Robotica (2024), **42**, pp. 2991–3004
doi:10.1017/S026357472400122X

CAMBRIDGE UNIVERSITY PRESS

RESEARCH ARTICLE

Obstacle avoidance control of UGV based on adaptive-dynamic control barrier function in unstructured terrain

Liang Guo[1], Suyu Zhang[1], Wenlong Zhao[1], Jun Liu[1] and Ruijun Liu[2]

[1]College of Information Engineering, Nanchang Hangkong University, Nanchang, China
[2]Jiangxi DonGRUI MACHINERY CO. LTD. Nanchang, China
Corresponding author: Suyu Zhang; Email: 1228474146@qq.com

Received: 25 March 2024; **Revised:** 15 June 2024; **Accepted:** 3 July 2024; **First published online:** 18 September 2024

Keywords: navigation in complex environments; collision avoidance; obstacle recognition; control barrier functions; model predictive control

Abstract

The widely used model predictive control of discrete-time control barrier functions (MPC-CBF) has difficulties in obstacle avoidance for unmanned ground vehicles (UGVs) in complex terrain. To address this problem, we propose adaptive dynamic control barrier functions (AD-CBF). AD-CBF is able to adaptively select an extended class of functions of CBF to optimize the feasibility and flexibility of obstacle avoidance behaviors based on the relative positions of the UGV and the obstacle, which in turn improves the obstacle avoidance speed and safety of the MPC algorithm when integrated with MPC. The algorithmic constraints of the CBF employ hierarchical density-based spatial clustering of applications with noise (HDBSCAN) for parameterization of dynamic obstacle information and unscaled Kalman filter (UKF) for trajectory prediction. Through simulations and practical experiments, we demonstrate the effectiveness of the AD-CBF-MPC algorithm in planning optimal obstacle avoidance paths in dynamic environments, overcoming the limitations of the point-by-point feasibility of MPC-CBF.

1. Introduction

As unmanned ground vehicles (UGVs) are increasingly used in complex and open terrains, especially in unstructured terrains (e.g., mines, forests, deserts, etc.), 2D raster map navigation methods commonly used in indoor environments are no longer applicable. In addition, it is challenging to accurately parameterize the obstacle point cloud and predict the motion trajectory during navigation, which involves a large amount of data with a high degree of uncertainty. This leads to difficulties in ensuring safety and speed in practical applications, especially in dynamic obstacle avoidance in unstructured terrain [1]. Thus, this paper is dedicated to proposing a 3D point cloud dynamic obstacle avoidance strategy for unstructured terrain, which enables UGVs to have all-terrain autonomous navigation capability.

As the manufacturing costs of LiDAR decrease, it is increasingly being researched for its use in obstacle detection and tracking. MacLachlan et al. [2] introduced feature-based Kalman filter tracking, Vatavu et al. [3] proposed a particle filtering (PF)-based obstacle estimation method, and Maurer et al. [4] expanded upon this with a hierarchical system using adaptive ground evaluation and interactive multi-model extended Kalman filtering (IMM-EKF). Brito et al. [5] used linear Kalman filtering (KF) for trajectory prediction of a moving obstacle modeled as a minimal Minkowski sum ellipsoid. Jian et al. [6] built on this, combining density-based spatial clustering of applications with noise (DBSCAN) and KF for obstacle trajectory prediction, evaluating prediction results with uncertainty.

In terms of the method to deal with the problem of vehicle obstacle avoidance, He S et al. [7] proposed a prescribed performance control strategy to ensure that the error between vehicles converges faster than

© The Author(s), 2024. Published by Cambridge University Press.

https://doi.org/10.1017/S026357472400122X Published online by Cambridge University Press

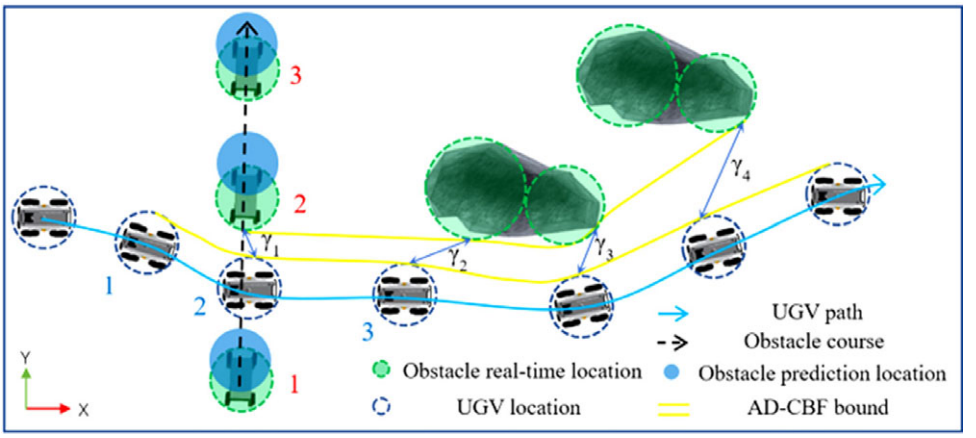


Figure 1. *Schematic diagram of the AD-CBF-MPC algorithm.*

the predetermined value, which ensures the anti-collision and connectivity between adjacent vehicles, and also ensures the obstacle avoidance of static obstacles. However, it does not solve the problem of fault-tolerant control of uncertain marine surface vehicle (MSV) with unknown inertia matrix under random disturbance. Compared to conventional dynamic obstacle avoidance methods, such as the artificial potential field [8] and the dynamic window approach (DWA) [9], model predictive control (MPC) has better predictive, constraint handling, and mathematical optimization capabilities. Additionally, it can handle complex obstacle avoidance problems with high nonlinearity and constraints, despite having higher computational complexity, ultimately achieving superior overall performance. It is now widely used in robot operating systems [10, 11]. Zeng et al. [12] developed a safety-centric MPC strategy using a discrete-time control barrier function (CBF) or MPC-CBF, but found it to underperform in dynamic environments. Jian et al. [6] proposed a dynamic control barrier function (D-CBF), integrating barrier uncertainty into MPC optimization, improving system safety and enabling dynamic obstacle avoidance. However, the direct approach of using a constant as an extend class K_∞ function γ of the CBF will affect the flexibility and feasibility of the system during operation.

Addressing identified issues, we introduce an adaptive dynamic control barrier function model predictive control (AD-CBF-MPC) method. This algorithm, depicted in Figure 1, uses blue and red numbers to represent UGV and obstacles, respectively, with positions incrementing over time. AD-CBF adapts feasible domains based on obstacle positions and sizes, reflected in the γ value. We use a 2D Gaussian function, considering the UGV-obstacle distance and obstacle size, to replace the constant CBF's K_∞ function γ, improving system feasibility during obstacle avoidance. We employ LiDAR for obstacle state information, stabilize the obstacle point cloud using smoothing filtering and hierarchical density-based spatial clustering of applications with noise (HDBSCAN), and propose unscented Kalman filtering (UKF) for tracking and predicting obstacle states. This information is integrated into the MPC optimization solution constraints, leading to a reliable UGV obstacle avoidance system.

The contributions of this work are summarized as follows:

- An effective solution is proposed for the state identification and prediction of complex random obstacles. The method innovatively utilizes HDBSCAN to realize real-time parameterization of obstacle states, and then passes into the UKF state prediction algorithm to complete the advanced prediction of obstacle trajectories.
- In contrast to the CBF-MPC methods in [6] and [12], the system usually has only point-by-point feasibility for a fixed γ. In this paper, we propose an adaptive control barrier function (A-CBF), which combines the correlation of each physical quantity between the unmanned vehicle and the

https://doi.org/10.1017/S026357472400122X Published online by Cambridge University Press

obstacle in obstacle avoidance in the form of a 2D Gaussian function to obtain the formula as the value of γ, to realize the lasting feasibility of the system.

