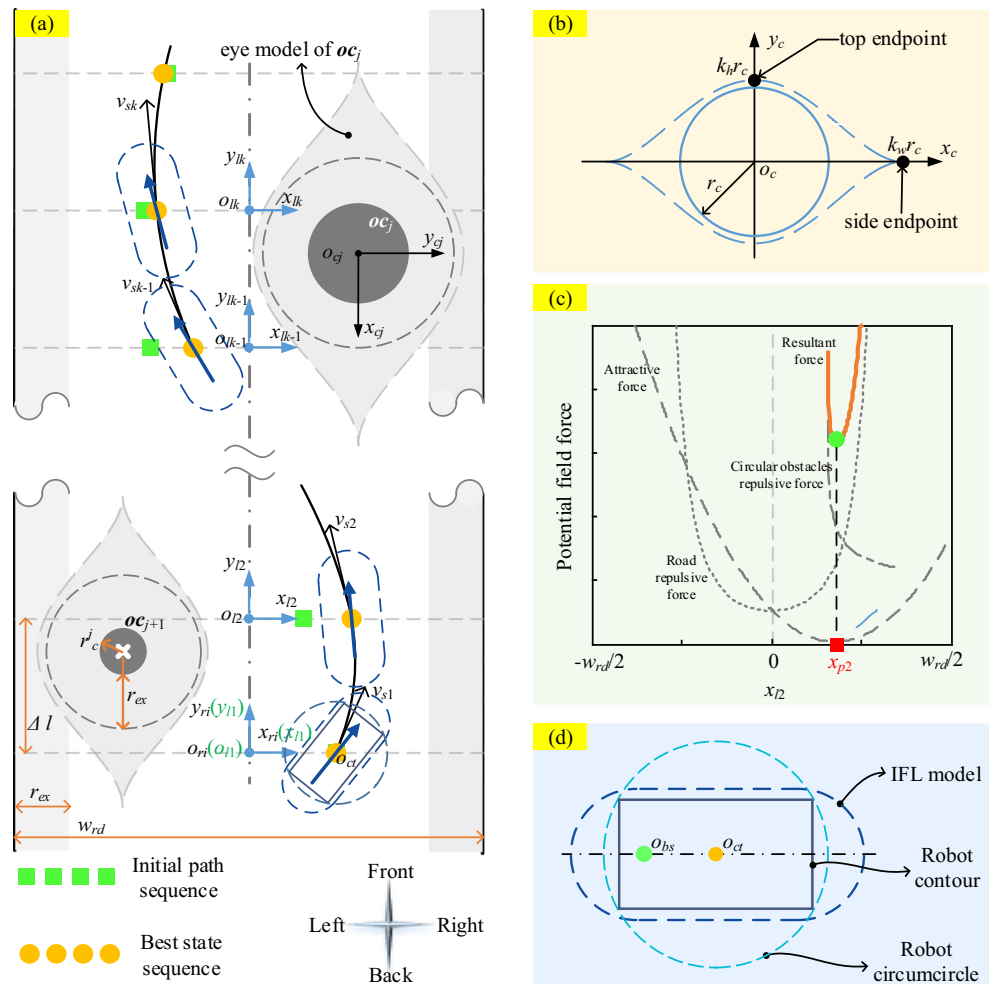


Fig. 1 Detailed operation of the proposed motion planner

Fig. 2 **a** Schematic diagram of motion planning model. **b** Eye model. **c** Potential field force distribution in $o_{l2}-x_{l2}y_{l2}$. **d** Robot model. The point o_{ct} is the robot geometry center, and the point o_{bs} is the center of the axis of the two rear wheels



$$\begin{cases} f_e(x) = Ax^4 + Bx^2 + C \quad (-k_w r_c \leq x \leq k_w r_c) \\ \text{s.t.} \quad f_e(0) = k_h r_c \\ f_e(k_w r_c) = 0 \\ f_e(-k_w r_c) = 0 \end{cases} \quad (4)$$

The polynomial coefficients are expressed as:

$$[A \ B \ C] = \left[\frac{k_h r_c}{(k_w r_c)^4} \quad -\frac{k_h r_c}{(k_w r_c)^2} \quad k_h r_c \right]$$

Here, $[k_h \ k_w]$ are the top and side gain coefficient, respectively, and r_c is the circle's radius (road obstacle).

The contour of the eye model can be changed by adjusting the parameter gain coefficient $[k_h \ k_w]$. To make the eye model boundary fit well to the boundary of road obstacle oc well, $[k_h \ k_w]$ needs to be determined. The optimal gain evaluation function with geometric constraints is designed as:

$$\begin{cases} \text{bestEyeDist}(k_h, k_w) = \min(\mu_h k_h + \mu_w k_w) \\ \text{s.t.} \quad 0 \leq x \leq r_c \\ \left| \sqrt{x^2 + [f_e(x)]^2} - r_c \right| \leq de_{max} \end{cases} \quad (5)$$

Here, $[\mu_h \ \mu_w]$ represents the weight value of $[k_h \ k_w]$. de_{max} represents the maximum allowable fit distance.

As shown in Fig. 3, through numerical traversal, condition judgment methods to select the results that meet the requirements, the colored area indicates all the gain coefficient $[k_h \ k_w]$ that satisfies the constraints of Eq. (5). Because k_h has a more

