IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 15 October 2023, accepted 30 November 2023, date of publication 5 December 2023,
date of current version 12 December 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3339539

RESEARCH ARTICLE

Motion Planning Method for Car-Like Autonomous Mobile Robots in Dynamic Obstacle Environments

ZHIWEI WANG[1], PEIQING LI[1,2], QIPENG LI[1], ZHONGSHAN WANG[1], AND ZHUORAN LI[3]
[1]School of Mechanical and Energy Engineering, Zhejiang University of Science and Technology, Hangzhou 310023, China
[2]School of Mechanical Engineering, Zhejiang University, Hangzhou 310058, China
[3]Faculty of Information Technology, City University Malaysia, Petaling Jaya 46100, Malaysia

Corresponding author: Peiqing Li (lpqing@hotmail.com)

This work was supported in part by the Zhejiang Lingyan Project 2022C04022, in part by the Natural Science Foundation of Zhejiang Province under Grant LGG20F020008, and in part by the Key (Team) Project of Zhejiang University of Science and Technology under Grant 2021JLZD004.

ABSTRACT Motion planning between dynamic obstacles is an essential capability to achieve real-world navigation. In this study, we investigated the problem of avoiding dynamic obstacles in complex environments for a car-like mobile robot with an incompletely constrained *Ackerman* front wheel steering. To address the problems of weak dynamic obstacle avoidance and poor path smoothing in motion planning with the traditional Timed Elastic Band (TEB) algorithm, We proposed a hybrid motion planning algorithm (TEB-CA,Timed Elastic Band-Collision Avoidance) that combines an improved traditional TEB algorithm and Optimal Reciprocal Collision Avoidance (ORCA) model to improve the ability of the robot to predict dynamic obstacles in advance and avoid collisions safely. Moreover, We also add new constraints to the traditional TEB algorithm, including: jerk constraints, smoothness constraints, and curvature constraints. The algorithm is implemented in $C++$ and evaluated experimentally in the *Gazebo* and *Rviz* simulation environments of the *Robot Operating System* (*ROS*), as well as in actual experimental tests on our car-like autonomous mobile robot "*Little Ant*" which proves the effectiveness of the method, and that the motion planning scheme is more effective in avoiding dynamic obstacles than the traditional TEB and DWA algorithms.

INDEX TERMS Car-like autonomous mobile robot, motion planning, dynamic obstacle avoidance, timed elastic band, optimal reciprocal collision avoidance.

I. INTRODUCTION

with the advancement in robotics, mobile robots are becoming more frequently involved in tasks in complex environments. Examples of such tasks include Automated Guided Vehicles (AGVs) moving materials with other workers [1] or vehicles in factories and warehouses and service robots moving in crowded and dynamic environments [2]. Motion planning techniques ensure that mobile robots can safely navigate from one location to a target location without colliding with other obstacles in the work area. Global path planning methods calculate the best path from the starting point to the endpoint in the map of a known environment, which can avoid static obstacles. A real-time and efficient local path planning method is required for dynamic obstacles, particularly human obstacles.

Typical motion planning methods include the dynamic window approach (DWA) [3], artificial potential field (APF) [4], rapidly exploring random trees (RRT) [5], model predictive control (MPC) [6], and ant colony optimization (ACO) [7]. Real-world environments are partially or entirely unknown, change dynamically, and are prone to the presence of moving objects that block the path of the robot; Autonomous mobile robots should be able to detect, identify, and avoid static and moving obstacles. Rostami et al. [8] proposed an APF approach by adding a conditioning

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

factor to improve the problem of robots caught in local minima; Rebai and Bouchama [9] used the extended Kalman filter (EKF) to track each detected moving obstacle and introduced collision risk information into the DWA objective function to navigate the robot in dynamic environments. Guo et al. [10] used B-spline curves to predict the motion of dangerous dynamic objects combined with a nonlinear model predictive control method to avoid moving obstacles in dynamic environments. When navigating dense and crowded environments, these methods encounter problems, such as poor real-time performance, abrupt trajectory changes, and weak stability. These problems are usually the case with most real robots.

Rösmann et al. [11] initially proposed the Timed Elastic Band(TEB) algorithm to solve the problem of avoiding collisions in real-time during robot motion planning. Compared to other typical motion planning algorithms, the TEB algorithm is more flexible and can adapt to different constraints according to the design requirements and complexity of the scenes. Rösmann et al. [12], [13] proposed extending the TEB technique to use parallel trajectory planning in a spatially unique topology. However, the original TEB algorithm pays more attention to the position of the obstacles and does not prevent or predict the potential collision of dynamic obstacles. Sun et al. [14] introduced a hybrid motion planning method based on the TEB algorithm and artificial potential field (APF) to solve the problem that mobile robots moving in unstructured scenes encounter unstable motion states in obstacle avoidance planning. Additionally, Liu et al. [15] proposed a method called trend-aware motion planning (TAMP), which was combined with the TEB algorithm to model the future position and movement trend of local dynamic obstacles to using them for dynamic obstacle avoidance.

Existing local path planning algorithms cannot avoid dynamic obstacles in complex environments, and robots tend to behave conservatively in highly dynamic environments to minimise risk, manifesting as stopping in place or oscillating between two directions, known as the freezing robot problem (FRP) [16]. The reason for FPR is that the robot cannot predict the trajectory of dynamic obstacles. The velocity obstacle (VO) model [17] is an advantageous method to predict the future trajectory of dynamic obstacles. However, existing research focused on improving and enhancing the velocity obstacle (VO) model and rarely integrates the model with path planning algorithms, which would significantly improve the mobile robot's ability to avoid dynamic obstacles, thus applying mobile robots to more complex real-world scenarios, e.g., hospitals, shopping malls, campuses and other dynamic environments with high human traffic.

The main contributions of this study are as follows.

(1) Aiming at the insufficient constraints existing in the traditional time-elastic band algorithm, we newly add acceleration constraints to reduce the velocity shock, smoothness constraints to make the path smooth, and curvature constraints to reduce the path curvature mutation problem on top of the original constraints;

(2) Combining the local path planning algorithm TEB with the Optimal Reciprocal Collision Avoidance (ORCA), the proposed new method, TEB-CA, can analyze and react to the speed of dynamic obstacles in a more timely manner in complex environments, effectively avoiding collisions with dynamic obstacles;

(3) We apply the new approach to a car-like autonomous mobile robot with an incompletely constrained Ackerman front-wheel steering model and its kinematics and analyze it in comparison with traditional TEB and DWA algorithms by testing the interaction and obstacle avoidance capabilities of the autonomous mobile robot with multiple pedestrians in real application scenarios in both open fields and narrow corridors.

- In Section II describes the traditional TEB algorithm in detail, and new constraints are added.
- Section III briefly describes the fundamentals of the ORCA.
- Section IV presents the overall framework of our improved collision avoidance time-elastic band algorithm (TEB-CA) for robot navigation.
- Section V provides some simulations and Real experiments and discusses them.
- Finally, Section VI summarizes our work and future works.

II. TIMED ELASTIC BAND

A. ACKERMANN STEERING KINEMATIC MODEL

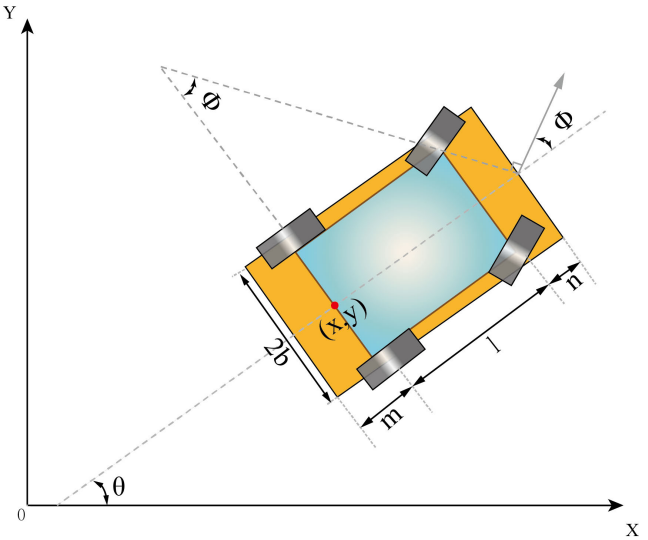


FIGURE 1. Kinematic model of ackerman steering vehicle.

The car-like mobile robot (excluding the four wheels) is assumed to be a rigid body, i.e. the coupled dynamics of the vehicle suspension is not considered.Additionally, the tire sideslip problem was neglected. The kinematic model of the

Ackermann front-wheel steering vehicle is expressed as:

$$\begin{cases} \omega = \dfrac{v}{R} \\ \varphi = \arctan\left(\dfrac{\omega L}{v}\right) \\ \dfrac{dx(t)}{dt} = v(t) \cdot \cos\theta(t) \\ \dfrac{dy(t)}{dt} = v(t) \cdot \sin\theta(t) \\ \dfrac{d\theta(t)}{dt} = \dfrac{v(t) \cdot \tan\phi(t)}{L} \end{cases} . \tag{1}$$

where (x, y) is the midpoint of the rear axle, θ is the directional angle, φ is the virtual steering angle, v is the linear velocity, ϕ is the steering angle of the front wheels, ω is the corresponding angular velocity, L is the wheelbase (i.e., the distance between the front and rear axles). Other parameters set concerning the vehicle geometry include the front overhang length: a,rear overhang length: b, and vehicle width: $2d$.