- Combining the first contribution and the second contribution, the current state and the predicted state of the obstacle obtained by the HDBSCAN-UKF algorithm are used as the dynamic obstacle avoidance constraints of the A-CBF method, and the adaptive dynamic control barrier function (AD-CBF) is proposed, which takes into account the uncertainty of the dynamic obstacle and improves the safety of the obstacle avoidance.

2. Preliminaries

This section defines the kinematic model of the UGV and presents the optimal control problem to be solved. Additionally, it rewrites the optimization form of the MPC-CBF to prepare for the AD-CBF-MPC algorithm presentation in Section 3.3.

If MPC is employed for controlling vehicle motion, a vehicle kinematics or dynamics model that is reasonably accurate must be developed. Since the application scenario of this study involves low-speed automated driving, utilizing solely the vehicle kinematics model is adequate. Additionally, the commonly utilized chassis for UGVs are the Ackermann and the four-wheel differential. Previous studies have widely discussed applying MPC on the Ackermann chassis [13]; however, this research specifically focuses on the application of MPC and its optimization algorithms on a four-wheel differential chassis. It should be noted that the results presented in this study are equally applicable to other types of UGV chassis. For the discrete system, the dynamics model of the UGV can be described as $\boldsymbol{x}_{t+1} = f(\boldsymbol{x}_t, \boldsymbol{u}_t)$, where $\boldsymbol{x}_t \in X \subset \mathbb{R}^n$ is the state of the system at time $t \in \mathbb{Z}^+$, $\boldsymbol{u}_t \in U \subset \mathbb{R}^m$ denotes the control input, and to simplify the model, the four-wheel differential chassis is restricted to keeping the same speed of the same side wheels, which is equivalent to a two-wheel differential model. The final kinematic model of the UGV is:

$$\boldsymbol{x}_{t+1} = f\left(\boldsymbol{x}_t, \boldsymbol{u}_t\right) = \boldsymbol{x}_t + \begin{bmatrix} \cos\theta_t & 0 \\ \sin\theta_t & 0 \\ 0 & 1 \end{bmatrix} \boldsymbol{u}_t \Delta t \tag{1}$$

where $\boldsymbol{x}_t = [x_t, y_t, \theta_t]^T$ contains the position and cornering state of the UGV at moment t, and $\boldsymbol{u}_t = [v_t, \omega_t]^T$ contains the linear and angular velocity inputs to the system at moment t.

Thus, the MPC optimal control problem can be expressed as [14, 15]:

$$J_t^*\left(\boldsymbol{x}_t\right) = \min_{\boldsymbol{u}_{t:\ t+N-1|t}} p\left(\boldsymbol{x}_{t+N|t}\right) + \sum_{k=0}^{N-1} q\left(\boldsymbol{x}_{t+k|t}, \boldsymbol{u}_{t+k|t}\right) \tag{2a}$$

$$s.t.\ \boldsymbol{x}_{t+k+1|t} = f\left(\boldsymbol{x}_{t+k|t}, \boldsymbol{u}_{t+k|t}\right), k = 0, \ldots, N-1 \tag{2b}$$

$$\boldsymbol{x}_{t+k|t} \in X, \boldsymbol{u}_{t+k|t} \in U, k = 0, \ldots, N-1 \tag{2c}$$

$$\boldsymbol{x}_{t|t} = \boldsymbol{x}_t, \boldsymbol{x}_{t+N|t} \in X_f \tag{2d}$$

$$g\left(\boldsymbol{x}_{t+k|t}\right) \geq 0, k = 0, \ldots, N-1 \tag{2e}$$

here $\boldsymbol{x}_{t+k|t}$ denotes the state vector at time step $t+k$ predicted at time step t obtained by starting from the current state $\boldsymbol{x}_t$ (2d), and applying the input sequence $\boldsymbol{u}_{t:\ t+N-1|t}$ to the system dynamics (2b), $q(\boldsymbol{x}_{t+k|t}, \boldsymbol{u}_{t+k|t})$ as stage cost, $p(\boldsymbol{x}_{t+N|t})$ as terminal cost. (2b) is the system dynamics constraint, (2c) encompasses permissible system and input states, and (2d) outlines initial and terminal states. (2e), the collision avoidance constraint, uses N as prediction step size and $g(\cdot)$ as the Euclidean constraint, abbreviating the method as MPC-DC.

https://doi.org/10.1017/S026357472400122X Published online by Cambridge University Press

In MPC-DC, Euclidean-based constraints initiate UGV motion regulation near obstacles. To start avoidance from a greater distance, larger obstacle avoidance fields of view are needed in MPC, but this increases optimal solution time. To solve this, CBF constraints have been proposed for MPC, providing a larger, set-invariant obstacle avoidance field and addressing the set-invariant optimal obstacle avoidance issue with an overshooting field of view [16, 17].

Define the safe set C of the optimal control as the set of superlevels of a continuously differentiable function h: $X \subset \mathbb{R}^n \to \mathbb{R}$:

$$C = \{\boldsymbol{x} \in X \subset \mathbb{R}^n : h(\boldsymbol{x}) \geq 0\}. \tag{3}$$

if for all $\boldsymbol{x} \in \partial C$ there is $\partial h/\partial \boldsymbol{x} \neq 0$ and for system (1) there exists an extend class of K_∞ functions γ, then h is a CBF function which h satisfies:

$$\exists u \, s.t. \, \dot{h}(\boldsymbol{x}, \boldsymbol{u}) \geq -\gamma(h(\boldsymbol{x})), \ \gamma \in K_\infty \tag{4}$$

Generalization to discrete-time domain:

$$\Delta h(\boldsymbol{x}_k, \boldsymbol{u}_k) \geq -\gamma(h(\boldsymbol{x}_k)), \ \gamma \in K_\infty. \tag{5}$$

where $\Delta h(\boldsymbol{x}_k, \boldsymbol{u}_k) = h(\boldsymbol{x}_{k+1}) - h(\boldsymbol{x}_k)$. Here instead of defining γ as a scalar, as in the literature [6] and [12], it is defined as a K_∞ function. For the discrete-time domain, it can be assumed that $0 < \gamma(h(\boldsymbol{x}_k)) \leq h(\boldsymbol{x}_k) \leq 1$ is satisfied for any $h(\boldsymbol{x}_k)$.

Thus, for the definition of the MPC-CBF optimal control problem the collision avoidance control constraint (2e) in MPC-DC can be replaced by:

$$\Delta h(\boldsymbol{x}_k, \boldsymbol{u}_k) \geq -\gamma(h(\boldsymbol{x}_k)), k = 0, \ldots, N-1. \tag{6}$$

3. Implementation

Existing obstacle trajectory prediction algorithms work best in structured settings with simple kinematic models. We propose a novel method for complex, unstructured environments, such as forests or mines. Knowledge of obstacle trajectories can boost UGV obstacle avoidance reaction speed, improving safety. Our proposed HDBSCAN-UKF trajectory prediction method addresses these needs.

Traditional MPC algorithms with Euclidean constraints are slow and ineffective for dynamic obstacles, demanding more adaptable avoidance constraints. We initially merge MPC with CBF for improved static and partial dynamic obstacle avoidance. Then, we introduce and incorporate the AD-CBF into the MPC, enabling rapid and stable dynamic obstacle avoidance.