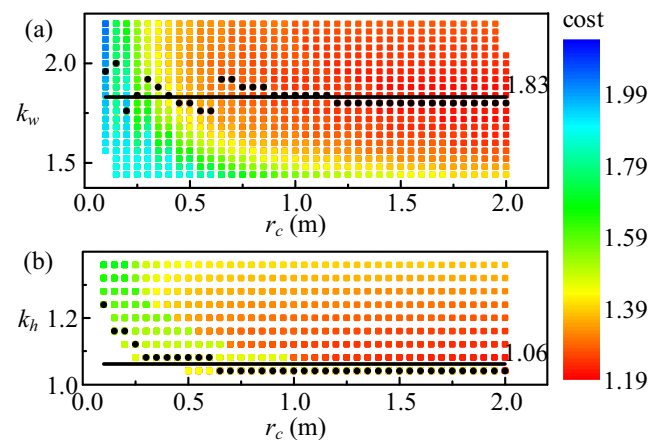


Fig. 3 Eye model gain coefficient optimization results. **a** The relationship between k_h and r_c . **b** The relationship between k_w and r_c

significant influence on the intersection of eye model and circle, $\mu_h = 0.8$ and $\mu_w = 0.2$ are set to Eq. (5), and the optimal gain coefficient value (black dots) tends to be horizontal in Fig. 3. Therefore, the sum of mean and standard deviation of the optimal gain coefficient value set (black dots sequences) are used as the best estimate value (black straight lines): $k_h^* \approx 1.1$, $k_w^* \approx 2.0$. The eye model of oc_j is only a univariate function of oc_j radius, expressed explicitly as:

$$eye(oc_j) = \{(x_e^j, y_e^j) \mid |x_e^j| \leq |k_w D_3|, |y_e^j| \leq |f_e(x_e^j)|\} \quad (6)$$

Here, point (x_e^j, y_e^j) is inside the eye model of oc_j in $o_{cj}x_{cj}y_{cj}$. Thus, the shape of the eye(oc) is automatically adjustable with the change of the radius of oc .

Therefore, the eye model of OC can be expressed as:

$$EOC = \{eoc_j \mid eoc_j = Trd_{oc_j}^r eye(oc_j), j = 1, 2, \dots, m\} \quad (7)$$

Here, eoc_j is eye model area of the j -th road obstacle, Trd_{oc_j} is transformation matrix from $o_{cj}x_{cj}y_{cj}$ to $o_{ri}x_{ri}y_{ri}$.

2.1.3 Initial Path Planning Model

Taking $o_{ri}x_{ri}y_{ri}$ as the starting coordinate system in Fig. 2a, and picking n points in sequence with the step length Δl as the origin along the road centerline (y_{ri} direction) to establish a local coordinate system set $\{o_{lk}x_{lk}y_{lk} \mid k = 1, 2, \dots, n\}$, then the initial path sequence is expressed as:

$$Path = \{(x_{pk}, y_{pk}) \mid k = 1, 2, \dots, n\} \quad (8)$$

Here, (x_{pk}, y_{pk}) is the coordinate of the k -th initial path point P_{rk} in $o_{ri}x_{ri}y_{ri}$, $y_{pk} = \Delta l(k-1) + y_{p1}$, and the resultant force $F_{mix}^{lk}(x_{pk})$ at P_{rk} can be expressed as:

$$F_{mix}^{lk}(x_{pk}) = F_{EOC}^{lk}(x_{pk}) + F_{OL}^{lk}(x_{pk}) + F_{ATT}^{lk}(x_{pk}) \quad (9)$$

Here, $F_{EOC}^{lk}(x_{pk})$ and $F_{OL}^{lk}(x_{pk})$ present the repulsive force along the x_{lk} axis direction generated by EOC and OL at P_{rk} , respectively (calculate the repulsive force only for obstacles that intersect with the x_{lk} axis). $F_{ATT}^{lk}(x_{pk})$ present the attractive force between P_{rk-1} and P_{rk} , and the resultant force $F_{mix}^{lk}(x_{pk})$ is the scalar sum of the absolute values of the above three components.

Specifically, As shown in Fig. 2c, all potential field force curves can be obtained by Eq. (9), the closer the position to the obstacles, the greater the obstacle repulsion force $F_{EOC}^{lk}(\cdot)$ and $F_{OL}^{lk}(\cdot)$, and vice versa, the smaller, where, the $F_{OL}^{lk}(\cdot)$ function curve is symmetric about the road centerline, the repulsion force in the inner region of the obstacle is set to infinity, and the direction of the obstacle repulsion force will only be

parallel to the x_{lk} axis; the smaller the distance between P_{rk-1} and P_{rk} , the smaller the attraction force $F_{ATT}^{lk}(\cdot)$, and vice versa, the larger it is, the $F_{ATT}^{lk}(\cdot)$ function curve is symmetric about P_{rk-1} , and the green dot represents the minimum value of the resultant potential field force, which corresponding to the abscissa position x_{p2} (the red square dot) on the x_{l2} axis. Then the path points in $Path$ can be sequentially calculated based on known P_{r1} measured by sensors.

Different from the PFM, the resultant force $F_{mix}^{lk}(x_{pk})$ in Eq. (9) has no attraction force of goal point, and the resultant force is not the vector sum of each component force. The proposed method only needs to successively calculate the abscissa x_{pk} of the smallest resultant force point in the $o_{lk}x_{lk}y_{lk}$, and the ordinate y_{pk} of P_{rk} is calculated in advance, and the path length is positively correlated with the set path number n , so the improved artificial potential field model proposed does not have the problem of falling into local minima due to the combined force vector being zero as in the PFM.

As shown in Fig. 4, the initial path based on the eye model has no mutation, which verifies that the eye model can improve the smoothness of the path curve generated by the artificial potential field model while ensuring safety.

To sum up, the proposed improved artificial potential field model can adaptively and quickly generate smooth and safe initial path curves of arbitrary length in road scenes (including obstacles) without falling into local minima.