B. BASIC PRINCIPLE OF TEB ALGORITHM

The motion planning method with dynamic obstacle prediction and avoidance proposed in this paper is mainly based on the TEB framework proposed by Rosmann et al. In the study, Rosmann implemented and published the TEB method as a core component of the robot operating system(ROS) local path planning function package. This section briefly describes the basic principles of the TEB algorithm.

The "*elastic band*" approach optimizes the robot trajectory by modifying the initial trajectory generated by the global path planner in real time to obtain a collision-free path while maintaining the closest global path point and continuously deforming it to maintain the distance from the obstacle. The "time-elastic band" approach builds on this by converting the initial path, consisting of a series of path points, into a time-dependent trajectory, as shown in equation (2), explicitly taking into account temporal information about the motion, such as dynamic constraints on robot speed and acceleration.

$$\begin{cases} \mathrm{Q} = \{\mathrm{S_i}\}_{\mathrm{i=0...n}}, \mathrm{n} \in \mathrm{N} \\ \tau = \{\Delta \mathrm{T_i}\}_{\mathrm{i=0...n-1}} \end{cases} , \tag{2}$$

where S_i is the pose of the vehicle at the time i, Q indicates the sequence of poses in the world coordinate system XOY, τ is the time interval between two consecutive poses S_i and S_{i+1}. Therefore, the TEB trajectory is composed of two sets of pose sequence and time interval sequence, which can be expressed as:

$$\begin{aligned} \mathrm{B} &= (\mathrm{Q}, \tau) \\ &= \{\mathrm{s_1}, \Delta \mathrm{T_1}, \mathrm{s_2}, \Delta \mathrm{T_2}, \ldots, \mathrm{s_{n-1}}, \Delta \mathrm{T_{n-1}}, \mathrm{s_n}\}, \end{aligned} \tag{3}$$

The core idea of the TEB algorithm is to solve the optimal pose sequence of vehicles with time intervals. The optimization problem is defined as a nonlinear least-squares

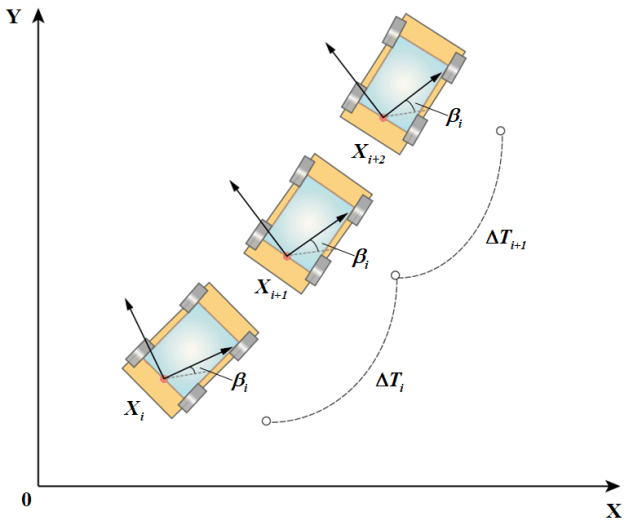


FIGURE 2. Car-like mobile robot trajectory sequence diagram.

cost function that makes the weighted sum of B∗ aggregate:

$$f(B) = \sum_{\mathrm{k}} \gamma_{\mathrm{k}} f_{\mathrm{k}}(B), \tag{4}$$

$$\mathrm{B}* = \underset{\mathrm{B}\setminus\{S_1, S_n\}}{\arg\min} f(B). \tag{5}$$

where,$f_{\mathrm{k}}(B)$ is a constraint function of many different constraints, γ_{k} is the weight coefficient corresponding to the different constraint functions, and $\mathrm{B}\setminus\{S_1, S_n\}$ includes the starting pose S_1 and the target pose S_n are fixed and are not affected by optimization.

The objective function depends on a few successive local poses. This locality of TEB leads to the sparse matrix of the system that can be solved using the method of Levenberg-Marquardt (LM). The G2O framework [18] can effectively solve the LM problem.TEB was executed repeatedly with a response frequency higher than the vehicle control cycle. During trajectory correction, the TEB algorithm continually adds new exposures to the vehicle and removes previous ones. The G2O framework optimizes the TEB trajectory in batches. It iteratively solves it several times in a vehicle control cycle to solve the optimal trajectory, and each iteration generates a new graph. The control input speed *u(t)* in each sampling period is derived from the trajectory sequence.

C. CONSTRAINTS OF THE TEB ALGORITHM

1) NONHOLONOMIC KINEMATIC CONSTRAINTS

As shown in Fig. 4,when the car-like robot travels along the curve on the planned path. Two consecutive motion postures S_i and S_{i+1} are located in a typical arc with the constant curvature. v_i is the angle between posture S_i and direction $d_i = \left[\mathrm{x}_{i+1} - \mathrm{x}_i, \mathrm{y}_{i+1} - \mathrm{y}_i, 0\right]$ at the time i, which must be equal to the corresponding angle v_{i+1} in continuous posture S_{i+1}.

$$v_i = v_{i+1}, \tag{6}$$

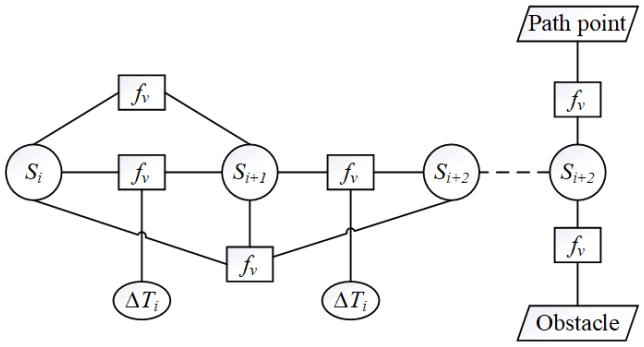


FIGURE 3. Simplified schematic diagram of G2O.

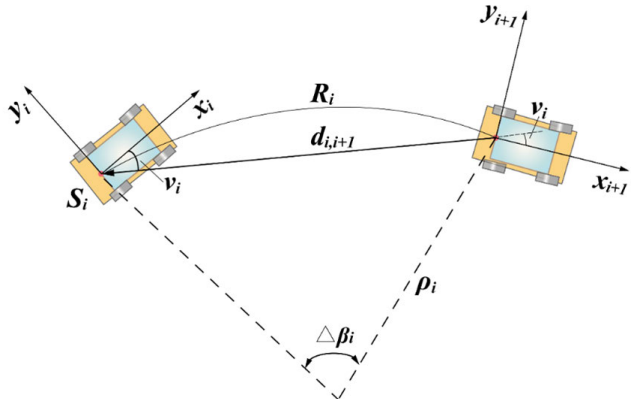


FIGURE 4. Schematic diagram of nonholonomic kinematic constraints between adjacent poses.

Nonholonomic kinematic constraints are expressed as:

$$f_{nh}(x_i, x_{i+1}) = \left\| \left[\begin{pmatrix} \cos\beta_i \\ \sin\beta_i \\ 0 \end{pmatrix} + \begin{pmatrix} \cos\beta_{i+1} \\ \sin\beta_{i+1} \\ 0 \end{pmatrix} \right] \times d_{i,i+1} \right\|^2, \tag{7}$$

2) TIME OPTIMAL CONSTRAINT

TEB considers time information. Minimizing the time interval between all the poses is necessary to ensure that the robot obtains the fastest path. The time-optimal constraint function is given by

$$f_{time} = \left(\sum_{i=0}^{n} \Delta T_i \right)^2, i \in N, \tag{8}$$

3) VELOCITY AND ACCELERATION CONSTRAINTS

The robot speed planned by the TEB algorithm is limited by the maximum speed and acceleration of the robot itself. The constraints of the robot's speed and acceleration are realized by a geometric constraint penalty function. The moving linear velocity and angular velocity can be calculated according to the Euclidean distance or angular distance between two consecutive poses S_i and S_{i+1}, and the time interval ΔT_i between two poses:

$$v_i = \Delta \mathrm{T}_i^{-1} \left\| [x_{i+1} - x_i, y_{i+1} - y_i]^{\mathrm{T}} \right\| \gamma(\mathrm{s}_i, \mathrm{s}_{i+1}), \tag{9}$$

$$\omega_i = \Delta T_i^{-1} (\beta_{i+1}, \beta_i), \tag{10}$$

where $\gamma(\mathrm{s}_i, \mathrm{s}_{i+1})$ is the function for extracting the translation velocity sign. For the nonholonomic robot $\gamma(\mathrm{s}_i, \mathrm{s}_{i+1}) = sign(< d_i, d_{i,i+1} >)$, the acceleration constraint is calculated based on two consecutive average velocities, and three consecutive postures are considered simultaneously with their corresponding time intervals.

$$a_i = \frac{2(v_{i+1} - v_i)}{\Delta T_i + \Delta T_{i+1}}, \tag{11}$$

4) PATH AND OBSTACLE CONSTRAINTS

The objective function of the TEB algorithm depends only on adjacent pose information. This locality results in the generation of a sparse system matrix. The sparse-constrained optimization algorithm is not available in robot frameworks (ROS). These constraints are formulated as piecewise continuous and differentiable cost functions, which penalize the violation of a constraint.

$$e_\tau(x, x_r, \varepsilon, S, n) \approx \begin{cases} \left(\frac{x - (x_r - \varepsilon)}{S} \right)^n, x > x_r - \varepsilon \\ 0, x \leqslant x_r - \varepsilon \end{cases}, \tag{12}$$

x_r is the bound, S is the deformation factor, n is the polynomial coefficient,and ε is a small translation of the approximation.