3.1. HDBSCAN obstacle recognition

HDBSCAN, a refinement of DBSCAN, creates a hierarchical structure instead of choosing clusters based on a global core point radius *Eps* threshold. The minimal cluster size is determined by integrating all possible *Eps* values with the minimum number of covered points *MinPts*, reducing the impact of clustering parameters on DBSCAN's clustering accuracy [18]. Unlike DBSCAN, HDBSCAN does not identify each trajectory point but selects some core points for cluster generation. This decreases the number of queries and eases data analysis and processing.

HDBSCAN defines the core distance ($d_{mreach-k}$) as the distance between an object and its *MinPts* closest neighbors. A hierarchy can be constructed based on two objects using a distance of:

$$d_{mreach-k}(a, b) = \max\{core_k(a), core_k(b), d(a, b)\} \tag{7}$$

where $d(a, b)$ denotes the Euclidean distance according to the chosen measure. $core_k(a)$ denotes the Euclidean distance from point a to the kth core point [19].

HDBSCAN provides numerous benefits. It retains DBSCAN's advantages and is better suited for high-dimensional, complex datasets. HDBSCAN requires only the minimum cluster size, eliminating

https://doi.org/10.1017/S026357472400122X Published online by Cambridge University Press

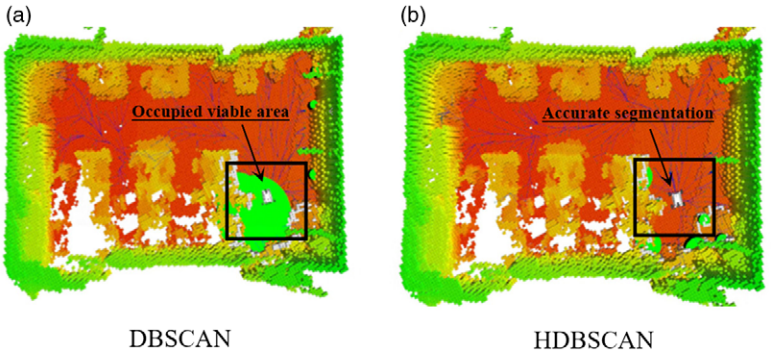


Figure 2. *(a) Shows the result of obstacle recognition using the DBSCAN algorithm, while (b) Displays the result using the HDBSCAN algorithm.*

the need for manual parameter setting as it automatically uncovers clustering clusters and adapts to the dataset. This makes HDBSCAN useful for segmenting nearby obstacles and a sensible choice for recognizing obstacles with significant length-width ratio differences.

Performing the HDBSCAN clustering process on the set of obstacles point cloud ($C = \{X, Y, Z\}$) input from LiDAR generates a set ($O^c_{obs} = \{O^c_1, O^c_2, \ldots, O^c_n\}$) of current states of n clusters of obstacles, each including the position and radius of the obstacle.

A comparison of the two algorithms‘ recognition effects is shown in Figure 2, where the green cylinder represents the obstacle's position and size as determined by each algorithm. Figure 2a shows that DBSCAN struggles with irregular obstacles, like complex stacking, leading to over-inflation of the green cylinder and occupying the feasible region. Conversely, the HDBSCAN results, depicted in Figure 2b, demonstrate clear advantages in reasonably segmenting and recognizing small-interval and complex-shaped obstacles, making it suitable for outdoor unstructured spaces.

3.2. UKF obstacle trajectory prediction

The Kalman filter (KF) is widely used for system state estimation and prediction but works best with Linear Gaussian (LG) systems. The extended Kalman filter (EKF) is useful when dealing with non-linear mappings but can increase system error and complexity. The unscented Kalman filter (UKF), used for state estimation in non-linear LiDAR observing systems, outperforms in handling non-linear issues, offering superior probability distribution approximation.

The measurement model of LiDAR in the Cartesian coordinate system can be nonlinear, with a system state vector of $X = [x, y, z]^T$ representing the position of the measurement target in Cartesian coordinates. The measurement vector, denoted as $Z = [r, \theta, \varphi]^T$, consists of three components: the distance r, horizontal angle θ, and vertical angle φ, all of which are measured by the LiDAR. The measurement model can be expressed as follows:

$$Z = \begin{bmatrix} r \\ \theta \\ \varphi \end{bmatrix} = H(X) = \begin{bmatrix} \sqrt{x^2 + y^2 + z^2} \\ \arctan\left(\frac{y}{x}\right) \\ \arcsin\left(\frac{z}{\sqrt{x^2 + y^2 + z^2}}\right) \end{bmatrix} \tag{8}$$

let $H(X)$ be the nonlinear mapping from the state vector X to the observation vector Z.

The nonlinear system described above can be solved by UKF using the unscented transform (UT), with the mean and variance being approximated through sampling and weight assignment according to a specific regularity. The procedure for applying UT to this problem is described as follows [20]:

https://doi.org/10.1017/S026357472400122X Published online by Cambridge University Press

In the discrete-time domain, assuming that the prediction model is a variable acceleration period motion model $X_{k+1|k} = F(X_{k|k})$ and the LiDAR nonlinear observation model is $Z_{k+1|k} = H(X_{k+1|k})$, and knowing that the state mean value at moment k is $\overline{X}_{k|k}$, the observation value is z_k, and the variance is $P_{k|k}$, the statistical properties of the system can be obtained by UT constructing the $2n+1$*Sigma*-points $\chi^i_{k+1|k}$, and at the same time, constructing the corresponding weights $W^i_{k+1|k}$, which are $\chi^i_{k+1|k}$'s corresponding weights [21]:

$$\chi^0_{k|k} = \overline{X}_{k|k} \tag{9a}$$

$$\chi^i_{k|k} = \overline{X}_{k|k} + \left(\sqrt{(n+\lambda)\,P_{k|k}}\right)_i \; i = 1, \ldots, n \tag{9b}$$

$$\chi^i_{k|k} = \overline{X}_{k|k} - \left(\sqrt{(n+\lambda)\,P_{k|k}}\right)_{i-n} \; i = n+1, \ldots, 2n \tag{9c}$$

$$W^0_m = \frac{\lambda}{n+\lambda} \tag{9d}$$

$$W^0_c = W^0_m + \left(1 - \alpha^2 + \beta\right) \tag{9e}$$

$$W^i_m = W^i_c = \frac{1}{2\,(n+\lambda)} i = 1, \ldots, 2n \tag{9f}$$

where $\lambda = \alpha^2(n+\kappa) - n$ is the scale factor. $(P^{1/2})^T(P^{1/2}) = P$, $(P^{1/2})_i$ denote the ith column of the square root of the matrix.

The distribution state of the *Sigma*-points around $\overline{X}$ is determined by α. Adjusting α reduces the effect of higher order terms on the system, and is usually set to a small positive number $0 \le \alpha \le 1$, here $\alpha = 0.001$. There is no specific restriction on the value of κ, but at least ensure that the matrix $(n+\lambda)P$ is a semipositive definite matrix, and is usually set to $\kappa = 3 - n$, and κ is a non-negative number, here $\kappa = 0$.The state distribution parameter $\beta \ge 0$, by setting β, the accuracy of the variance can be improved, and for Gaussian distribution, $\beta = 2$ is optimal [22].