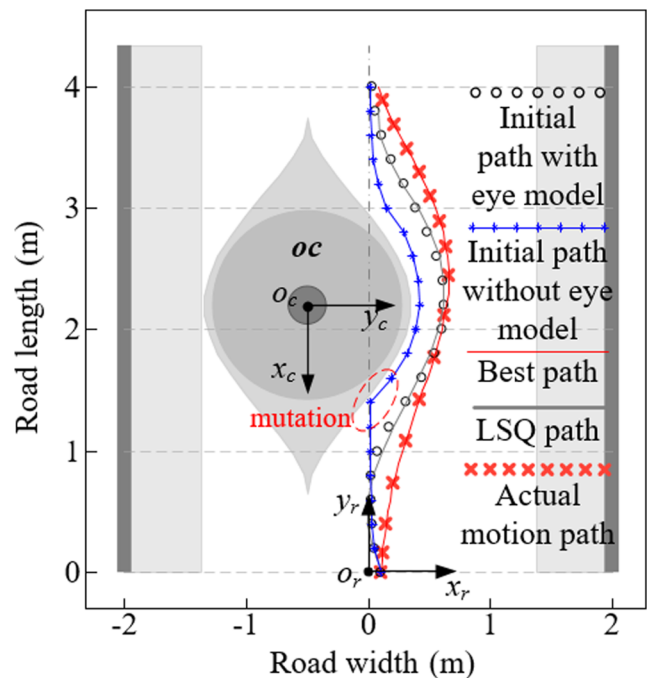


Fig. 4 Different types of trajectory planning result. The initial state: $x_{s1} = 0.1$ m, $y_{s1} = 0$ m, $\theta_{s1} = 0.45\pi$, $\rho_{s1} = 0.1215$ m⁻¹, $v_{s1} = 0.1$ m/s. The best path is generated by LSQ-QN solver, LSQ path is generated by LSQ solver, actual motion path is simulation motion based odometry model

2.2 Path Optimization

Path satisfies the obstacle avoidance constraint but does not fully meet the requirements of curvature constraints. Therefore, a path optimization model considering multiple constraints is designed to generate a new optimal path based on *Path*.

2.2.1 State Sequence with Constraints

The motion state sequence is expressed as:

$$\text{State} = \{S_{rk} | k = 1, 2, \dots, n\} \tag{10}$$

Here, $S_{rk} = [x_{sk}, y_{sk}, \theta_{sk}, \rho_{sk}, v_{sk}]^T$, is the k -th state of o_{ct} in $o_{rr}x_{ri}y_{ri}$. (x_{sk}, y_{sk}) is the position of the o_{ct} ; v_{sk} , θ_{sk} , and ρ_{sk} represent the speed, direction angle, and steering curvature of o_{ct} .

Path optimization constraints at S_{rk} include obstacle avoidance, curvature, and path curve deviation constraint:

$$\begin{cases} o_{sk} \geq o_{min} \\ |\rho_{sk}| \leq \rho_{max} \\ |d\rho_{sk}| \leq d\rho_{max} \\ od_{sk} \leq od_{max} \end{cases} \tag{11}$$

Here, o_{sk} is the nearest distance between the inflation line footprint (ILF) model (the IFL model is composed of a straight-line segment and a circle in Fig. 2d, which is more suitable than the circumscribed circle) and the obstacles, which should be over the set safety distance o_{min} . The curvature ρ_{sk} and its derivative $d\rho_{sk}$ need to be within the maximum curvature ρ_{max} and curvature change rate $d\rho_{max}$, respectively. The distance od_{sk} between S_{rk} and P_{rk} cannot exceed the set value od_{max} .

2.2.2 Optimization Objective Function

To consider the complexity of the path curve, the standard fifth-degree polynomial curve $q(x)$ is applied. The multi-constrained path optimization problem is transformed into an unconstrained optimization model by transforming the inequality constraints Eq. (11) into 4 monotonically increasing sub-cost functions, where, sub-cost function conversion at S_{rk} is expressed as $\chi(e_k, e_m, e_\lambda) = e_\lambda ||\max(0, (|e_k| - e_m))||^2$. The comprehensive cost function $\tilde{V}(\mathbf{a})$ is obtained by summing up the above sub-cost functions of each waypoint (k from 2 to n), and when the comprehensive cost $\tilde{V}(\mathbf{a})$ reaches the lowest value, the optimal path curve polynomial coefficient is obtained, as follows:

$$\mathbf{a}^* = \operatorname{argmin} \tilde{V}(\mathbf{a}) \tag{12}$$

$$\tilde{V}(\mathbf{a}) = \sum_{k=2}^n \left[\begin{matrix} \chi(o_{sk}, o_{min}, \lambda_o) + \chi(\rho_{sk}, \rho_{max}, \lambda_\rho) + \\ \chi(d\rho_{sk}, d\rho_{max}, \lambda_{d\rho}) + \chi(od_{sk}, od_{max}, \lambda_{od}) \end{matrix} \right] \tag{13}$$

Here, $\mathbf{a} = [a_0, a_1, a_2, a_3, a_4, a_5]$, is coefficient of $q(x)$, $[\lambda_o, \lambda_\rho, \lambda_{d\rho}, \lambda_{od}]$ is penalty factor matrix.

2.2.3 Optimization Model Solution

Initial condition: the S_{r1} is calculated by the perception system; so the low-order coefficient \mathbf{a}_l of $q(x)$ is expressed as:

$$\begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} x_{s1} \\ -\cot(\theta_{s1}) \\ \rho_{s1} \left(1 + (\cot\theta_{s1})^2 \right)^{3/2} / 2 \end{bmatrix} \tag{14}$$

So the variables that need to be optimized are reduced to $\mathbf{a}_h = [a_3, a_4, a_5]$, which reduces the optimization computation cost.

Solution method: the LSQ-QN solver combines the least-square method and the quasi-Newton method. First, the least-squares method is used to fit *Path* to obtain the coefficient of $q(x)$ and take the higher-order term coefficient $\mathbf{a}_{h0} = [a_{30}, a_{40}, a_{50}]$. Then, the quasi-Newton method is used to solve Eq. (12) and set \mathbf{a}_{h0} as the initial value to iteratively obtain the local optimal solution \mathbf{a}_h^* . If the path curves corresponding to the optimal solution $\mathbf{a}^* = [\mathbf{a}_l, \mathbf{a}_h^*]$ both meet the constraints in Eq. (11), the optimal solution is retained; otherwise, it is calculated and optimized again from the initial path model.