A path point is a trajectory constraint generated by several waypoints nearby when the vehicle is driving. Which can be expressed as:

$$f_{path} = e_\tau(d_{\min,j}, r_{P\max}, \varepsilon, S, n), \tag{13}$$

The distance between the robot and the obstacle is $d_{\min}$, and the safe distance between the robot and the obstacle is d. The obstacle constraint can be expressed as:

$$f_{ob} = e_\tau(-d_{\min,j}, -r_{O\min}, \varepsilon, S, n). \tag{14}$$

D. ADDITION OF NEW CONSTRAINTS

1) JERK CONSTRAINT

The jerk indicates how fast the acceleration changes. If the acceleration change rate is too high, the torque output of the robot driving motor will suddenly change, which will shock the robot. The jerk constraint can limit the acceleration change rate within a reasonable range to obtain a more accurate and stable speed control.

$$f_{jerk} = \frac{\ddot{s}_{i+1} - \ddot{s}_i}{\Delta T_i}, \tag{15}$$

2) SMOOTHNESS CONSTRAINT

This constraint improves the smoothness of the planned path and moves the robot more in line with the actual situation.

It also reduces the energy loss caused by sudden stops and sharp turns when the robot walks along a planned path.

$$f_{sm} = w_{sm} \sum_{i=1}^{N-1} (\Delta\theta_{i+1} - \Delta\theta_i)^2, \tag{16}$$

w_{sm} is the smoothness weight, which indicates the degree of influence of the change in the path points.

3) CURVATURE CONSTRAINT

This constraint considers the curvature continuity of the entire path, reduces the path curvature mutation, and ensures that the robot smoothly tracks the planned path.

$$f_{curv} = \lim_{\Delta l} \left\| \frac{\Delta\theta}{\Delta l} \right\| = \left| \frac{d\theta}{dl} \right| = \frac{|\ddot{y}|}{\sqrt{(1+\dot{y}^2)^3}}, \tag{17}$$

The curvature is the limit of the angle differentiation and arc length differentiation, where l is the arc length of the path.

In summary, the total cost function of the improved TEB algorithm is

$$\begin{aligned} & f(B) \\ &= \sum_k \gamma_k f_k(B) = \gamma_{nh} \cdot f_{nh}(B) + \gamma_{time} \cdot f_{time}(B) \\ &\quad + \gamma_v \cdot f_v(B) + \gamma_a \cdot f_a(B) + \gamma_{path} \cdot f_{path}(B) + \gamma_{ob} \cdot f_{ob}(B) \\ &\quad + \gamma_{jerk} \cdot f_{jerk}(B) + \gamma_{sm} \cdot f_{sm}(B) + \gamma_{curv} \cdot f_{curv}(B). \end{aligned} \tag{18}$$

III. OPTIMAL RECIPROCAL COLLISION AVOIDANCE

Trajectory planning tasks in dynamic obstacle environments require the robot to predict obstacle avoidance in advance. To meet this requirement, we introduce ORCA model based on the TEB.

The VO (Velocity Obstacle) model has been widely developed and innovated. In the early years, research focused on improving the shortcomings of the traditional VO model, such as the RVO (Reciprocal Velocity Obstacle) [19], the HRVO (Hybrid Reciprocal Velocity Obstacle) [20], and the ORCA (Optimal Reciprocal Collision Avoidance) [21]. Wilkie et al. proposed the GVO (Generalized Velocity Obstacle) [22] method to solve the navigation problem of car-like mobile robots in dynamic environments.

In this section, we briefly review the original concept of the VO model and then focus on the improvement method of the model, which is the basic principle of the ORCA.

A. CONVENTIONAL VELOCITY OBSTACLE MODEL

Assume that A is a moving robot(agent), and its position on the plane is recorded as P_A, B is a moving obstacle and its position is recorded as P_B. The Velocity Obstacle model is represented as $VO_B^A(v_B)$, which represents the speed setting of robot A that will cause A and B to collide in the future when dynamic obstacle B moves at speed V_B.

Fig. 5 demonstrates a geometric representation of the velocity obstacle model, where $A \oplus B$ is the *Minkowski* sum

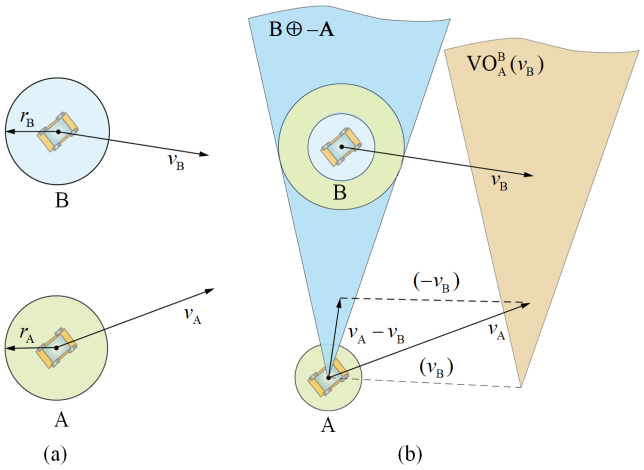


FIGURE 5. Conventional velocity obstacle model.

of A and B. The geometric operation increases the shape of obstacle B so that robot A can be considered a prime point. -A denotes the robot A point set taken inversely and defined as:

$$A \oplus B = \{a + b | a \in A, b \in B\}, -A = \{-a | a \in A\}, \tag{19}$$

$\lambda(p, v)$ is assumed to be a ray starting at position p and pointing in the direction of velocity v.

$$\lambda(p, v) = \{p + tv | t \geqslant 0\}, \tag{20}$$

velocity obstacle(VO)can be defined as:

$$VO_B^A(v_B) = \{v_A | \lambda(p_A, v_A - v_B) \cap B \oplus -A \neq \emptyset\}. \tag{21}$$

Eq. (21) indicates that if $v_A \in VO_B^A(v_B)$ occurs, then the robot A with speed v_A and moving obstacle B with speed v_B will collide in the future time. If $v_A \notin VO_B^A(v_B)$ occurs, then robots A and B will not collide.

B. BASIC PRINCIPLES OF THE ORCA MODEL

The ORCA model proposed by van den Berg et al. [21]was developed based on the VO and the RVO models, which can ensure that robots can avoid collisions with other moving objects in a cluttered workspace. The ORCA model was successfully applied to multi-robot collision avoidance and crowd evacuation simulations. The basic idea is calculating an optional collision-free velocity plane (in the velocity space) for each moving obstacle.Then, the robot will select its optimal velocity from the intersection of all collision-free velocity planes.

The "optimality" of Optimal Reciprocal Collision Avoidance (ORCA) is relative to the collision avoidance problem. It refers to the fact that under a collision avoidance strategy, the mobile robot seeks to minimise the risk of collision, emphasising that the system maintains the superiority of the mission objectives while avoiding collisions. ORCA ensures that the mobile robot maintains its original goals as much as possible while avoiding collisions through mutual

respect and calculating the set of collision-free velocities of multiple mobile obstacles to achieve the overall system optimality under the given constraints. Collision-free velocity sets for multiple moving obstacles to achieve overall system optimality under given constraints. The overall framework of the ORCA collision prediction model is shown in Fig. 6, where the shape of the moving obstacle is represented as an expanding circle in order to be more conservative in avoiding collisions, with an expanding radius that needs to ensure that it encircles the entire obstacle and the use of multiple expanding circles that can more closely approximate the original shape of the obstacle.

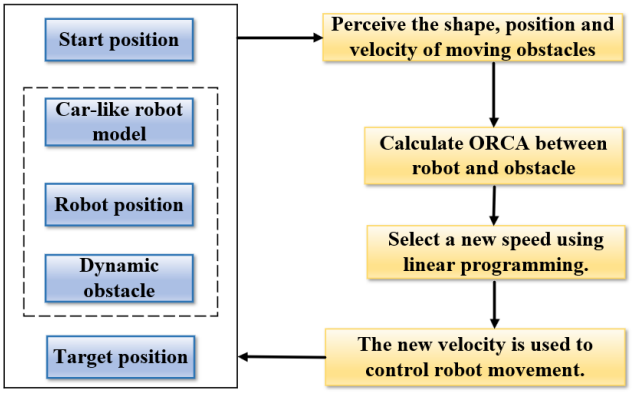


FIGURE 6. General framework diagram of the collision prediction model.