For the system prediction model, the predicted state value $X_{k+1|k}$ and variance $P_{k+1|k}$ can be obtained through the UT prediction process:

$$X_{k+1|k} = \sum_{i=0}^{2n} W^i_m \chi^i_{k+1|k},\; \chi^i_{k+1|k} = F\left(\chi^i_{k|k}\right) \quad i = 0, \ldots, 2n \tag{10a}$$

$$P_{k+1|k} = \sum_{i=0}^{2n} W^i_c \left(\chi^i_{k+1|k} - X_{k+1|k}\right)\left(\chi^i_{k+1|k} - X_{k+1|k}\right)^T + Q_k \tag{10b}$$

The implementation of the HDBSCAN-UKF obstacle trajectory prediction algorithm is shown in Alg. 1. Assuming the sampling time Δts, the obstacle trajectory prediction for $\Delta t \cdot Ns$ can be realized by Alg. 1, which leads to the set of future states $O^p_{obs} = \{O^p_1, O^p_2, \ldots, O^p_n\}$ of n clusters of obstacles.

For example, a variable acceleration periodic motion of an obstacle position over time satisfies:

$$\begin{cases} x = x_0 - A\sin(2\pi\omega t) \\ y = y_0 - A\cos(2\pi\omega t) \end{cases} \tag{11}$$

Given the initial obstacle state $(x_0, y_0, r_0) = (8.5, 2.0, 0.48)$, amplitude $A = 2.5m$, frequency $\omega = 0.015$, sampling time $\Delta t = 0.5\,s$, and prediction step size $N = 30$, the algorithm can predict the obstacle's motion 3 s ahead. Figure 3a and Figure 3b depicts the actual path of the obstacle along the x- and y-axes, as well as the tracking and predicted trajectories generated by the HDBSCAN and UKF algorithms, respectively.

In addition, the root-mean-square error (RMSE, Eq. 12) can be used to judge the effectiveness of the trajectory tracking of the HDBSCAN algorithm, as well as the effectiveness of the trajectory prediction of the UKF algorithm. The calculation results are shown in Table I.

https://doi.org/10.1017/S026357472400122X Published online by Cambridge University Press

Algorithm 1: HDBSCAN-UKF Trajectory Prediction.

Data: Prediction step size: N, Lidar point clouds: $C = \{X, Y, Z\}$
Result: State prediction : $X_{K+N|k+N}$, Variance prediction: $P_{k+N|k+N}$
while $C \neq \emptyset$ **do**
  $(X_{k|k}, P_{k|k}) \leftarrow HDBSCAN(C)$;
  for $i \leftarrow 1$ **to** N **do**
    $(X_{k+i|k+i-1}, P_{k+i|k+i-1}) \leftarrow UT_{prediction}(X_{k+i-1|k+i-1}, P_{k+i-1|k+i-1})$;
    $(X_{k+i|k+i}, P_{k+i|k+i}) \leftarrow UT_{correction}(X_{k+i|k+i-1}, P_{k+i|k+i-1})$;
  end
end

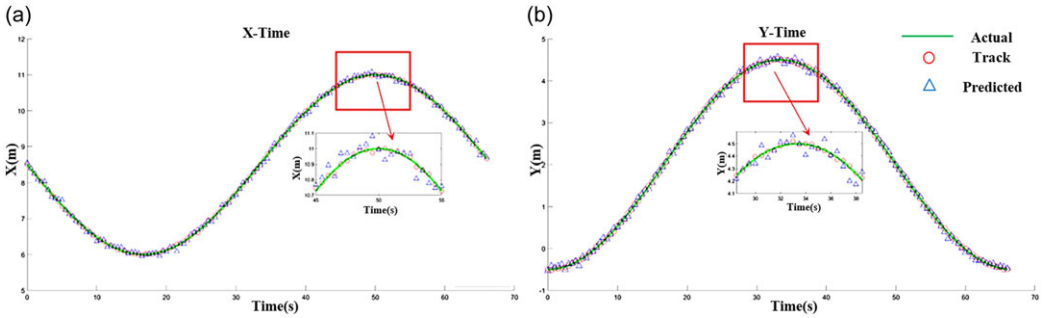


Figure 3. *(a) and (b) Show the actual, tracked, and predicted obstacle trajectories in the x and y directions.*

Where x_i, x_i^{tra}, and x_i^{pre} are the actual trajectory, tracking trajectory, and predicted trajectory of the obstacle in the x-axis direction at the ith step, respectively, and $n$is the total number of samples, and the same for the y-axis.

$$\begin{cases} RMSE_x^{tra} = \sqrt{\sum_{i=0}^{n-1} (x_i - x_i^{tra}) / n} \\ RMSE_x^{pre} = \sqrt{\sum_{i=0}^{n-1} (x_i - x_i^{pre}) / n} \end{cases} \tag{12}$$

3.3. AD-CBF-MPC control strategy

Section 2 explains the basic principle and implementation process of the MPC-CBF algorithm. While the algorithm is effective in avoiding static obstacles, it struggles to avoid dynamic obstacles that change their trajectories frequently.

In order to make up for the shortcomings of this algorithm, this section firstly proposes to fuse the current state set O_{obs}^c of the obstacle obtained in Section 3.1, with the future state set O_{obs}^p of the obstacle obtained in Section 3.2, and the resulting $O_{obs} = \{O_{obs}^c, O_{obs}^p\}$ is used as the avoidance target of the MPC-CBF algorithm to improve the dynamic performance of the algorithm, and to realize the dynamic control barrier function (D-CBF). Where each element of O_{obs} is of the form $O_i = [x_{obs}^i, y_{obs}^i, r_{obs}^i](i = 1, \ldots, 2n)$, the current and future states of the obstacle at discrete time k can be represented by $O_i(k) = [x_{obs}^i(k), y_{obs}^i(k), r_{obs}^i(k)]$ and the center position of the UGV is represented by $p(k) = [x(k), y(k)]$.

https://doi.org/10.1017/S026357472400122X Published online by Cambridge University Press

Table I. *Error analysis table for HDBSCAN-UKF prediction and tracking.*

	x_{tra}	y_{tra}	x_{pre}	y_{pre}
RMSE	0.015	0.014	0.063	0.051

Then the functional form of D-CBF can be chosen as:

$$h\left(X_k^i\right) = \left(x\left(k\right) - x_{obs}^i\left(k\right)\right)^2 + \left(y\left(k\right) - y_{obs}^i\left(k\right)\right)^2 - \left(r_c + r_{obs}^i\left(k\right) + r_s\right)^2 \tag{13a}$$

$$\Delta h\left(X_k^i, u_k\right) \geq -\gamma\left(h\left(X_k^i\right)\right), k = 0, \ldots, M_{CBF} - 1; \quad i = 1, \ldots, 2n \tag{13b}$$

where the admissible state is $X_k^i = \{p(k), O_i(k)\}$, u_k is the optimal input to the MPC solution, r_c and r_s are the UGV body radius and safety distance, respectively, M_{CBF} is the CBF constraint step, and n is the number of obstacles.

The flexible selection of $\gamma(h(\cdot))$ is crucial for the CBF planning process. It usually takes a smaller value ($0 < \gamma \leq 1$), indicating the control performance trade-off for system stability. Higher γ values increase stability but may decrease control performance [23]. Therefore, it is important to choose the value of γ in a rational and dynamic manner to balance stability and control performance under different obstacles.

