

# **DBSCAN-Based Minimum Enclosing Ellipse Using the Control Barrier Function for Safe Navigation of Mobile Robots**

Ju-Feng Wu, Chia-Chun Huang, Ming-Yang Cheng\*

Department of Electrical Engineering, National Cheng Kung University, Tainan, Taiwan, ROC

Received 03 December 2024; received in revised form 28 February 2025; accepted 11 March 2025

DOI: <https://doi.org/10.46604/aiti.2025.14579>

## **Abstract**

This paper aims to reduce the redundant unsafe area in quadratic program approaches based on the Control Barrier Function (CBF) and the Control Lyapunov Function (CLF) for collision avoidance, hereafter referred to as the CBF-CLF approach. Existing CBF-CLF quadratic program approaches typically construct CBF based on Euclidean distance; however, the redundant unsafe area due to obstacles is excessively large, which may prevent finding feasible solutions. To address this issue, this study employs density-based spatial clustering of applications with noise (DBSCAN) and the Minimum Enclosing Ellipse (MEE) to reduce the unsafe area. The proposed approach is referred to as the DBSCAN-MEE-CBF. The effectiveness of the proposed method is demonstrated through both computer simulations and real-world experiments. Specifically, the proposed method reduces the size of the redundant unsafe area by up to 26.52% while maintaining robust collision avoidance.

**Keywords:** collision avoidance, mobile robot navigation, Control Barrier Function, Control Lyapunov Function

## **1. Introduction**

Various robotic systems are commonly used in automated factories [1]. In addition to industrial robot manipulators that are typically fixed to a specific site, other robotic systems, such as automated guided vehicles (AGV) and autonomous mobile robots (AMR), are also deployed in factories to improve production efficiency. In particular, AGVs move along a pre-defined path. The AGV will stop moving when it detects obstacles that may cause a possible collision. However, the number of AGVs used in modern automated factories or logistics centers is very large. Since the predefined paths for AGVs to move along may intersect, collision events will likely occur and lead to a deterioration in production efficiency. To cope with this problem, developing a mobile robot that does not need to move along a predefined path and can avoid obstacles autonomously is crucial. In particular, commonly used collision avoidance algorithms for AMRs include the potential field method [2], vector field histogram (VFH) [3], and Rapidly-Exploring Random Tree (RRT) and its improved variant, RRT\*, etc. These aforementioned methods cannot guarantee the safety of AMRs or AGVs.

To cope with this problem, the ideas of Control Barrier Function (CBF) and Control Lyapunov Function (CLF) are adopted in this paper to avoid collisions with obstacles. Moreover, this paper uses density-based spatial clustering of applications with noise (DBSCAN) in combination with a Minimum Enclosing Ellipse (MEE) to define the unsafe regions around obstacles. Compared to the enclosing circle, the MEE provides more space for the robot. Both computer simulations and actual experiments are conducted to verify the effectiveness of the proposed DBSCAN-MEE-CBF approach. In particular, a TurtleBot running on Robot Operating System (ROS) is used to perform navigation while avoiding obstacles. Both the simulation and experimental results indicate that the proposed DBSCAN-MEE-CBF approach can reduce the size of the redundant unsafe area of obstacles from 1.602% to 26.52% and accomplish collision avoidance.

---

\* Corresponding author. E-mail address: mycheng@mail.ncku.edu.tw

The remainder of this paper is organized as follows. Section 2 provides a brief review of previous works related to obstacle avoidance algorithms, CBF-CLF-based approaches, and DBSCAN clustering. Section 3 gives a brief introduction to the CBF, which serves as the foundation for the proposed method. Section 4 introduces the proposed DBSCAN and MEE method, explaining how it improves obstacle avoidance by reducing the size of the redundant unsafe areas. Section 5 presents simulation and experimental results to validate the effectiveness of the proposed approach. Finally, Section 6 concludes the paper and discusses potential future research directions.