Assuming that V_A and V_B are the current moving velocities of robots A and B, respectively, the velocity obstacle set $VO^t_{A|B}$ indicates that A collides with B at time t when it chooses this velocity in this set. According to the theory of the VO model, if $v_B \in V_B$ and $v_A \notin VO^t_{A|B} \oplus V_B$, then the robots A and B will not collide at t time at the current velocity. The collision avoidance interval of A is defined when B moves at a velocity of V_B, which can be expressed as

$$CA^t_{A|B}(V_B) = \left\{ v | v \notin VO^t_{A|B} \oplus V_B \right\}, \tag{22}$$

For velocity sets V_A and V_B of the robots A and B, if Eq. (22) is satisfied, A and B are said to be reciprocal collision avoidance:

$$V_A \subseteq CA^t_{A|B}(V_B) \ and \ V_B \subseteq CA^t_{B|A}(V_A), \tag{23}$$

Furthermore, if Eq. (24) is satisfied, the velocity sets V_A and V_B are reciprocally maximal, which is given by

$$V_A = CA^t_{A|B}(V_B) \ and \ V_B = CA^t_{B|A}(V_A), \tag{24}$$

If the requirement is added to choose a pair of near-optimal velocities among a reciprocally maximal set of optional collision avoidance velocities (e.g., the velocity of A is v_A^{opt} and the velocity of B is v_B^{opt}), then define $ORCA^t_{A|B}$ for robot A and $ORCA^t_{B|A}$ for robot B. As shown in Fig. 7, One of the yellow areas indicates the permissible velocity set of the optimal interaction collision avoidance model of A with respect to B is $ORCA^t_{A|B}$, where u is the vector starting from $v_A^{opt} - v_B^{opt}$ to the closest $VO^t_{A|B}$ boundary point; n is the normal vector extending outward from $VO^t_{A|B}$, the $VO^t_{A|B}$ boundary point; The second yellow area indicates the permissible velocity set of the ORCA model of autonomous mobile robot B with respect to A is defined as $ORCA^t_{B|A}$, and it is required that the optimal set of optional velocities maximizes each other, as shown in Eq. (25).

$$\begin{aligned} CA^t_{A|B}(ORCA^t_{B|A}) &= ORCA^t_{A|B} and\, CA^t_{B|A}(ORCA^t_{A|B}) \\ &= ORCA^t_{B|A}, \end{aligned} \tag{25}$$

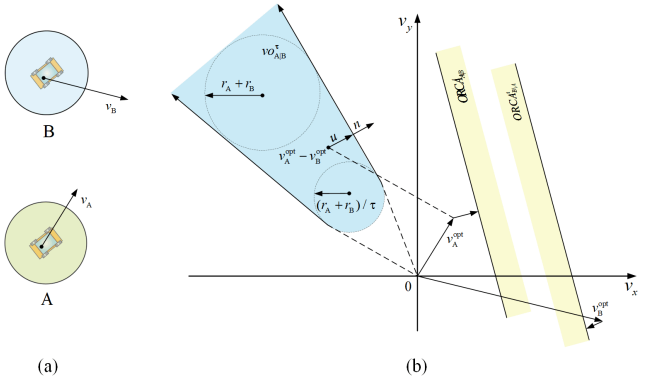


FIGURE 7. Schematic of the optimal reciprocal collision avoidance model.

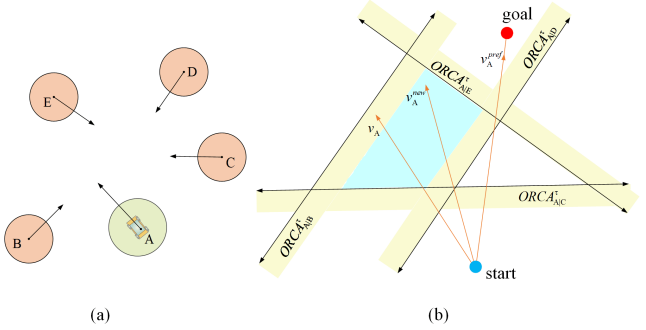


FIGURE 8. Schematic diagram of the optimal reciprocal collision avoidance model for multiple obstacles.

When the number of moving obstacles is more than one, as shown in Fig. 8(a), a schematic diagram of the moving obstacles around autonomous mobile robot A is shown. Autonomous mobile robot A performs continuous perception detection according to the time step Δt and acquires the shape and size, current position, and velocity information of other mobile objects and itself, based on which robot A deduces its collision-free velocity set $ORCA^t_{A|B}$,$ORCA^t_{A|C}$,$ORCA^t_{A|D}$ and $ORCA^t_{A|E}$ within the time τ concerning mobile obstacle A. At the same time, robot A generates the velocity set with the other mobile obstacles, which are constantly influencing each other. The allowable speed set of robot A is the intersection of collision-free speed sets with respect to all moving obstacles.

Define the set of collision-free velocities in time τ of A with respect to all other moving obstacles as $ORCA_A^\tau$:

$$ORCA_A^\tau = D(0, v_A^{\max}) \underset{B\neq A}{\cap} ORCA_{A|B}^\tau \underset{C\neq A}{\cap} ORCA_{A|C}^\tau \underset{D\neq A}{\cap} ORCA_{A|D}^\tau \underset{E\neq A}{\cap} ORCA_{A|E}^\tau, \tag{26}$$

The blue region in Fig. 8(b) is $ORCA_A^\tau$, which contains the set of permissible velocities of the autonomous mobile robot A with respect to all moving obstacles, where $D(0, v_A^{\max})$ is a circle with the radius of A's maximum velocity, which represents the maximum velocity constraint of A. v_A denotes the current velocity of A, and v_A^{new} denotes a new velocity chosen by A after the collision prediction model calculations, which is closest to the direct target (without moving obstacles) within the set of permissible velocity regions with the preferred velocity v_A^{pref}.

$$v_A^{new} = \underset{v\in ORCA_A^\tau}{arg\min} \left\| v - v_A^{pref} \right\|, \tag{27}$$

Autonomous mobile robot A moves to a new position with speed v_A^{new}:

$$p_A^{new} = p_A + v_A^{new} \Delta t. \tag{28}$$

IV. PROPOSED COLLISION AVOIDANCE TIME ELASTIC BAND ALGORITHM

In order to make full use of the advantages of TEB trajectory planning and ORCA collision-speed-prediction model. We proposed a collision avoidance–time elastic band algorithm(TEB-CA)in dynamic social environments. Our proposed algorithm considers the dynamic constraints of mobile robots and incorporates ORCA to predict non-collision velocity regions and generates safe and smooth motion trajectories.

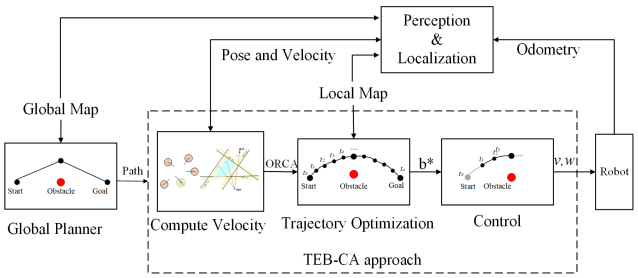


FIGURE 9. Overall framework of the proposed TEB-CA motion planning method.

The core idea of a conventional TEB algorithm is to solve the optimal sequence of robot poses with time intervals. We incorporated the obstacle avoidance velocity v_A^{ORCA} predicted by the ORCA model into the total cost function of the TEB algorithm to obtain the optimal collision avoidance trajectory. The nonlinear least-squares optimization problem can be re-expressed as:

$$\begin{aligned} \mathrm{B}* = \underset{\mathrm{B}\setminus\{S_1,S_\mathrm{N}\}}{arg\min} \sum_k \gamma_k f_k(B) = \underset{\mathrm{B}\setminus\{S_1,S_\mathrm{N}\}}{arg\min} \{ & \gamma_{nh}\cdot f_{nh}(B) \\ & + \gamma_{time}\cdot f_{time}(B) + \gamma_v \cdot f_v(B) + \gamma_a \cdot f_a(B) \\ & + \gamma_{path}\cdot f_{path}(B) + \gamma_{ob}\cdot f_{ob}(B) + \gamma_{jerk}\cdot f_{jerk}(B) \\ & + \gamma_{sm}\cdot f_{sm}(B) + \gamma_{curv}\cdot f_{curv}(B) + \gamma_{ORCA}\cdot f_{ORCA}(B)\}. \end{aligned} \tag{29}$$

Among them, $f_{nh}(B)$ is the constraint function of nonholonomic kinematic constraint, γ_{nh} is the weight coefficient of nonholonomic kinematic constraint function; $f_{time}(B)$ is the constraint function of time optimal constraint, and γ_{time} is the weight coefficient of time optimal constraint function; $f_v(B)$ and $f_a(B)$ are the constraint functions of velocity and acceleration constraints, and γ_v and γ_a are the weight coefficients of the constraint functions of velocity and acceleration; $f_{path}(B)$ and $f_{ob}(B)$ are the constraint functions of path points and obstacles, and γ_{path} and γ_{ob} are the weight coefficients of the constraint functions of path points and obstacles. $f_{jerk}(B)$ is the constraint function of the acceleration constraint, and γ_{jerk} is the weight coefficient of the acceleration constraint function; $f_{sm}(B)$ is the constraint function of smoothness constraint, and γ_{sm} is the weight coefficient of smoothness constraint function; $f_{curv}(B)$ is the constraint function of curvature constraint, and γ_{curv} is the weight coefficient of curvature constraint function. $f_{ORCA}(B)$ is the constraint function of the optimal interactive collision avoidance model, and γ_{ORCA} is the weight coefficient of the constraint function of the optimal interactive collision avoidance model.

The traditional rule-based autonomous driving navigation scheme consists of sensing, localization, decision planning, and motion control modules. We incorporated the improved TEB-CA model into a conventional autopilot navigation scheme, as illustrated in Figure 8. The multi-objective detection and tracking module detects the position and velocity information of the surrounding moving obstacles in real-time. The ORCA model is used to predict the potential collision velocity range between the robot and obstacles based on the obtained obstacle information, which will then feed this velocity range into the proposed TEB-CA algorithm to generate an optimal collision-free trajectory by considering all constraints. Finally, it calculates control commands based on the planned trajectory sent to the robot chassis to achieve motion control of the robot.