When considering the actual situation of UGV obstacle avoidance, the selection of γ is influenced by two aspects:

- The distance, $d^i(k)$, between the position of the UGV at moment k and the center position of the ith obstacle. As $d^i(k)$ decreases, $\gamma(h(\cdot))$ also decreases and this relationship is positively correlated.
- The size of the radius of the obstacle, $r_{obs}^i(k)$. As $r_{obs}^i(k)$ becomes smaller, $\gamma(h(\cdot))$ increases, and this relationship is negatively correlated.

According to Eq. (13a), $h(X_k^i)$ already includes both $d^i(k)$ and $r_{obs}^i(k)$ attributes. The final adaptive functional form of $\gamma(h(\cdot))$ can be written as:

$$\gamma\left(h\left(X_k^i\right)\right) = A\left(1 + \frac{erf\left(d^i\left(k\right) - d_{\mathrm{E}}\right)}{\sqrt{2}\sigma_d}\right)\left(1 - \frac{erf\left(r_{obs}^i\left(k\right) - r_{\mathrm{E}}\right)}{\sqrt{2}\sigma_r}\right)$$

$$k = 0, \ldots, M_{CBF} - 1; \quad i = 1, \ldots, 2n \tag{14}$$

The equation defines A as the amplitude, with d_{E} and r_{E} representing the expected means of d and r, respectively. σ_d and σ_r indicate the standard deviation of d and r, respectively, and $erf(\cdot)$ is the error function. Given the environmental conditions, obstacle features, and dimensions of UGVs, the parameter $[A, d_{\mathrm{E}}, r_{\mathrm{E}}, \sigma_d, \sigma_r] = [0.5, 2.0, 0.4, 0.6, 0.2]$ is chosen.

The UGV was kept stationary, and obstacle 1 with a radius of 0.4 m was set to approach from 4.0 m away from the UGV, and obstacle 2 with a radius of 0.6 m was set to move away from 1.0 m away from the UGV. Figure 4 depicts the $\gamma(h)$ value distribution for this case. Figure 4a shows the total $\gamma(h)$ distribution for all revealed obstacle motion features, and Figure 4b shows the distribution of $\gamma(h)$ in the 50 results solved for two obstacles with different states from the beginning of AD-CBF planning. Eq. (14) allows $\gamma(h)$ to autonomously select a suitable value based on the obstacle and UGV's relative state, ensuring a balance between system safety and feasibility.

Ultimately, the associative formulations (2), (13), and (14), such that $M_{CBF} = N$, can be obtained as the control strategy of AD-CBF-MPC:

$$J_t^*\left(x_t\right) = \min_{u_{t:\, t+N-1|t}} p\left(x_{t+N|t}\right) + \sum_{k=0}^{N-1} q\left(x_{t+k|t}, u_{t+k|t}\right) \tag{15a}$$

https://doi.org/10.1017/S026357472400122X Published online by Cambridge University Press

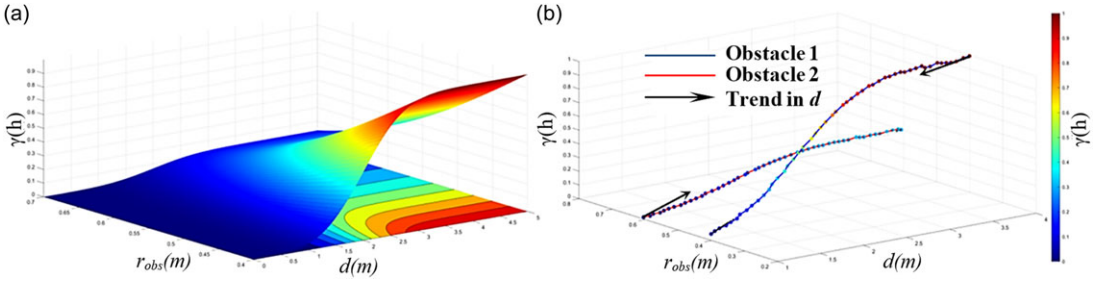


Figure 4. *(a) Is a plot of the total distribution of $\gamma(h)$ values for the test case, and (b) Shows the distribution of $\gamma(h)$ values corresponding to the 50 steps of obstacle states backward from the start of AD-CBF planning.*

$$s.t.\ x_{t+k+1|t} = f\left(x_{t+k|t}, u_{t+k|t}\right), k = 0, \ldots, N-1 \tag{15b}$$

$$x_{t+k|t} \in X, u_{t+k|t} \in U, k = 0, \ldots, N-1 \tag{15c}$$

$$x_{t|t} = x_t, x_{t+N|t} \in X_f \tag{15d}$$

$$X^i_{t+k|t} = \left\{x_{t+k|t}, O_i\ (k)\right\}, k = 0, \ldots, N-1; \quad i = 1, \ldots, 2n \tag{15e}$$

$$h\left(X^i_{t+k|t}\right) = \left(x\ (k) - x^i_{obs}\ (k)\right)^2 + \left(y\ (k) - y^i_{obs}\ (k)\right)^2 - \left(r_c + r^i_{obs}\ (k) + r_s\right)^2 \tag{15f}$$

$$\gamma\left(h\left(X^i_{t+k|t}\right)\right) = A\left(1 + \frac{erf\left(d^i\ (t+k) - d_{\mathrm{E}}\right)}{\sqrt{2}\sigma_d}\right)\left(1 - \frac{erf\left(r^i_{obs}\ (t+k) - r_{\mathrm{E}}\right)}{\sqrt{2}\sigma_r}\right) \tag{15g}$$

$$\Delta h\left(X^i_{t+k|t}, u_{t+k|t}\right) \geq -\gamma\left(h\left(X^i_{t+k|t}\right)\right), k = 0, \ldots, N-1; \quad i = 1, \ldots, 2n \tag{15h}$$

4. Experiments

In the experiments, several high-performance methods are combined to create a safe and reliable UGV navigation framework (Figure 5) for collision-free navigation. Through the data fusion between LiDAR, IMU, and wheel odometer, the environment is 3D mapped and saved as a Point Cloud Data (PCD) using Fast-lio-lc [24, 25], and handed over to the Removert [26] algorithm for the removal of dynamic trajectories, and finally a static PCD point cloud is obtained; subsequently, the Amcl-3 dl [27] algorithm performs the position by calling the point cloud map data and comparing it with the current surrounding point cloud of the UGV for position estimation; after that, the estimated position is used as the reference position of the UGV for the PF-RRT* algorithm [28–30], which is given as the target point of the UGV on the point cloud a priori map, and global path planning is carried out to obtain the reference path of the UGV to reach the target point.

In addition, the improved HDBSCAN-UKF algorithm [19, 22] is used to parameterize the obstacles around the unmanned vehicle and predict its future trajectory, which is passed to the AD-CBF-MPC [6, 12] local planning algorithm for obstacle avoidance, while the MPC algorithm obtains the reference path of global path planning in real time for following until it finally arrives at the target point safely.