In summary, it can be seen that the best path is a suboptimal solution near the initial path and is comprehensively influenced by the penalty coefficients of each sub-cost function. As shown in Fig. 4, the best path curve is discretized into the best path sequence according to step length Δl , which is expressed as

$$\{[x_{sk}, y_{sk}, \theta_{sk}, \rho_{sk}]^T | k = 1, 2, \dots, n\} \tag{15}$$

2.3 Speed Profile Generation

The speed profile generation model based on Eq. (15) is designed by considering the motion constraints, etc. Similar to the path optimization model, the optimal speed change $\Delta v_{k,k+1}^*$ is obtained by an unconstrained optimization model, as follows:

$$\Delta v_{k,k+1}^* = \operatorname{argmin} J_k(\Delta v_{k,k+1}) \tag{16}$$

$$J_k = \lambda_w \Delta w_{k,k+1}^2 + \lambda_v \Delta v_{k,k+1}^2 + \lambda_T \Delta T_k^{-1} + \lambda_{ig} \Delta v_{ig}^2 \tag{17}$$

Here, J_k represents the speed cost function from S_{rk} to S_{rk+1} ; $[\lambda_w, \lambda_v, \lambda_T, \lambda_{ig}]$ represents the weight coefficient matrix; $\Delta w_{k,k+1}$, and $\Delta v_{k,k+1}$ represent the velocity change in ΔT_k ;

$\Delta v_{ig} = |v_{ig} - v_{sk+1}|$ represents the target speed following error, where v_{ig} and v_{sk} is target velocity and actual velocity, respectively. The high-level decision planner can adjust the actual speed of the robot by changing the target velocity v_{ig} .

J_k is a quadratic function about $\Delta v_{k,k+1}$, derivate J_k to get the minimum value:

$$\Delta v_{k,k+1}^* = \frac{\left[2\lambda_w \rho_{sk+1} (\rho_{sk+1} - \rho_{sk}) - 2\lambda_{ig} \frac{\lambda_T}{\text{Ed}(\mathbf{S}_{rk}, \mathbf{S}_{rk+1})} \right] v_{sk} + 2\lambda_{ig} v_{ig}}{2\lambda_w \rho_{sk+1}^2 + 2\lambda_v + 2\lambda_{ig} + (\lambda_T / 2 \text{Ed}(\mathbf{S}_{rk}, \mathbf{S}_{rk+1}))} \tag{18}$$

Here, $\text{Ed}(\mathbf{S}_{rk}, \mathbf{S}_{rk+1})$ denotes the Euclidean distance between \mathbf{S}_{rk} and \mathbf{S}_{rk+1} . Combining the Eq. (15) and initial velocity v_{s1} , the best speed profile sequence \mathbf{V} can be calculated, as follows:

$$\mathbf{V} = [V_{s1}, \Delta T_1, V_{s2}, \dots, \Delta T_{n-1}, V_{sn}]^T \tag{19}$$

Here, $V_{sk} = [v_{sk}, w_{sk}]^T$ is the velocity vector at o_{ct} .

Finally, combining Eqs. (15) and (19) to get **State** with time intervals. Based on the Ackermann model, the motion control command sequence is transformed from \mathbf{V} can drive MCR. The actual motion path is shown in Fig. 4, the path length is about 4 m, but the tracking error is within 2 mm at the x_r direction.

3 Experiments and Results

3.1 Numerical Experiments

The robot motion simulation experiments in two typical scenarios of straight and arc are designed based on the analysis in Chapter 2. The parameters of the numerical simulation environment refer to the parameters of the actual scene in Chapter 3-B. The robot geometric size, steering curvature limit, and motion model parameters are the same as those of the actual prototype, road width, obstacle size, and motion speed refer to the actual scene. The performance of the HPFM without global path reference is tested, as shown in Figs. 5 and 6.

As shown in Fig. 5, in the straight and curved roads scenarios, there are two shapes of road obstacles: circle obstacle **oc** and rectangle obstacle **op**, where the rectangle obstacle will be decomposed into multiple circles, and each circle is processed using eye model, as in Fig. 5a where **op₂** is decomposed into two circle obstacles, similarly, obstacles of arbitrary shape can also be decomposed into multiple circles of different sizes, just to ensure that the area occupied by these circles can envelop the corresponding obstacles. Pink snapshots record the relative posture of the robot and the adjacent dynamic road obstacles at the specified time. The dynamic obstacles move at a uniform speed in the direction of the light blue arrow, where the beginning and end of the arrow indicate

the start and end positions of the obstacle; the geometric center point o_{ct} of the robot is used as the starting point of the re-planned trajectory. The robot moves along the path curve of the i -th plan, and the total time required is T_{rti} (start time t_{si} to end time t_{ei}). The actual moving time is T_{rp} , where the actual motion path is represented by the color curve.

According to the simulation results, the HPFM can dynamically adjust the trajectory in real-time so that the robot can avoid all static obstacles and dynamic obstacles in lateral motion, oblique motion, conjugate motion, and opposite motion effectively, and the scene change is small, the adjacent re-planned path curves have good coincidence and consistency in Fig. 5.

The re-planned path length is about 3-4 m, whose curvature is within the range $[-0.59, 0.59]$, and the change is uniform and stable. The difference is that the curvature of the curve road (0.2 m^{-1} , road centerline radius $r_{rd} = 5 \text{ m}$) itself forces the curvature of the robot's movement to be maintained near 0.1 m^{-1} .

The axis of the eye model changes with the trend of the road centerline. The eye model contour can automatically deform to adapt to the road, and the outer contour curve remains smooth, which has a positive effect on the smoothness of the path curve in different road scenes and has good practicability.

Figure 6a and b describe the speed profile of the point o_{ct} in Fig. 5a and b, the re-planned path in Fig. 5 is coupled with the corresponding re-planned speed profiles, orange and green point express the start point of re-planned speed profiles. As shown in Fig. 6, the actual linear speed of the robot is positively correlated with the target speed change, and the actual speed responds quickly. In the steady-state phase, the adjacent re-planned linear velocity profile has a high degree of coincidence, and the steady-state error e_{ss} is due to the limitation of other penalty terms in the speed cost function.

The above simulation experiment uses the Intel(R) Core(TM) i5-3230M 2.60GHz computing platform. After more than 30 re-planned tests, the average re-planned period is about 184 ms, whose three main parts are: initial path planning(92 ms), path optimization(90 ms), speed planning(1.8 ms).