V. EXPERIMENT AND DISCUSSIONS

To validate the effectiveness of the proposed Time Elastic Band for Collision Avoidance (TEB-CA) algorithm, we implemented and tested the system in both a simulation environment and a real experimental platform. The entire navigation framework was developed based on the *Robot Operating System* (*ROS*). The software part of the algorithm was implemented using the C++ programming language, and the overall framework inherited and modified the

standard TEB local path planning package [23]. The simulation platform was tested using *Gazebo* [24] and *Stage* [25] on Ubuntu 20.04, simulating two different experimental scenarios. The real-life test was conducted using our car-like mobile robot "*Little Ant*".

A. SIMULATION EXPERIMENTS IN THE GAZEBO ENVIRONMENT

Simulation tests were carried out on the Gazebo visualization platform using the *unified robot description format*(*URDF*) to build the *Ackermann* front-wheel steering car-like robot system, including the chassis, LIDAR, camera, and other parts of the structure. Table 1 lists the critical parameters of the simulation experiments.

TABLE 1. Parameter Settings of Robot Part.

Parameter	Value
Maximum linear velocity(m/s)	1.4
Maximum retreating linear velocity(m/s)	1
Maximum angular velocity(rad/s)	0.8
Maximum linear acceleration(m/s2)	1
Maximum angular acceleration(rad/s2)	0.5
Minimum obstacle distance(m)	0.5
Dynamic barrier expansion distance(m)	0.1
Dynamic obstacle weight	100

In this test, we chose a cluttered warehouse scene to bridge the gap between the simulation and real-world experiments.We chose a cluttered warehouse scene, as displayed in the Fig. 10 in which many shelves and cardboard boxes are randomly placed as static obstacles. Additionally, we used an *Actor Plugin* to create a dynamic model of nine walking or running people, wrote motion trajectory coordinates, and added collision attributes to pedestrians, which constitute dynamic obstacles. We presented a robot sensing system [26], [27] to obtain information about the position and speed of the moving obstacles.

FIGURE 10. Gazebo simulation experiment scenario.

We designed an experimental simulation approach, as demonstrated in Fig. 11. The mobile robot moves forward from the initial position to the target position and meets three obstacles in turn, with dynamic obstacles 01 and 02 crossing the robot's path vertically and dynamic obstacle 03 traveling opposite to the robot.

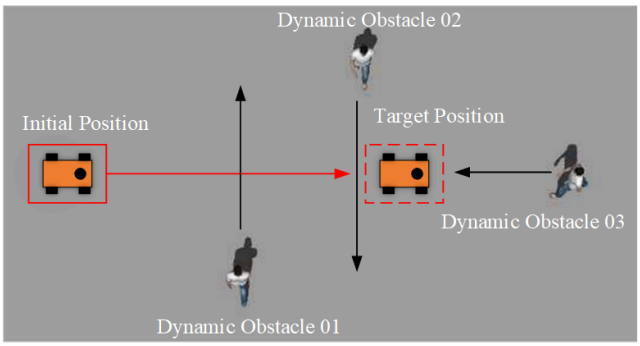


FIGURE 11. Schematic diagram of the simulation experiment method.

The experiment process is shown in Fig. 12. In Fig. 12(a), when the mobile robot is driving to the target position, the global path planned by the $A*$ algorithm [28], [29] is a straight line connecting the starting and ending points. In Fig. 12(b), pedestrian 01 walks from right to left. When the robot detects a dynamic obstacle, the speed decreases to avoid the obstacle's movement route. In Fig. 12(c), after the obstacle enters the non-collision speed zone, the robot accelerates from 0.1m/s to 0.6m/s. Going forward, pedestrian 02 walks from left to right; the mobile robot maintains a speed of about 0.4m/s to avoid pedestrian obstacles. In Fig. 12(d), when walking opposite pedestrian 03, the robot drives to the left to avoid and keep a safe distance from pedestrians.

Testing the improved TEB-CA algorithm against the conventional TEB and DWA algorithms, as shown in Fig. 13,14; When a car-like mobile robot using the traditional TEB algorithm encounters a dynamic obstacle, the robot's steering jumps and the planned path constantly shifts, resulting in the robot being unable to move in place or colliding with the obstacle. After the improved TEB-CA algorithm, when the perception module detects obstacles, the optimal interactive collision avoidance model calculates the speed area for avoiding collisions and then uses the time. The elastic band algorithm combines other constraints to calculate the optimal trajectory, executes a new planned trajectory, and avoids moving people. In addition, the planned trajectory does not change significantly while meeting the collision avoidance requirements, so the path length And the time consumed are also optimal.

B. SIMULATION EXPERIMENTS IN A STAGE ENVIRONMENT

We created a rectangular closed scene in the Stage simulation platform. In this scene, thirteen moving obstacles were distributed and moved back and forth between the two walls to test the avoidance of the robot in a denser and disordered dynamic obstacle scene. The car-like mobile robot navigates

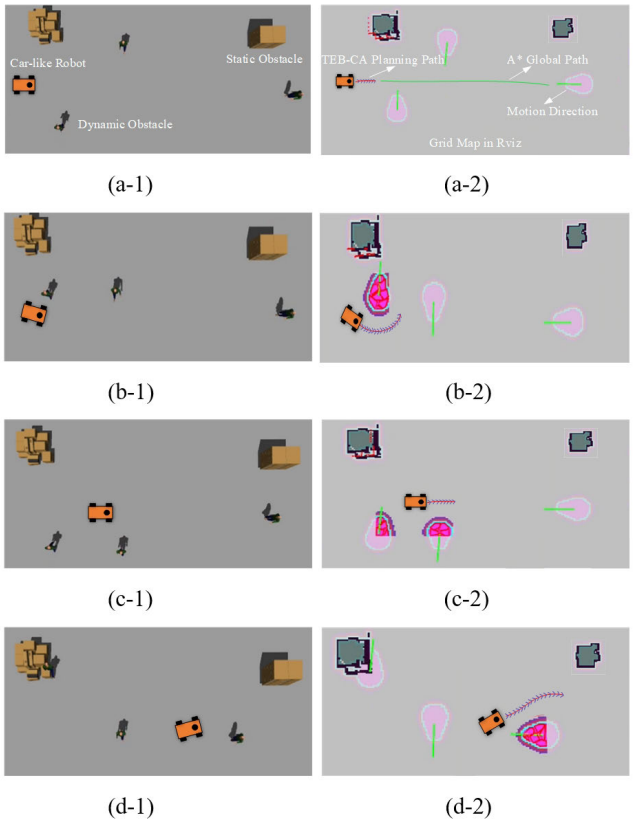


FIGURE 12. Experimental performance of the car-like robot simulation in the Gazebo environment, with the Gazebo dynamic obstacle simulation environment on the left and the Rviz visual graphical interface on the right.

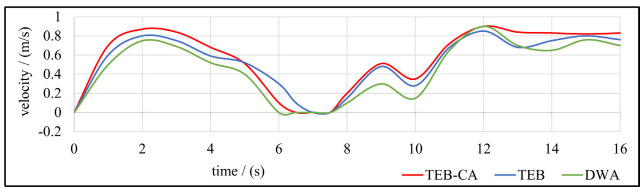


FIGURE 13. Linear velocity during robot motion.

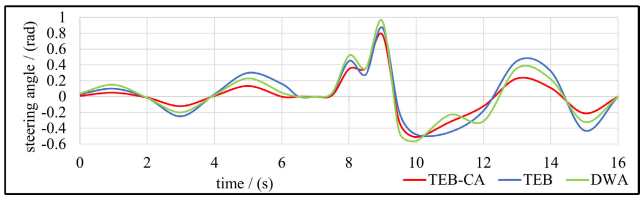


FIGURE 14. Angular velocity during robot motion.

from left to right and must pass through these dynamic obstacles moving back and forth.

In Fig. 15(a), the shortest path of global path planning for an autonomous mobile robot perception system when no moving obstacle is detected ahead is a straight line connecting the start and end points, i.e., the green line segment in Fig. 15(a-2). In Fig. 15(b), the sensing module acquires dynamic obstacle position and velocity information when the car-like robot continues to move and detects a

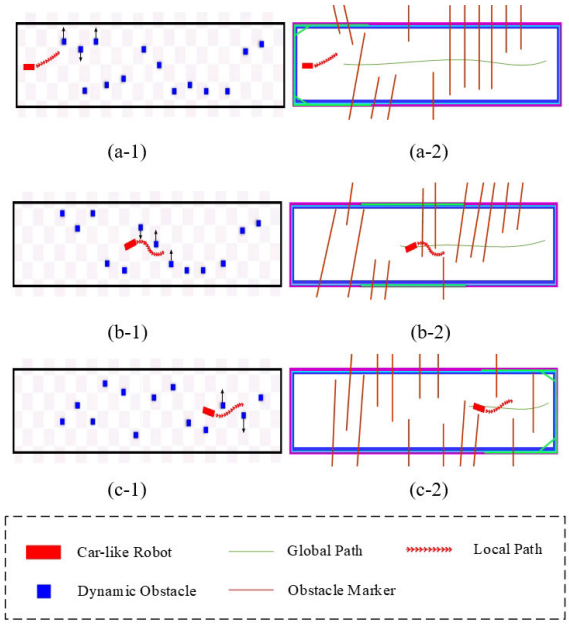


FIGURE 15. Experimental performance of the Car-like robot simulation in the Stage environment.

dynamic obstacle. The robot will only consider the movement of dynamic obstacles nearby. It will not perform velocity obstacle model prediction for obstacles that are too far away to reduce the amount of CPU computation. Generally, the robot has two choices when avoiding an approaching obstacle: accelerate through or stop and wait. The choice must be based on the position of the dynamic obstacle, its velocity information, and the robot's dynamic constraints to calculate the final motion speed. When encountering obstacles far away, as shown in Fig. 15(c), the optimal way for the robot is to change the speed and direction of motion at the same time, that is, the trajectory will be re-planned to bypass the obstacle without staying in place Waiting, this approach will reduce the time of the movement process and improve the efficiency of the robot. When an obstacle is approaching quickly, as shown in Fig. 15(b), the robot must collide with it. At this time, the robot's speed rapidly drops to close to 0, and it waits for the obstacle to pass first before continuing to move.