## 2. Literature Review

In recent years, environmental modeling and risk-aware navigation have become crucial in public health surveillance [4] and mobile robot navigation [5]. In particular, the role of fluid dynamics in airborne disease transmission is investigated in Koley [4], demonstrating that precise environmental modeling is essential for optimizing ventilation strategies and infection risk assessment. Similarly, mobile robots operating in complex environments also require an accurate perception of obstacle distributions and risk areas to navigate safely and efficiently.

In the existing obstacle avoidance approaches, the potential field [6] approach can be applied to obstacle avoidance problems for different types of robotic systems, such as mobile robots [7-9] and industrial robot manipulators [10-12]. To increase computation speed and enhance the stability of error detection, the moving direction of a mobile robot is represented by a histogram in the VFH method [13]. The moving direction with a higher histogram has a higher cost and is thus less desirable. The VFH\* approach [14], an enhanced version of the VFH method, integrates the VFH method with the A-star pathfinding (A\*) algorithm to handle the dead-end problem. However, the common drawback of the approaches mentioned above is that the mobile robot may be trapped at a local minimum, so it may fail to reach its destination.

In contrast, the RRT approach [15] adopts a different strategy. In the RRT approach, the environment model is constructed in advance. The tree and nodes between the starting and destination points are also built. If there are obstacles between the starting point and the destination point, these obstacles can be avoided during the tree construction process. After completing the tree and nodes, the mobile robot can move along the planned path to prevent it from being trapped in a local minimum. However, one of the biggest problems for the RRT approach is that if an object accidentally appears on the planned path when the mobile robot is moving along the path, the mobile robot must stop to replan the path to avoid the obstacle. This may significantly slow down the moving task, and the production efficiency will be affected as well.

To remedy the aforementioned drawback, a motion planning approach that combines the CBF and CLF was proposed [16-17]. In particular, the CLF is employed as the control scheme to control the mobile robot moving from the starting point to the destination point. At the same time, the CBF is used as a safety filter to ensure the safety of the mobile robot when avoiding obstacles. Moreover, through some mathematical manipulations, CBF and CLF are unified as a quadratic program, referred to as the CBF-CLF quadratic program. Nevertheless, when using the CBF-CLF quadratic program to perform motion planning, it may not be possible to find feasible solutions since the redundant unsafe area due to obstacles is excessively large.

To tackle this problem, this paper proposes an approach that exploits the idea of DBSCAN clustering [18] to find the MEE for the CBF-CLF quadratic program, which is called the DBSCAN-MEE-CBF in this paper. The proposed DBSCAN-MEE-CBF approach can reduce the size of the redundant unsafe area of obstacles while ensuring that the mobile robot can successfully avoid obstacles. Compared with the original CBF-CLF quadratic program, the moving space of the mobile robot, as well as the chance of finding feasible solutions, becomes larger for the proposed DBSCAN-MEE-CBF approach. In addition, the trajectory planned by the proposed DBSCAN-MEE-CBF is shorter. The relationship and comparison of the existing literature are shown in Table 1.

Table 1 The relationship and comparison of the literature

Method	Key Concept	Advantages	Disadvantages	References
Potential field	Uses attractive and repulsive forces to navigate obstacles	Easy to implement; real-time responsiveness	May get stuck in a local minimum	[2, 6]
Vector field histogram (VFH)	A bar chart can represent the traveling cost in different directions. The higher the bar, the greater the cost of moving in that direction, indicating a lower likelihood of passage.	Efficient obstacle detection; fast computation	The performance is not good when the robot encounters a dead end and needs a sharp turn.	[13]
VFH*	Combines VFH with A* algorithm to address dead-end issues	Improved pathfinding in complex environments	Still susceptible to local minima	[14]
Rapidly-Exploring Random Tree (RRT)	Constructs a tree to find feasible paths from start to goal	Can handle complex environments	Requires re-planning if new obstacles appear	[15]
Traditional CBF-CLF quadratic program	Uses Control Barrier Function (CBF) for safety and Control Lyapunov Function (CLF) for control	Ensures safety constraints; robust control	It needs to define unsafe regions. If the unsafe region is too large, the space available for the robot to move will be limited.	[16-17, 19-20]
DBSCAN-MEE-CBF	Uses DBSCAN clustering to refine CBF-CLF safety constraints	Reduces the size of redundant, unsafe areas; improves feasible solutions	Requires additional computation for clustering	This study