3.2 Practical Robot Results

As shown in Fig. 7, the dimensions of the experimental prototype are $1085 \times 616 \times 925 \text{ (mm}^3\text{)}$, the total mass is about 80 kg, and the total power of the drive motor is about 550 W.

The experimental prototype is equipped with a ZED Stereo camera and RPLidar A2 lidar to collect environmental information, ZED camera mainly collects 3D point cloud data in the front area of the robot, while lidar mainly collects 2D point cloud data in 360 degrees around the robot. According to the geometry, color, and other features of the obstacles, the above point cloud is cut into a collection of point cloud clusters using

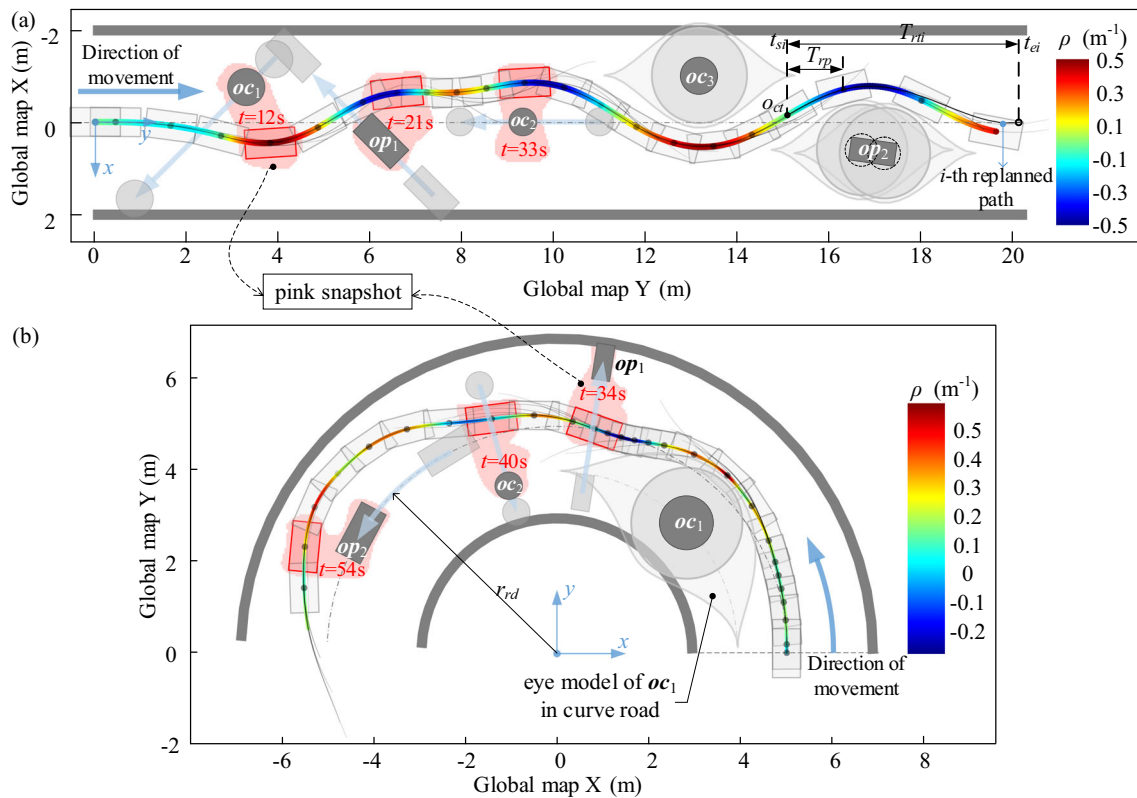


Fig. 5 Numerical simulation experiments of the motion planning model. **a** Straight road scene. **b** Curve road scene. In the initial path generation model, all road obstacles are replaced with corresponding eye models,

however, only the eye model of the static road obstacle (oc_3 and op_2 in subfigure (a), oc_1 in subfigure (b)) in the road scene is drawn for simplified expression

a threshold segmentation method, and then each point cloud cluster is fitted to an approximate geometry (such as circle, straight line segment, rectangle, etc.) according to its features in turn to obtain a collection of road obstacles and boundary obstacles.

The above algorithms for sensor data processing, trajectory planning, autonomous positioning, and motion control are developed based on the Robot Operating System, and the

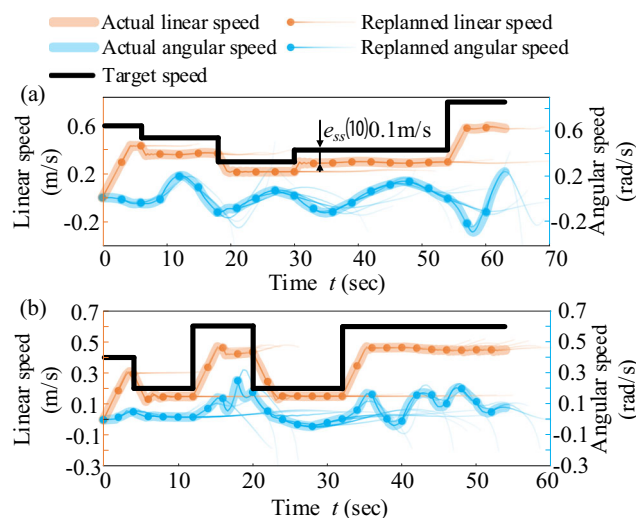


Fig. 6 Simulation results of speed generation. **a** Speed profile in straight road scene, **b** Speed profile in curve road scene