C. EXPERIMENTS IN REAL ENVIRONMENTS

In order to test the proposed hybrid motion planning algorithm TEB-CA, a car-like mobile robot was built. This section describes its hardware configuration and the various functional modules.For the motion planning algorithm in dynamic obstacle scenarios proposed in this paper, experimental analyses are carried out in several scenarios

1) EXPERIMENTAL PLATFORM

"*Little Ant*" is a rear-wheel drive, Ackerman front-wheel steering structure with the dimensions and dynamic constraints shown in Table 2. It has a *Robosense RS-Helios-16P* LiDAR, two *RICHBEAM* single-line LiDARs,

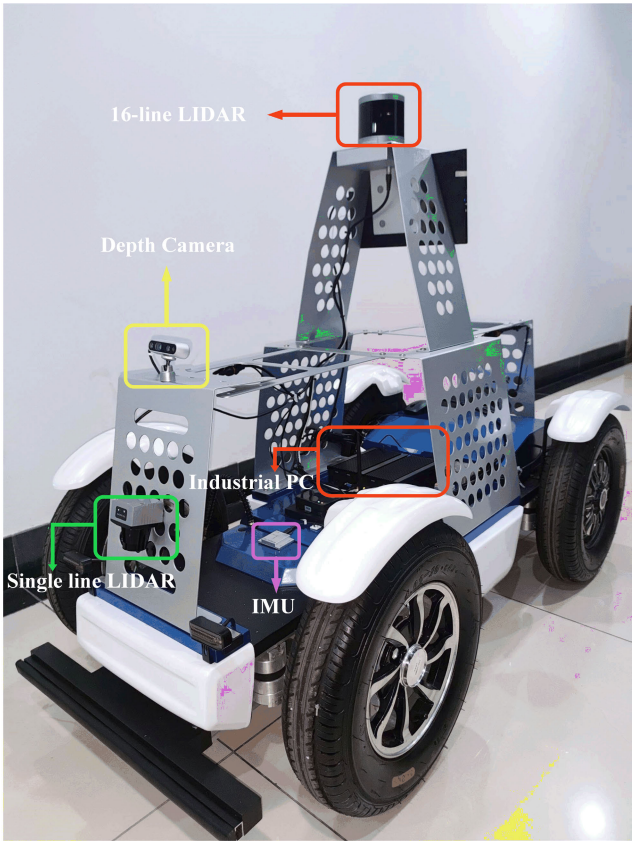


FIGURE 16. The Little Ant Car-like robot.

TABLE 2. Dimensions and Dynamic Parameters of "*Little Ant*".

Name	Value
Length(cm)	1620
Width(cm)	820
Height(cm)	1080
Maximum speed(m/s)	2.8
Turning radius(m)	2.54
Braking distance(m)	0.3
Climbing angle(%)	20
Gross chassis weight(kg)	87

Inter RealSense D435i depth camera, and 9-axis IMU for localization, obstacle detection, and state estimation. The "*Little Ant*" Robot processor consists of two modules, all software modules for perception, map building, and motion planning, running on a six-core 2.90 GHz IPC with 256 GB and 8 GB RAM, and a vehicle chassis motion controller, Cortex 32, which receives control commands from the upper-level autonomous driving system and controls actuator actions such as acceleration and braking.

2) EXPERIMENTAL SCENARIOS

A typical zigzag experimental test scenario was defined, containing two right-angle bends and a 5 × 20m open field with static obstacles such as display cabinets, tables and chairs, and walking students as dynamic obstacles to test the robot's performance in avoiding dynamic obstacles.

We use the method proposed in [30] and [31] to construct a point cloud map and locate it online, open a 3D point cloud map of the current scene in *Rviz*, use the *2D Pose Estimate* tool to select the approximate location and orientation of the current vehicle in the map, remotely move the vehicle by 1 2m, and simultaneously observe the change of the vehicle coordinate system in the visualization area to judge whether the initialized vehicle This is used to determine whether the vehicle's initial position is accurate. The 2D Nav Goal tool selects a target point on the map, and the $A*$ Global Path Planner will plan a global path from the starting point to the target point.

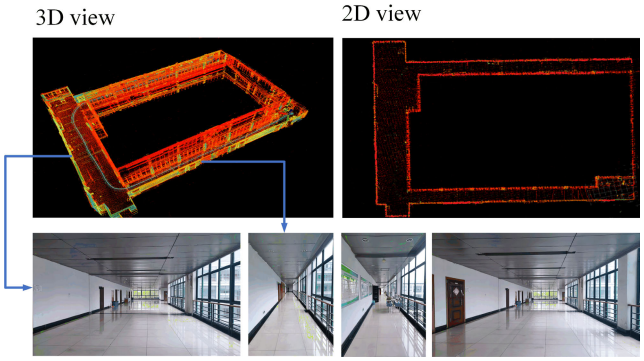


FIGURE 17. Map of the experimental test scenario and the 3D point cloud map created.

For the site conditions, two working conditions were designed to test the practical effects of the hybrid motion planning algorithm TEB-CA proposed in this thesis and to compare and analyze the performance of the traditional TEB and DWA algorithms on actual vehicles, where the experimental procedure and detailed data are shown below.

3) MULTIPLE DYNAMIC OBSTACLE CONDITIONS IN AN OPEN FIELD

The test method was modeled on that shown in Fig. 11, with one person walking opposite the robot and two people walking vertically crosswise with the robot in a $5\times20m$ rectangular field, both at a speed of around 0.6m/s. The TEB-CA and traditional TEB and DWA algorithms were used to replace the local path planner module in the program, respectively, and ten experimental obstacle avoidance tests were carried out. Experimental parameter data could be obtained using the *rqt* and *rosbag* tools in *ROS*, and the data were analyzed according to the evaluation metrics we developed. The results are shown in Table 3.

In order to evaluate the effectiveness and safety of navigation and obstacle avoidance for mobile robots, several evaluation metrics were developed to measure the

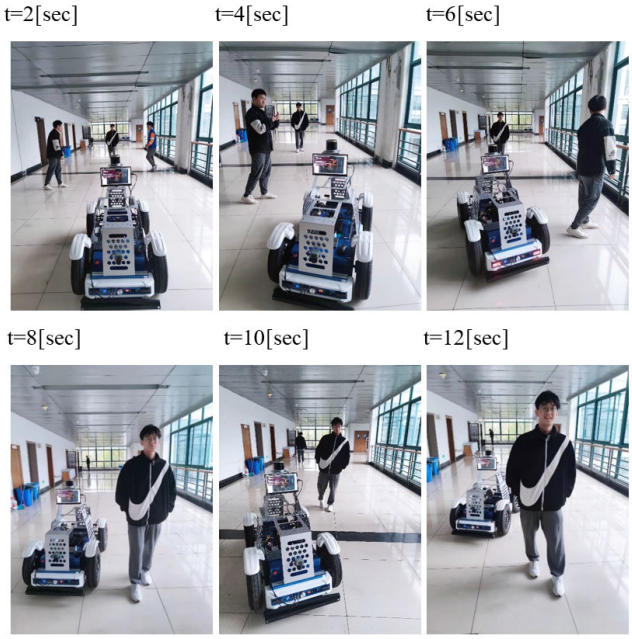


FIGURE 18. Mobile robot performing dynamic obstacle avoidance tests in an open field.

performance of different obstacle avoidance strategies, which are defined as follows.

- ***Total time spent:*** The total time taken by the mobile robot to successfully reach the target point from the starting point in a dynamic obstacle scenario.
- ***Average velocity:*** The average value of the linear velocity of the mobile robot during its movement.
- ***Extra time:*** The difference between the average time taken by the average time taken to reach the target point in the presence of dynamic obstacles minus the mobile robot to successfully reach the target point in the absence of dynamic obstacles, which can measure the efficiency of the robot in avoiding dynamic obstacles.
- ***Extra distance:*** The difference between the average distance required for the mobile robot to successfully reach the target point in the absence of dynamic obstacles minus the average distance to the target point in the presence of dynamic obstacles.
- ***Obstacle distance:*** The minimum distance between the mobile robot and the obstacle during obstacle avoidance.

From the index data in Table 3 and Fig. 19, it can be seen that the DWA algorithm has poor obstacle avoidance effect, and there are many collisions with dynamic obstacles, which is not applicable to the Ackermann steering four-wheeled vehicle model, and the motion planner of this paper's algorithm, TEB-CA, has an average total time increase of 13.28% compared with the traditional TEB algorithm, the extra time used for obstacle avoidance has an increase

TABLE 3. Results of dynamic obstacle avoidance experiments.