4.1. Simulation scenarios

In order to evaluate the performance of the proposed method, a simulation environment with complex terrain and dynamic obstacles is built in Gazebo as shown in Figure 6a. The blue solid arrows show

https://doi.org/10.1017/S026357472400122X Published online by Cambridge University Press

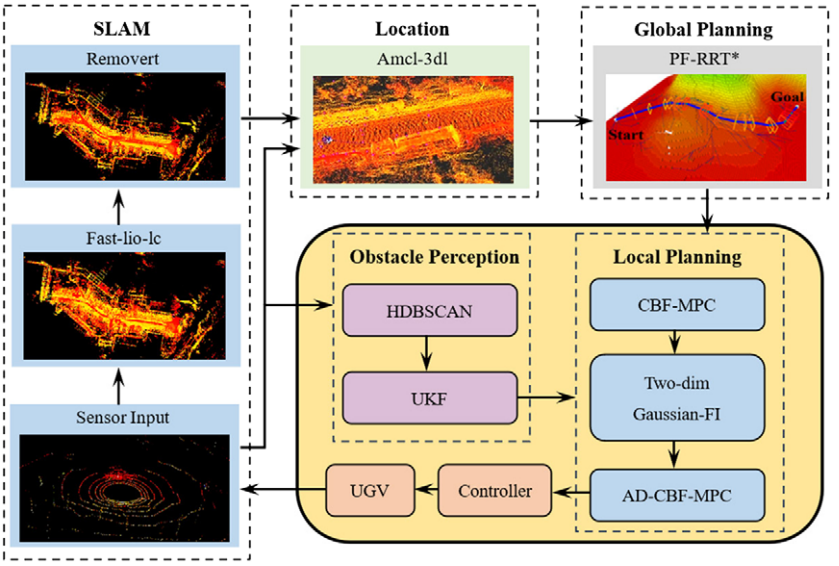


Figure 5. *Framework for UGV navigation systems.*

the global reference path (35.869 m), red dashed arrows indicate obstacle movement, and yellow dashed boxes mark static obstacles. The terrain is a bumpy mud slab. A simulated UGV, same size as the actual vehicle, is used. The AD-CBF-MPC obstacle avoidance navigation algorithm is compared with four other benchmark methods in this environment:

- MPC-DC: Model predictive controllers with Euclidean safety distance constraints.
- MPC-CBF: Model predictive controller with safety prioritized control barrier function, where $\gamma = 0.1$.
- D-CBF-MPC: Safety-first obstacle avoidance model predictive controller with HDBSCAN-UKF obstacle trajectory prediction, where $\gamma = 0.1$.
- A-CBF-MPC: Model predictive controller with A-CBF.

The performance of five algorithms was compared using the following metrics: Nav-time: Time for UGV to travel from start to end. Min-dist: Minimum distance between UGV and obstacle during navigation. Ref-time: Time from obstacle detection to initiation of avoidance behavior. Vel-var: Velocity variance when avoiding obstacles. Path-len: Total distance traveled by the UGV from start to end.

These performance metrics are presented in Table II.

Each algorithm's driving trajectories are shown in different colors in Figure 6b.

- MPC-DC (red) does not have advance avoidance constraints or obstacle prediction and therefore is not able to perform safe avoidance behaviors for dynamic obstacles and collides with the first dynamic obstacle.
- MPC-CBF (blue) does not consider the influence of future trajectories of obstacles and CBF is only point-by-point feasible, which leads to the algorithm's difficulty in avoiding dynamic obstacles, small safety distances, large speed variations, and steep driving trajectories.
- D-CBF-MPC (green) takes into account the future trajectory of the obstacle and is able to avoid dynamic obstacles safely and in advance, and has the shortest navigation time compared to the other four algorithms, but does not have the global persistent feasibility of the CBF, which leads

https://doi.org/10.1017/S026357472400122X Published online by Cambridge University Press

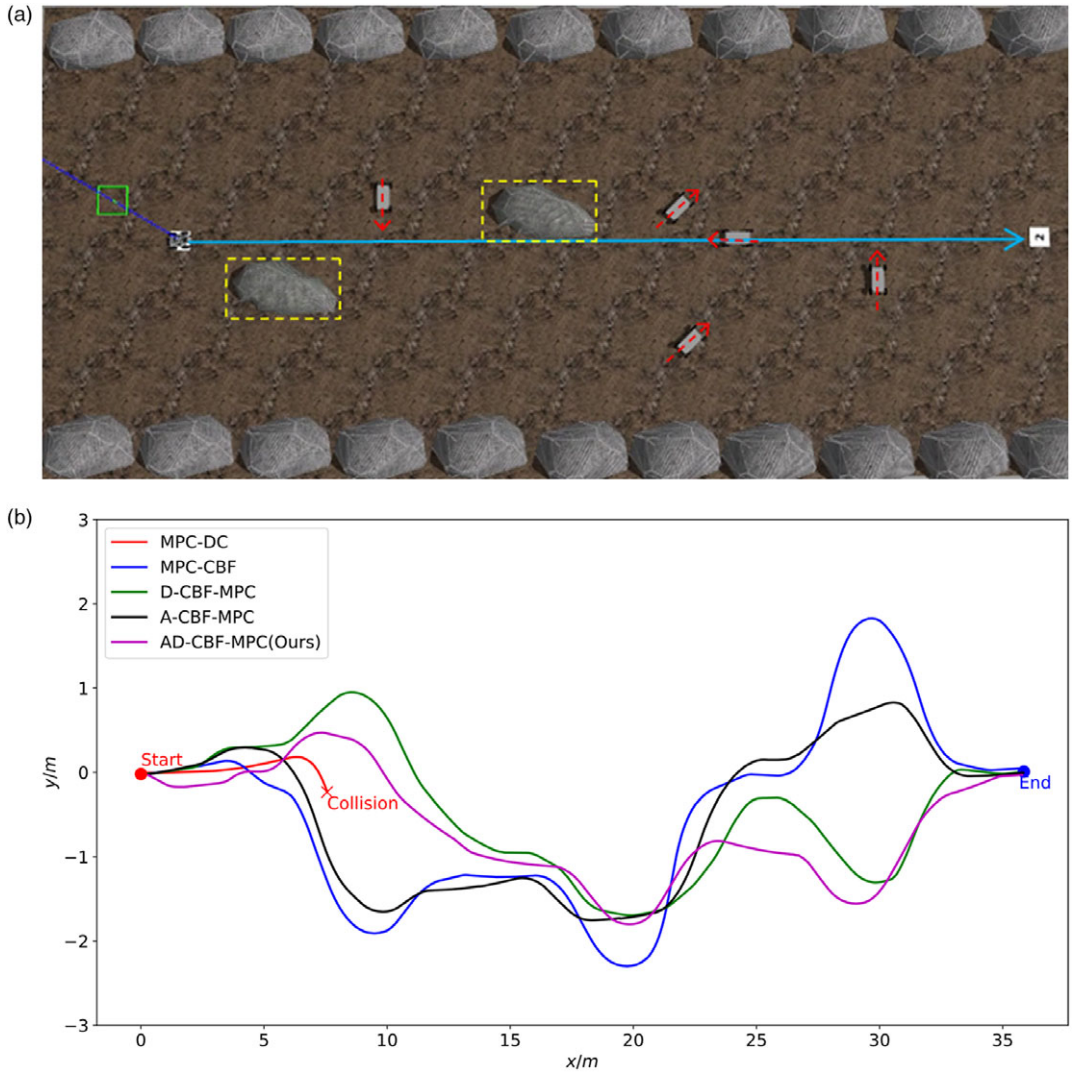


Figure 6. *(a) Shows the Gazebo dynamic obstacle avoidance simulation environment. (b) Shows the trajectory comparison of multiple obstacle avoidance algorithms in the simulation environment.*

to the planning of the obstacle avoidance paths is not optimal, and the final length of the traveled path is on the large side.