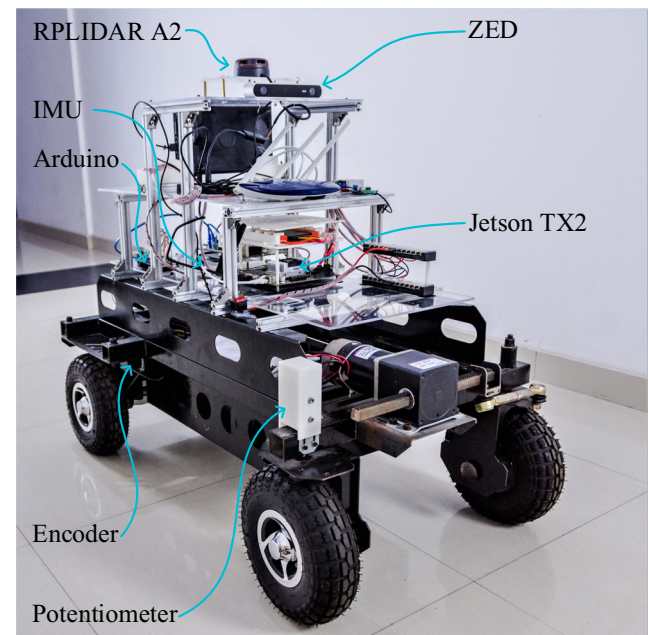


Fig. 7 Experimental prototype

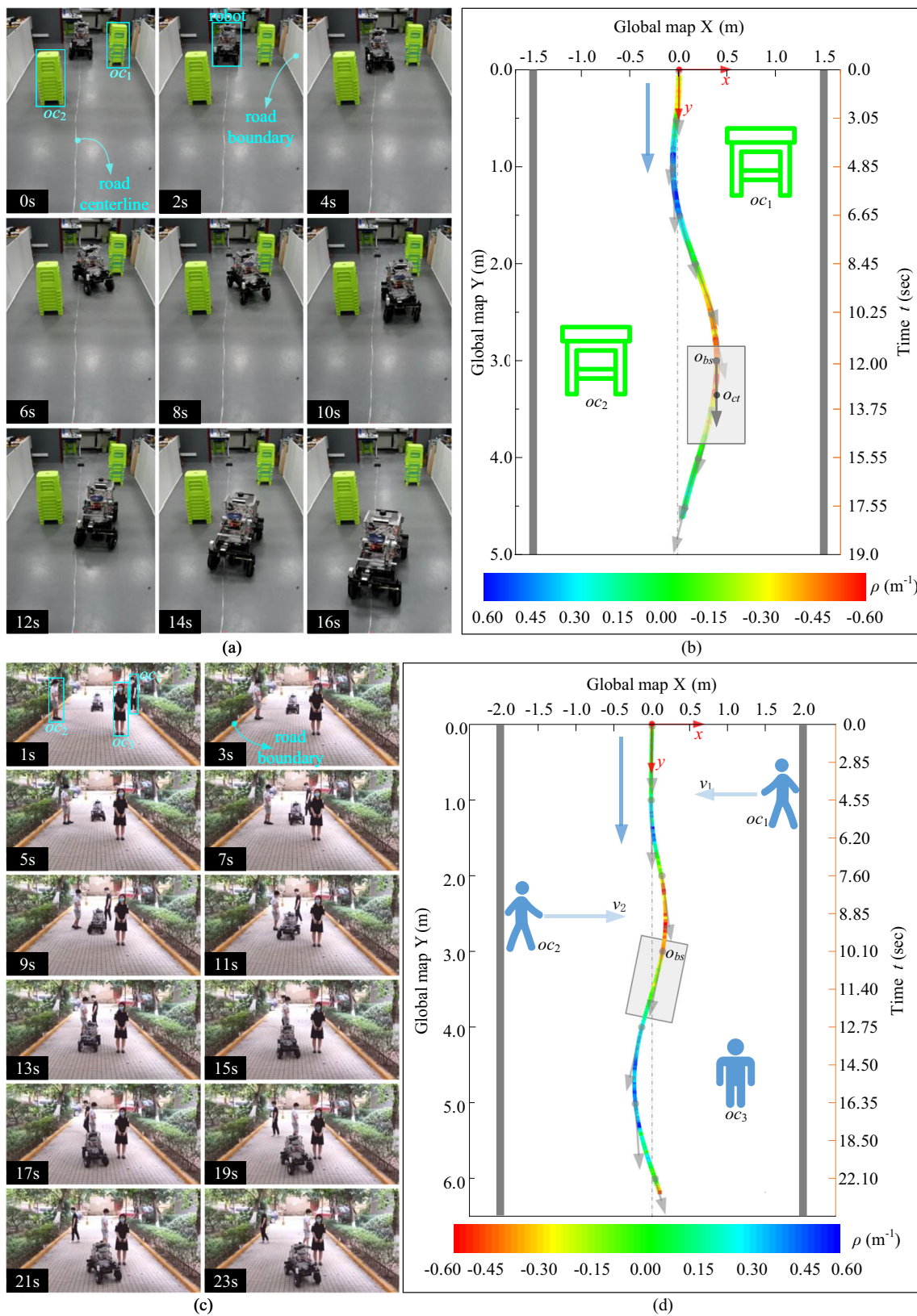


Fig. 8 Prototype experiments. a Indoor scene, b Motion path in the indoor scene, c Outdoor scene, d Motion path in the outdoor scene

autonomous navigation system running on the Jetson TX2 development board sends speed commands to the Arduino microcontroller. The Arduino converts the command into the control signal to drive the motor rotation and collects data from the IMU and encoders mounted on both rear wheels in real time, and feeds it to the autonomous navigation system. The positioning subsystem combines the IMU data, encoders data, and the relative position information between the prototype and the boundary obstacles, and then uses the EKF algorithm to fuse and calculate the relationship diagram of the state (pose and speed) of the prototype (point o_{bs}) and time (See Figs. 8b, d and 9).

As shown in Fig. 8, subfigures (a) and (c) describe the experimental effects of the indoor and outdoor straight road scenes, respectively. Subfigures (b) and (d) respectively record the actual motion path curve of the o_{bs} of the robot and the process of robot posture (light gray arrow) over time.

As can be seen from Fig. 8, the robot moves longitudinally along the road and passes through the obstacles (subfigures (c) and (d) contain two dynamic obstacles oc_1 and oc_2). After that, the robot returns to the road centerline. The curvature of the path curve changes steadily within the limit range $[-0.59, 0.59]m^{-1}$, which can be verified from the changing trend of the front wheel rotation angle and attitude.