Algorithm	Evaluation metrics	Exp. 1	Exp.2	Exp.3	Exp.4	Exp.5	Exp.6	Exp.7
DWA	Total-time-spent	19.19	18.38	18.13	18.85	19.83	19.35	18.87
	Average velocity	0.62	0.65	0.66	0.63	0.60	0.62	0.63
	Extra time	13.19	12.38	12.13	12.85	13.83	13.35	12.87
	Extra distance	8.35	9.21	8.16	7.98	7.89	7.80	7.56
	Obstacle distance	0.38	0.00	0.41	0.40	0.47	0.38	0.46
TEB	Total-time-spent	16.41	17.95	16.26	17.45	17.23	15.77	16.99
	Average velocity	0.73	0.66	0.73	0.68	0.69	0.76	0.70
	Extra time	10.41	11.95	10.26	11.45	11.23	9.77	10.99
	Extra distance	7.88	8.27	9.34	8.68	8.67	8.07	7.96
	Obstacle distance	0.29	0.30	0.00	0.32	0.23	0.27	0.36
TEB-CA	Total time spent	15.99	14.91	14.34	14.04	14.28	14.60	14.23
	Average velocity	0.75	0.80	0.83	0.85	0.84	0.82	0.84
	Extra time	9.99	8.91	8.34	8.04	8.28	8.60	8.23
	Extra distance	6.31	5.18	5.34	6.04	5.64	6.24	5.69
	Obstacle distance	0.30	0.33	0.23	0.30	0.22	0.25	0.31

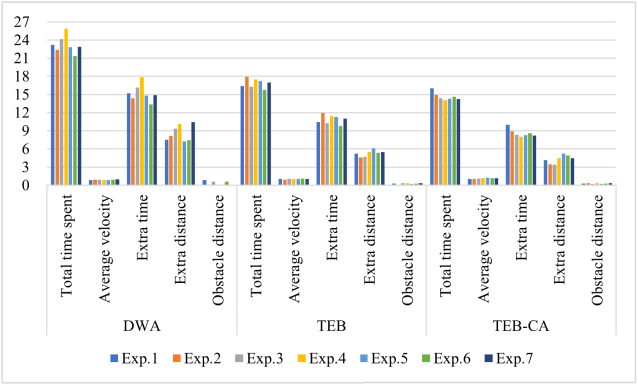


FIGURE 19. Dynamic obstacle avoidance test results in an open field.

of 20.6% compared with the traditional TEB algorithm, and the speed and angle of the fluctuation is normal and control is stable, which can ensure that no contact collision with obstacles occurs in the obstacle avoidance process, the planned travelling path meets the requirements of mobile robot dynamics constraints, and the robot's various state quantities change smoothly, which guarantees the stability of manipulation.

4) DYNAMIC OBSTACLE CONDITIONS IN NARROW CORRIDORS

The mobile robot plays a confrontational game with pedestrians in the corridor as it walks straight ahead in the opposite direction, testing its precise obstacle avoidance and behavioral decision-making capabilities.

Unlike avoiding dynamic obstacles in an open space, in a narrow passage, subject to walls and pedestrians, the robot can move within a small range. When it walks opposite to pedestrians, it can be seen from Fig. 20,21,22 that the mobile robot is in the 7th second. At a standstill, when a pedestrian moves to the side, the mobile robot immediately drives to the left to avoid, and the minimum contact distance with the pedestrian is about 0.23m.

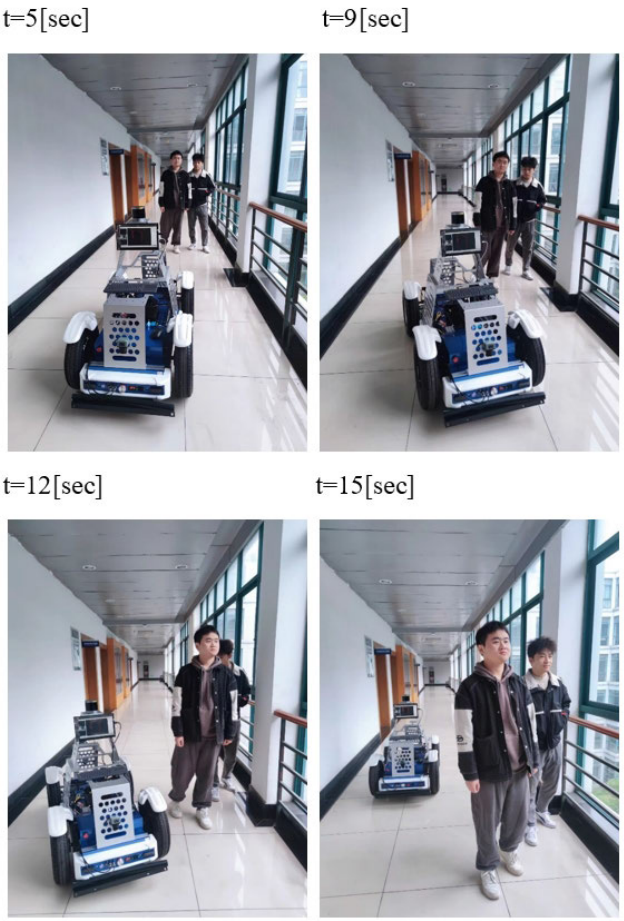


FIGURE 20. Actual photograph of the dynamic obstacle avoidance test in a narrow corridor.

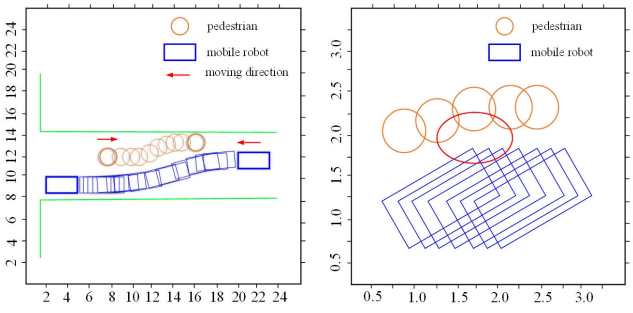


FIGURE 21. Left diagram of the whole obstacle avoidance process, right diagram is a partial enlargement of the minimum distance between the mobile robot and the dynamic obstacle.

D. DISCUSSION

The problem of motion planning for car-like mobile robots in dynamic obstacle environments has long received research attention. Researchers such as [3], [5], and [11] solutions prefer the perception module to detect the position information of obstacles promptly and feed it back to the motion planning module in the form of a cost map.

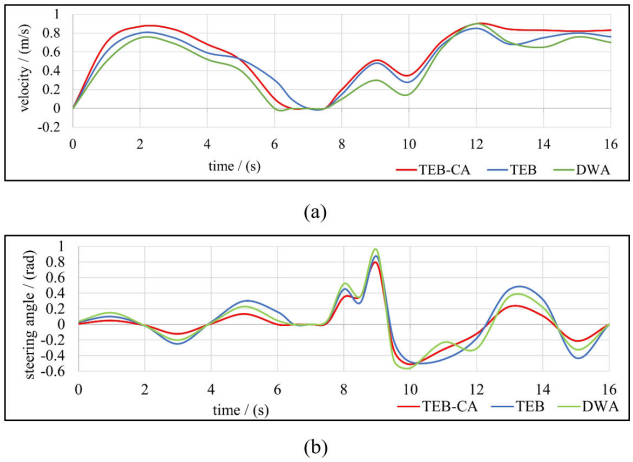


FIGURE 22. Graph of the speed and steering angle changes of the mobile robot during the obstacle avoidance process.

Our innovative way is to add the Optimal Reciprocal Collision Avoidance model to the traditional motion planning algorithm TEB, and obtain the speed information of obstacles, to strengthen the TEB algorithm's ability to predict the future motion trajectory of dynamic blocks and make timely decisions. In addition, the algorithm in this paper focuses on solving the problem of dynamic obstacle avoidance by optimization and modeling, which is more friendly and saves hardware costs for transplanting to actual vehicles for mass production.

It should be noted that this study only examined a car-like mobile robot with incomplete constrained *Ackerman* front wheel steering, and no verification was made for two-wheeled or other forms of robots. In addition, the dynamic obstacle pedestrian movement patterns and speeds in the experimental section were relatively homogeneous, and further experiments and tests are needed for more complex and diverse dynamic obstacle environments.

VI. CONCLUSION

We verified the effectiveness and safety of the method through real experiments. In order to verify the motion planning algorithm of autonomous mobile robot under dynamic obstacle scene proposed in this paper, two dynamic scenes are built in Gazebo and Stage simulation tools for experiments, comparing with traditional TEB and DWA algorithms, it performs better in the success rate of obstacle avoidance and efficiency, and meanwhile, test and analyse the algorithms in the real scene. In addition, the autonomous mobile robot can effectively avoid any collision contact with dynamic obstacles during the movement process, which effectively solves the "freezing" problem of the mobile robot in the dynamic obstacle environment.

Future work and plans include experimental tests on more complex and diverse dynamic obstacle environments, as well as exploring the integration of dynamic obstacle trajectory prediction from the sensing system and an intelligent decision-making system based on deep reinforcement learning. Since the programme is applicable to car-like mobile robots, further extension of the programme to self-driving cars is promising and important..

REFERENCES

[1] G. Demesure, M. Defoort, A. Bekrar, D. Trentesaux, and M. Djemaï, "Decentralized motion planning and scheduling of AGVs in an FMS," *IEEE Trans. Ind. Informat.*, vol. 14, no. 4, pp. 1744–1752, Apr. 2018.

[2] Y. Jiang, F. Yang, S. Zhang, and P. Stone, "Task-motion planning with reinforcement learning for adaptable mobile service robots," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Nov. 2019, pp. 7529–7534.

[3] Y. Zhang, Z. Xiao, X. Yuan, S. Li, and S. Liang, "Obstacle avoidance of two-wheeled mobile robot based on DWA algorithm," in *Proc. Chin. Autom. Congr. (CAC)*, Nov. 2019, pp. 5701–5706.