- A-CBF-MPC (black) is the opposite case of the D-CBF-MPC method, where the availability of the global persistent feasibility of CBF leads to an improvement in navigation time and velocity change for obstacle avoidance (with respect to MPC-CBF), but the absence of future position constraints on the trajectory of the dynamic obstacle leads to algorithmic results in terms of minimum safe distances, velocity changes, and path lengths that do not reach the ideal case.
- The method proposed in this paper, AD-CBF-MPC (purple), possesses both the global persistent feasibility of CBF and the future position constraints of dynamic obstacle trajectories, which makes our method reach the optimal case compared with other methods, except for the navigation time due to the adaptive solution, which is slightly higher than that of the D-CBF-MPC method, which makes the unmanned vehicle navigate the process kind of the unmanned vehicle, which

https://doi.org/10.1017/S026357472400122X Published online by Cambridge University Press

Table II. *Comparison of AD-CBF-MPC with other baseline methods.*

Algorithms	Nav-time(s)	Min-dist(m)	Ref-time(s)	Vel-var	Path-len(m)
MPC-DC	–	0.00	0.938	–	–
MPC-CBF	49.90	0.53	0.370	0.1257	38.525
D-CBF-MPC	48.41	0.83	0.357	0.0172	37.155
A-CBF-MPC	49.23	0.57	0.433	0.0331	37.131
Ours	47.23	0.94	0.345	0.0101	36.728

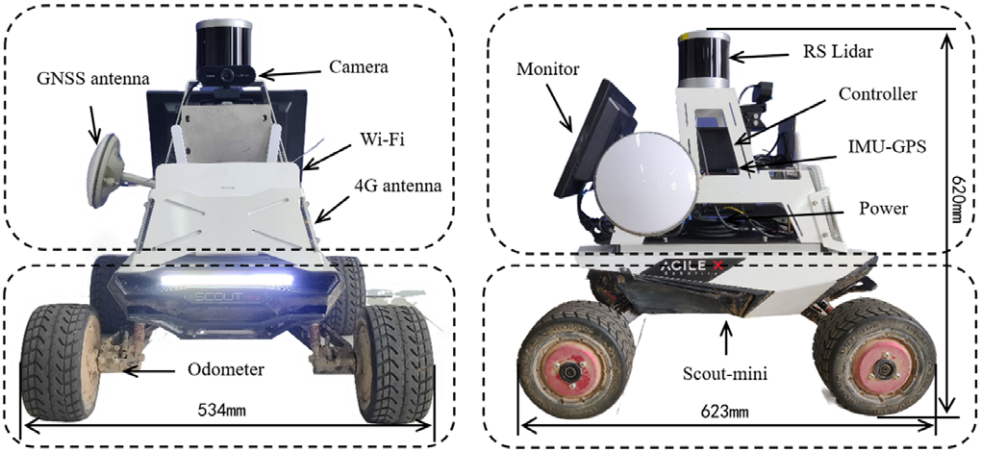


Figure 7. *Schematic of UGV platform hardware.*

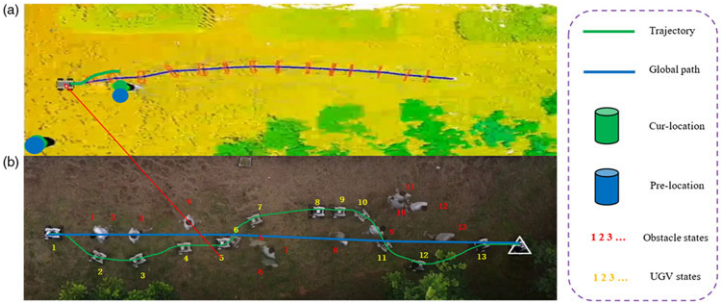


Figure 8. *Dynamic obstacle avoidance experiments in real environments.*

can safely and quickly arrive at the target point, and keep a safer and minimum distance with dynamic obstacles on the way, and also ensure the unmanned vehicle's speed to smoothly change, and generate the traveling trajectory with the shortest distance.

4.2. Real-world scenarios

Real scenario experiments are performed on the UGV platform depicted in Figure 7. The algorithms will operate under the ROS Noetic operating system. The LiDAR's scanning frequency is 10 Hz with a planning frequency of 30 Hz. Additionally, an industrial computer with 8-core Arm 64-bit 2.2 GHz CPU, 32 GB video memory, and 64 GB storage is employed as the algorithm solver and simulation operating system.

https://doi.org/10.1017/S026357472400122X Published online by Cambridge University Press

A forest with uneven terrain was used for real-world navigation and obstacle avoidance testing. Figure 8 shows pedestrians randomly obstructing the UGV's path. The UGV's local perception range is set to $8 \times 8\,m$ with a 0.1m map resolution. HDBSCAN-UKF and AD-CBF-MPC both have a predicted step size of 30. The UGV has a body radius of $r_c = 0.3\,m$ and a safety distance of $r_d = 0.2\,m$.

Figure 8 illustrates the real-time obstacle avoidance process. The AD-CBF-MPC algorithm produces a local obstacle avoidance path (green line), and the PF-RRT* algorithm generates a global reference trajectory (blue line). Green and blue cylinders represent the current and predicted obstacle positions, respectively. Red and yellow numbers correspond to the obstacle and UGV states at discrete moments.

Figure 8a displays the algorithm's real-time solutions at discrete moments, while Figure 8b shows state changes during navigation. The algorithm effectively avoids obstacles in complex outdoor environments. Upon detecting an obstacle, its state is parameterized by HDBSCAN (green column) and its future state is predicted by UKF (blue column). These states form the AD-CBF-MPC's obstacle avoidance constraints, generating a safe, smooth local path.

As shown in Figure 8b, the UGV avoids obstacles by moving in the opposite direction, with the trajectory prediction and A-CBF constraints greatly reducing the cost of optimal solution, enhancing safety and providing smoother, faster trajectories.

5. Conclusion and future work

The CBF-MPC and D-CBF-MPC algorithms are enhanced to handle complex terrain navigation and dynamic obstacle avoidance. Obstacle states are parameterized by HDBSCAN and their trajectories are predicted using the UKF algorithm for real-time tracking and position prediction. These positions are integrated into CBF constraints to form the D-CBF. An adaptive version, AD-CBF, is proposed to solve the point-by-point feasibility issue in classical CBF, ensuring global persistent feasibility.

The proposed algorithm's effectiveness is demonstrated in both simulated and real environments. It outperforms others in all performance metrics, especially compared to the D-CBF-MPC algorithm. The improved AD-CBF-MPC algorithm increases the minimum safe distance by 13% and reduces speed variations by 41%, enhancing safety, speed, and smoothness of UGV navigation.

Because AD-CBF algorithm needs to adaptively calculate the control performance parameter γ of CBF before obstacle avoidance planning, this method has limitations in real-time performance and is only suitable for low-speed motion models. In the future, it is planned to develop more reasonable, effective, and robust obstacle avoidance strategies by further improving the adaptive functional form of CBF and taking into account end-to-end methods such as deep learning.

Author contributions. All authors contributed to the study's conception and design. Material preparation, data collection, and analysis were performed by Liang Guo and Suyu Zhang. The first draft of the manuscript was written by Liang Guo and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

Competing interests. The authors declare no competing interests exist.

Ethical approval. None.

References

[1] I. S. Mohamed, K. Yin and L. Liu, "Autonomous navigation of AGVs in unknown cluttered environments: Log-MPPI control strategy," *IEEE Robot Autom Lett* **7**(4), 10240–10247 (2022).

[2] R. MacLachlan and C. Mertz, "Tracking of Moving Objects from a Moving Vehicle Using a Scanning Laser Rangefinder," **In:** *IEEE Intelligent Transportation Systems Conference*, Edmonton, Canada (IEEE, 2006) pp. 301–306.