As shown in Fig. 9, subfigures (a) and (b) describe the speed profile of the point o_{bs} on the robot in the subfigure (a) and (b) of Fig. 8, respectively. As shown in Fig. 8, the actual linear speed can follow the target speed effectively, and the actual speed profile changes more smoothly than the simulation result curve. During 3–16 s in Fig. 9a, the average linear speed is 0.28 m/s, and the standard deviation is 0.049 m/s; during 2–16 s in Fig. 9b, the larger the target speed, the larger the steady-state error, which is consistent with the above simulation results.

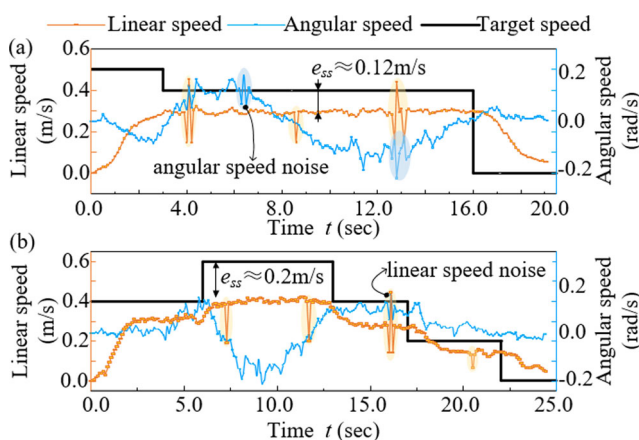


Fig. 9 The speed profile. **a** Speed profile in the indoor scene, **b** Speed profile in the outdoor scene

3.3 Methods Analysis and Summary

Five motion planning algorithms with high similarity in the road scene are selected and compared from five aspects. The evaluation indicators [19] are: ①scene complexity (SC), including the types of road scenes and obstacles; ②path quality (PQ), including the smoothness, clearance, length, and flexibility of the path; ③speed profile quality (SPQ), including response speed, tracking error, stability, and adjustability; ④computation efficiency (CE), including model complexity, real-time performance; ⑤experimental level (EL), including completion about numerical simulation and prototype experiments. According to the data provided in the references, the evaluation results in Fig. 10 are as follows.

1) *SC & EL*: HPFM, AKMP [6], and FP [5] have completed simulation tests in a variety of road scene types that contain a certain number of dynamic and static obstacles, but the AKMP [6] has a relatively low obstacle density in the scene and the remaining methods only complete part of the scene tests.

2) *PQ*: the path length of the LCM [3] is short and improvident, and the smoothness of the motion path is poor. The path quality of the remaining algorithms is all good, and the path length and clearance generated by HPFM are adjustable.

3) *SPQ*: AKMP [6] and FP [5] perform very well in this respect, HC-TEB [10] and CES [13] cannot directly adjust the speed. Although HPFM's speed profile smoothness is slightly weaker, it is better than LCM [3].

4) *CE*: HPFM has good real-time performance without considering hardware performance and the significant differences in the environment map.

4 Conclusion

A new on-road motion planning algorithm for the MCRs in straight/curve road scenes containing dynamic and static obstacles is proposed. Conclusions are as follows:

1) The potential field method integrated with the eye model can improve the initial path more smooth and safer sufficiently, and path optimization model based on the fifth-degree polynomial curve, which can adapt to the dynamic road scenes effectively, and has good real-time performance (182 ms), smoothness, and safety. This method can reduce the complexity of path planning and significantly improve the quality of the path curve. The speed profile generation method provides an analytic solution that can deal with any fifth-degree polynomial curve, which has good real-time performance (1.8 ms) and tracking effect.

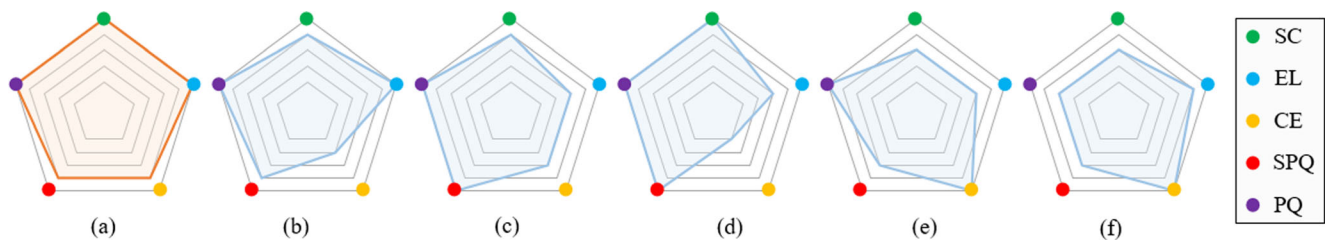


Fig. 10 Motion planning algorithms comparison. The 6 radar charts in the figure correspond to the 6 algorithms, including: (a) HPFM, which is the model proposed in the paper, (b) homology classes Timed-Elastic-

Band(HC-TEB) algorithm [10], (c) AKMP [6], (d) FP [5], (e) CES algorithm [13], (f) LCM [3]

2) The robot can move autonomously and steadily in the experimental scene, which verifies that the HPFM has excellent dynamic adaptability. Besides, the HPFM is also applicable to the motion planning of differential-drive mobile robots and omnidirectional mobile robots in road scenes, where only the curvature constraints need to be modified. HPFM provides a practical and feasible solution for wheeled mobile robots to move on indoor and outdoor roads.

In future works, we will further explore the coupling relationship between the parameters in the motion planning model, and reduce the difficulty of adjusting the model parameters by analyzing the environmental conditions. In addition, follow-up research on the design of the perception system for curve road and other scenes is carried out to broaden the application scope of the robot in the road scenes.

Acknowledgments This work was supported by National Key R & D Program of China (Grant No. 2020YFB1313000), National Natural Science Foundation of China (Grant No. 51975070, 62003060).

Author Contribution Conceptualization, L. Bai; software, Z. Huang; resources, X. Chen, Z. Huang, R. Gu; writing original draft preparation, X. Chen, Z. Huang; writing—review and editing, X. Chen, Y. Sun, Y. Zhong, L. Bai; supervision, L. Bai. All authors have read and agreed to the published version of the manuscript.