[4] J.-F. Duhé, S. Victor, and P. Melchior, "ContribUtions on artificial potential field method for effective obstacle avoidance," *Fractional Calculus Appl. Anal.*, vol. 24, no. 2, pp. 421–446, Apr. 2021.

[5] J. Xu and K.-S. Park, "Moving obstacle avoidance for cable-driven parallel robots using improved RRT," *Microsyst. Technol.*, vol. 27, no. 6, pp. 2281–2292, Jun. 2021.

[6] A. S. Lafmejani and S. Berman, "Nonlinear MPC for collision-free and deadlock-free navigation of multiple nonholonomic mobile robots," *Robot. Auto. Syst.*, vol. 141, Jul. 2021, Art. no. 103774.

[7] C. Miao, G. Chen, C. Yan, and Y. Wu, "Path planning optimization of indoor mobile robot based on adaptive ant colony algorithm," *Comput. Ind. Eng.*, vol. 156, Jun. 2021, Art. no. 107230.

[8] S. M. H. Rostami, A. K. Sangaiah, J. Wang, and X. Liu, "Obstacle avoidance of mobile robots using modified artificial potential field algorithm," *EURASIP J. Wireless Commun. Netw.*, vol. 2019, no. 1, pp. 1–19, Dec. 2019.

[9] R. Karima and B. Samira, "EKF-DWA for car-like robot navigation in presence of moving obstacles," *Int. J. Comput. Digit. Syst.*, vol. 12, no. 1, pp. 205–214, Jul. 2022.

[10] B. Guo, N. Guo, and Z. Cen, "Motion saliency-based collision avoidance for mobile robots in dynamic environments," *IEEE Trans. Ind. Electron.*, vol. 69, no. 12, pp. 13203–13212, Dec. 2022.

[11] C. Roesmann, W. Feiten, T. Woesch, F. Hoffmann, and T. Bertram, "Trajectory modification considering dynamic constraints of autonomous robots," in *Proc. 7th German Conf. Robot. (ROBOTIK)*, May 2012, pp. 1–6.

[12] C. Rösmann, F. Hoffmann, and T. Bertram, "Planning of multiple robot trajectories in distinctive topologies," in *Proc. Eur. Conf. Mobile Robots (ECMR)*, Sep. 2015, pp. 1–6.

[13] C. Rösmann, F. Hoffmann, and T. Bertram, "Integrated online trajectory planning and optimization in distinctive topologies," *Robot. Auto. Syst.*, vol. 88, pp. 142–153, Feb. 2017.

[14] X. Sun, S. Deng, T. Zhao, and B. Tong, "Motion planning approach for car-like robots in unstructured scenario," *Trans. Inst. Meas. Control*, vol. 44, no. 4, pp. 754–765, Feb. 2022.

[15] S. Liu, F. Dai, S. Zhang, Y. Wang, and Z. Wang, "Trend-aware motion planning for wheeled mobile robots operating in dynamic environments," *Int. J. Adv. Robotic Syst.*, vol. 17, no. 4, Jul. 2020, Art. no. 172988142092529.

[16] T. Fan, X. Cheng, J. Pan, P. Long, W. Liu, R. Yang, and D. Manocha, "Getting robots unfrozen and unlost in dense pedestrian crowds," *IEEE Robot. Autom. Lett.*, vol. 4, no. 2, pp. 1178–1185, Apr. 2019.

[17] P. Fiorini and Z. Shiller, "Motion planning in dynamic environments using velocity obstacles," *Int. J. Robot. Res.*, vol. 17, no. 7, pp. 760–772, Jul. 1998.

[18] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "g^2o: A general framework for graph optimization," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2011, pp. 3607–3613.

[19] J. van den Berg, M. Lin, and D. Manocha, "Reciprocal velocity obstacles for real-time multi-agent navigation," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2008, pp. 1928–1935.

[20] J. Snape, J. Van Den Berg, S. J. Guy, and D. Manocha, "The hybrid reciprocal velocity obstacle," *IEEE Trans. Robot.*, vol. 27, no. 4, pp. 696–706, Aug. 2011.

[21] J. Van Den Berg, S. J. Guy, M. Lin, and D. Manocha, "Reciprocal *n*-body collision avoidance," in *Proc. 14th Int. Symp. Robot. Res. (ISRR)*. Cham, Switzerland: Springer, 2011, pp. 3–19.

[22] D. Wilkie, J. van den Berg, and D. Manocha, "Generalized velocity obstacles," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2009, pp. 5573–5578.

[23] C. Rösmann, F. Hoffmann, and T. Bertram, "Online trajectory planning in ROS under kinodynamic constraints with timed-elastic-bands," in *Robot Operating System (ROS): The Complete Reference (Volume 2)*. Cham, Switzerland: Springer, 2017, pp. 231–261.

[24] D. Chikurtev, "Mobile robot simulation and navigation in ROS and gazebo," in *Proc. Int. Conf. Automatics Informat. (ICAI)*, Oct. 2020, pp. 1–6.

[25] R. P. Srivastava, A. Tripathi, A. K. Singh, and R. Singh, "Robotics concurrency: A holistic implementation using actors and promises," in *Proc. Int. Conf. Comput. Intell. Sustain. Eng. Solutions (CISES)*, May 2022, pp. 154–159.

[26] S. Macenski, D. Tsai, and M. Feinberg, "Spatio-temporal voxel layer: A view on robot perception for the dynamic world," *Int. J. Adv. Robotic Syst.*, vol. 17, no. 2, Mar. 2020, Art. no. 172988142091053.

[27] Z. Li, B. Li, Q. Liang, W. Liu, L. Hou, and X. Rong, "A quadruped robot obstacle avoidance and personnel following strategy based on ultra-wideband and three-dimensional laser radar," *Int. J. Adv. Robotic Syst.*, vol. 19, no. 4, Jul. 2022, Art. no. 172988062211147.

[28] Z. Liu, H. Liu, Z. Lu, and Q. Zeng, "A dynamic fusion pathfinding algorithm using Delaunay triangulation and improved A-star for mobile robots," *IEEE Access*, vol. 9, pp. 20602–20621, 2021.

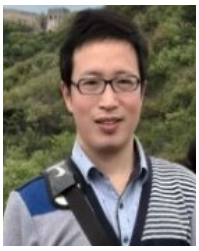
[29] S. Erke, D. Bin, N. Yiming, Z. Qi, X. Liang, and Z. Dawei, "An improved A-star based path planning algorithm for autonomous land vehicles," *Int. J. Adv. Robotic Syst.*, vol. 17, no. 5, Sep. 2020, Art. no. 172988142096226.

[30] T. Shan, B. Englot, D. Meyers, W. Wang, C. Ratti, and D. Rus, "LIO-SAM: Tightly-coupled LiDAR inertial odometry via smoothing and mapping," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Oct. 2020, pp. 5135–5142.

[31] H. T. Warku, N. Y. Ko, H. G. Yeom, and W. Choi, "Three-dimensional mapping of indoor and outdoor environment using LIO-SAM," in *Proc. 21st Int. Conf. Control, Autom. Syst. (ICCAS)*, Oct. 2021, pp. 1455–1458.

ZHIWEI WANG was born in Xuzhou, Jiangsu, China. He received the bachelor's degree in vehicle engineering from the Nantong Institute of Technology. He is currently pursuing the master's degree with the School of Mechanical and Energy Engineering, Zhejiang University of Science and Technology, China.

His research interests include vehicle control, motion planning technology, and SLAM technology of autonomous driving and autonomous mobile robots.



PEIQING LI received the Ph.D. degree in engineering from Southeast University.

He was a Master Tutor. He was also a Postdoctoral Fellow from Zhejiang University. He mainly engaged in vehicle dynamics, intelligent transportation, unmanned driving, vehicle-road coordination, road traffic safety, and other interdisciplinary teaching and research work. In recent five years, more than 20 papers have been published in international academic journals and important academic conferences, of which more than ten papers have been retrieved by SCI/EI and participated in the compilation of a monograph. He has presided over and participated in more than ten scientific research projects, applied for 15 national invention patents, and authorized three projects. He is currently in charge of one National Natural Science Foundation, one Zhejiang Natural Science Foundation, one postdoctoral fund, and several horizontal projects. He is also a special reviewer of several international SCI/EI journals.

QIPENG LI received the Ph.D. degree in engineering from Zhejiang University, in 2005.

He was a Professor and the Dean of the School of Mechanical and Energy Engineering (formerly ''School of Mechanical and Automotive Engineering''). He went to the Fraunhofer Association, Germany, the Italian Energy Economic Research Center, and the Ukrainian National Technical University for academic exchanges and visits. His research interests include intelligent special electromechanical equipment, intelligent assembly and detection systems, and electro-hydraulic servo/proportional control technology.

ZHONGSHAN WANG received the B.E. degree from the Zhejiang University of Science and Technology, Hangzhou, China, in 2020, where he is currently pursuing the master's degree with the Department of Automotive Engineering.

His research interests include reinforcement learning, decision making, and path planning technology of autonomous vehicle.

ZHUORAN LI received the B.E. degree from Shandong Agriculture and Engineering University, Jinan, China, in 2020. He is currently pursuing the master's degree with the Faculty of Information Technology, City University Malaysia, Petaling Jaya, Malaysia.

His research interests include intelligent driving, the Internet of Things, reinforcement learning, decision making, and path planning technology of autonomous vehicle.