[3] A. Vatavu, M. Rahm, S. Govindachar, G. Krehl, A. Mantha, S. R.Bhavsar, M. R. Schier, J. Peukert and M. Maile, "From particles to self-localizing tracklets: A multilayer particle filter-based estimation for dynamic grid maps," *IEEE Intel Transp Syst Mag* **12**(4), 149–168 (2020).

[4] J. Jung and S. Bae, "Real-time road lane detection in urban areas using liDAR data," *Electronics* **7**(11), 276 (2018).

https://doi.org/10.1017/S026357472400122X Published online by Cambridge University Press

[5] B. Brito, B. Floor, L. Ferranti and J. Alonso-Mora, "Model predictive contouring control for collision avoidance in unstructured dynamic environments," *IEEE Robot Autom Lett* **4**(4), 4459–4466 (2019).
[6] Z. Jian, Z. Yan, Lei X., Z. Lu, B. Lan, X. Wang and B. Liang, "Dynamic Control Barrier Function-based Model Predictive Control to Safety-Critical Obstacle-Avoidance of Mobile Robot," **In:** *IEEE International Conference on Robotics and Automation (ICRA)*, London, United Kingdom (IEEE, 2023) pp. 3679–3685.
[7] S. He, S.-L. Dai, Z. Zhao, T. Zou and Y. Ma, "UDE-based distributed formation control for MSVs with collision avoidance and connectivity preservation," *IEEE Trans Ind Inform* **20**(2), 1476–1487 (2024).
[8] F. Hogan and A. Rodriguez, "Reactive planar non-prehensile manipulation with hybrid model predictive control," *Int J Robot Res* **39**(7), 755–773 (2020).
[9] S. S. Ge and Y. J. Cui, "Dynamic motion planning for mobile robots using potential field method," *Auton Robot* **13**(3), 207–222 (2002).
[10] D. H. Lee, S. S. Lee, C. K. Ahn, P. Shi and C.-C. Lim, "Finite distribution estimation-based dynamic window approach to reliable obstacle avoidance of mobile robot," *IEEE Trans Ind Electron* **68**(10), 9998–10006 (2021).
[11] N. Scianca, D. De Simone, L. Lanari and G. Oriolo, "MPC for humanoid gait generation: Stability and feasibility," *IEEE Trans Robot* **36**(4), 1171–1188 (2020).
[12] J. Zeng, B. Zhang and K. Sreenath, "Safety-Critical Model Predictive Control with Discrete-Time Control Barrier Function," **In:** *American Control Conference (ACC)*, New Orleans, Louisiana (IEEE, 2021) pp. 3882–3889.
[13] H. Yu, J. Duan, S. Taheri, H. Cheng and Z. Qi, "A model predictive control approach combined unscented Kalman filter vehicle state estimation in intelligent vehicle trajectory tracking," *Adv Mech Eng* **7**(5), 1–14 (2015).
[14] U. Rosolia, S. De Bruyne and A. G. Alleyne, "Autonomous vehicle control: A nonconvex approach for obstacle avoidance," *IEEE Trans Contr Syst Technol* **25**(2), 469–484 (2017).
[15] X. Zhang, A. Liniger and F. Borrelli, "Optimization-based collision avoidance, "*IEEE Trans Contr Syst Tech* **29**(3), 972–983 (2021).
[16] T. D. Son and Q. Nguyen, "Safety-Critical Control for Non-affine Nonlinear Systems with Application on Autonomous Vehicle," **In:** *IEEE 58th Conference on Decision and Control (CDC)*, Nice, France (IEEE, 2019) pp. 7623–7628.
[17] U. Rosolia and A. D. Ames, "Multi-rate control design leveraging control barrier functions and model predictive control policies," *IEEE Control Syst Lett* **5**(3), 1007–1012 (2021).
[18] M. Dhanushree, R. Priya, P. Aruna and R. Bhavani, "A Keyframe Extraction Using HDBSCAN With Particle Swarm Optimization," **In:** *10th International Conference on Signal Processing and Integrated Networks (SPIN)*, Noida, India (IEEE, 2023) pp. 2445–450.
[19] A. C. A. Neto, J. Sander, R. J. G. B. Campello and M. A. Nascimento, "Efficient computation and visualization of multiple density-based clustering hierarchies," *IEEE Trans Knowl Data Eng* **34**(8), 3075–3089 (2021).
[20] E. A. Wan and R. Van Der Merwe, "The Unscented Kalman Filter for Nonlinear Estimation," **In:** *Proceedings of the IEEE. 2000 Adaptive Systems for Signal Processing, Communications, and Control Symposium*, Alberta, Canada (IEEE, 2000) pp. 153–158.
[21] H. Yu, J. Duan, S. Taheri, H. Cheng and Z. Qi, "A model predictive control approach combined unscented Kalman filter vehicle state estimation in intelligent vehicle trajectory tracking," *Adv Mech Eng* **7**(5), 655–663 (2015).
[22] B. Gao, S. Gao, Y. Zhong, G. Hu and C. Gu, "Interacting multiple model estimation-based adaptive robust unscented Kalman filter," *Int J Control Autom Syst* **15**(5), 2013–2025 (2017).
[23] J. Zeng, Z. Li and K. Sreenath, "Enhancing Feasibility and Safety of Nonlinear Model Predictive Control with Discrete-Time Control Barrier Functions," **In:** *60th IEEE Conference on Decision and Control (CDC)*, Texas, USA (IEEE, 2021) pp. 6137–6144.
[24] W. Xu and F. Zhang, "FAST-LIO: A fast, robust LiDAR-inertial odometry package by tightly-coupled iterated Kalman filter," *IEEE Robot Autom Lett* **6**(2), 3317–3324 (2021).
[25] W. Xu, Y. Cai, D. He, J. Lin and F. Zhang, "FAST-LIO2: Fast direct LiDAR-inertial odometry," *IEEE Trans Robot* **38**(4), 2053–2073 (2022).
[26] G. Kim and A. Kim, "Remove, then Revert: Static Point Cloud Map Construction using Multiresolution Range Images," **In:** *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Las Vegas, USA (IEEE, 2020) pp. 10758–10765.
[27] Y. Zhao, C. Fritsche, F. Yin, F. Gunnarsson and F. Gustafsson, "Sequential monte carlo methods and theoretical bounds for proximity report based indoor positioning," *IEEE Trans Veh Technol* **67**(6), 5372–5386 (2018).
[28] P. Fankhauser, M. Bloesch and M. Hutter, "Probabilistic terrain mapping for mobile robots with uncertain localization," *IEEE Robot Autom Lett* **3**(4), 3019–3026 (2018).
[29] J. D. Gammell, S. S. Srinivasa and T. D. Barfoot, "Informed RRT*: Optimal Sampling-Based Path Planning Focused via Direct Sampling of an Admissible Ellipsoidal Heuristic," **In:** *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Illinois, USA (IEEE, 2014) pp. 2997–3004.
[30] Z. Jian, Z. Lu, X. Zhou, B. Lan, A. Xiao, X. Wang and B. Liang, "PUTN: A Plane-fitting based Uneven Terrain Navigation Framework," **In:** *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Kyoto, Japan (IEEE, 2022) pp. 7160–7166.

Cite this article: L. Guo, S. Zhang, W. Zhao, J. Liu and R. Liu (2024). "Obstacle avoidance control of UGV based on adaptive-dynamic control barrier function in unstructured terrain", Robotica **42**, 2991–3004. https://doi.org/10.1017/S026357472400122X

https://doi.org/10.1017/S026357472400122X Published online by Cambridge University Press