Data Availability Not applicable.

Declarations

Ethics Approval Not applicable.

Consent to Participate Not applicable.

Conflict of Interest The authors declare that they have no conflict of interest.

References

1. Fox, D., Burgard, W., Thrun, S.: The dynamic window approach to collision avoidance. *IEEE Robotics & Automation Magazine*. **4**(1), 23–33 (1997)
2. Simmons, R.: The Curvature-Velocity Method for Local Obstacle Avoidance. In: *IEEE international conference on robotics and automation* **4**, 3375–3382 (1996)
3. Ko, N.Y., Simmons, R.G.: The Lane-Curvature Method for Local Obstacle Avoidance. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems. Innovations in Theory, Practice and Applications* **3**, 1615–1621 (1998)
4. Macek, K., Petrović, I., Ivanjko, E.: An Approach to Motion Planning of Indoor Mobile Robots. In: *IEEE International Conference on Industrial Technology* **2**, 969–973 (2003)
5. Werling, M., Kammel, S., Ziegler, J., Gröll, L.: Optimal trajectories for time-critical street scenarios using discretized terminal manifolds. *Int. J. Robot. Res.* **31**(3), 346–359 (2012)
6. Talamino, J.P., Sanfeliu, A.: Anticipatory kinodynamic motion planner for computing the best path and velocity trajectory in autonomous driving. *Robot. Auton. Syst.* **114**, 93–105 (2019)
7. Xu, W., Wei, J., Dolan, J. M., Zhao, H., Zha, H.: A Real-Time Motion Planner with Trajectory Optimization for Autonomous Vehicles. In: *IEEE International Conference on Robotics and Automation* 2061–2067 (2012)
8. Sattel, T., Brandt, T.: From robotics to automotive: lane-keeping and collision avoidance based on elastic bands. *Veh. Syst. Dyn.* **46**(7), 597–619 (2008)
9. Dolgov, D., Thrun, S., Montemerlo, M., Diebel, J.: Path planning for autonomous vehicles in unknown semi-structured environments. *Int. J. Robot. Res.* **29**(5), 485–501 (2010)
10. Rösmann, C., Hoffmann, F., Bertram, T.: Integrated online trajectory planning and optimization in distinctive topologies. *Robot. Auton. Syst.* **88**, 142–153 (2017)
11. Roesmann, C., Feiten, W., Woesch, T., Hoffmann, F., Bertram, T.: Trajectory modification considering dynamic constraints of autonomous robots. In: *ROBOTIK 2012 7th German Conference on VDE* 1–6 (2012)
12. Gu, T., Atwood, J., Dong, C., Dolan, J. M., Lee, J. W.: Tunable and stable real-time trajectory planning for urban autonomous driving. In: *IEEE/RSJ international conference on intelligent robots and systems (IROS)* 250–256 (2015)
13. Zhu, Z., Schmerling, E., Pavone, M.: A convex optimization approach to smooth trajectories for motion planning with car-like robots. In: *54th IEEE conference on decision and control (CDC)* 835–842 (2015)
14. Ge, S.S., Cui, Y.J.: New potential functions for mobile robot path planning. *IEEE Trans. Robot. Autom.* **16**(5), 615–620 (2000)
15. Ren, J., McIsaac, K.A., Patel, R.V.: Modified Newton's method applied to potential field-based navigation for mobile robots. *IEEE Trans. Robot.* **22**(2), 384–391 (2006)
16. Zucker, M., Ratliff, N., Dragan, A.D., Pivtoraiko, M., Klingensmith, M., Dellin, C.M., Bagnell, J.A., Srinivasa, S.S.: CHOMP: covariant Hamiltonian optimization for motion planning. *Int. J. Robot. Res.* **32**(9–10), 1164–1193 (2013)

17. Asadi, D., Sabzehparvar, M., Atkins, E.M., Talebi, H.A.: Damaged airplane trajectory planning based on flight envelope and motion primitives. *J. Aircr.* **51**(6), 1740–1757 (2014)
18. Asadi, D., Atkins, E.M.: Multi-objective weight optimization for trajectory planning of an airplane with structural damage. *Journal of Intelligent & Robotic Systems.* **91**(3), 667–690 (2018)
19. Youakim, D., Ridao, P.: Motion planning survey for autonomous mobile manipulators underwater manipulator case study. *Robot. Auton. Syst.* **107**, 20–44 (2018)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Xiaohong Chen received the B.S. and M.S. degrees in mechanical engineering from North western Polytechnical University, Xi'an, China, in 2007 and 2011, respectively. She is currently faculty with the College of Mechanical and Vehicle Engineering, Chongqing University, Chongqing, China. Her research interests include Robotics and Porous Lattice Structure.

Zhipeng Huang received the M.S. degree in mechanical and vehicle engineering from Chongqing University, Chongqing, China, in 2020. His research interests concern motion planning of mobile robot.

Yuanxi Sun received the B.S. and Ph.D. degrees in mechanical engineering from North western Polytechnical University, Xi'an, China, in 2011 and 2017, respectively. He is currently with the College of Mechanical and vehicle Engineering, Chongqing University, Chongqing, China. His research interests include design, actuation, and control of lower limb prostheses, exoskeleton, and bionic robotics.

Yuanhong Zhong received the Ph.D. degree in communications and information systems from Chongqing University, Chongqing, China, in 2011. He joined the School of Microelectronics and Communication Engineering, Chongqing University, where he is an associate professor. His research interests include the computer vision, machine learning and intelligent unmanned systems.

Rui Gu is a master's student of Chongqing University, Chongqing. His research interests concern motion planning of AUV and ocean current forecasts.

Long Bai received the B.S., M.S. and Ph.D. degrees in mechanical engineering from North western Polytechnical University, Xi'an, China, in 2005, 2008 and 2012, respectively. He is currently a full professor with the College of Mechanical and Vehicle Engineering, Chongqing University, Chongqing, China. His research interests include Robotics and Porous Lattice Structure.