### 3. Brief Review of CBF and CBF-CLF Quadratic Programming

In safety-critical systems, the system states must remain within the set defined by the safety constraint. One effective means for dealing with the abovementioned problem is to ensure that the system possesses the property of set invariance using CBF. CBF has been widely applied in safety-critical control tasks, which ensure the trajectories of an AGV remain within the safe regions. In addition, CLF provides a framework to let AGV move from the initial position to the final destination. The combination of CBFs and CLFs has been effectively used in quadratic programming [16-17, 19-20], allowing the system to stay in safe regions while moving from the initial position to the final destination. Consider a nonlinear control-affine system, where the system dynamics are a linear combination of the system state and control input. The system dynamics can be expressed as:

$$\dot{x} = f(x) + g(x)u \quad (1)$$

where  $f(\cdot): \mathbb{R}^n \rightarrow \mathbb{R}^n$  and  $g(\cdot): \mathbb{R}^n \rightarrow \mathbb{R}^m$ .

#### 3.1. Control Barrier Function (CBF)

The concept of the CBF is similar to that of the CLF. The goal of CBF is to ensure that the system states remain within the set defined by the safety constraints. Namely, the system possesses the property of set invariance. Consider a dynamical system described by Eq. (1), where  $f(\cdot)$ ,  $g(\cdot)$  are local Lipschitz continuous. According to the definition given in Xu [15], given a set  $C$  defined by a continuously differentiable function  $h(x): \mathbb{R}^n \rightarrow \mathbb{R}$  as described by:

$$\begin{aligned} C &= \{x \in \mathbb{R}^n : h(x) \geq 0\} \\ \partial C &= \{x \in \mathbb{R}^n : h(x) = 0\} \\ Int(C) &= \{x \in \mathbb{R}^n : h(x) > 0\} \end{aligned} \quad (2)$$

where  $\partial C$ ,  $Int(C)$  represents the set boundary and the interior of the set, respectively.

If there exists an extended class K function  $\alpha(\cdot)$  such that the system satisfies a certain inequality,

$$L_f h(x) + L_g h(x)u + \alpha(h(x)) \geq 0 \tag{3}$$

then, this  $h(x)$  is called the zeroing CBF. Based on this definition, a control law can be derived to ensure that the set  $C$  possesses the property of set invariance. The control law is expressed as:

$$K_{ZCBF} = \{u \in U : L_f h(x) + L_g h(x)u + \alpha(h(x)) \geq 0\} \tag{4}$$

### 3.2. CBF-CLF quadratic program

If the control system of a mobile robot only contains a CLF, then it can control the mobile robot moving toward the destination. However, the mobile robot cannot avoid obstacles. In contrast, if the control system of a mobile robot only contains a CBF, then the mobile robot can avoid obstacles but cannot control the mobile robot to reach its destination. One way to overcome this difficulty is by using the CLF as the control scheme while the CBF is used as a safety filter to ensure the safety of the mobile robot. To combine CLF and CBF, the CLF and CBF constraints can be reformulated as a quadratic program. The quadratic program is expressed as:

$$\begin{aligned} \min_{u \in U, \delta} & \frac{1}{2} u^T H u + F u + p \delta^2 \\ \text{s.t.} & L_f V(x) + L_g V(x)u + c V(x) \leq \delta \\ & L_f h(x) + L_g h(x)u + \alpha(h(x)) \geq 0 \end{aligned} \tag{5}$$

where  $H \in \mathbb{R}^{m \times n}$  is a positive definite matrix,  $F \in \mathbb{R}^m$  and  $p > 0$ . To prevent a situation wherein the constraints are too strict for feasible solutions, a relaxed term  $\delta$  is added to the inequality for the CLF to relax the constraint.

## 4. Proposed Methodology—DBSCAN and MEE

In this paper, the mobile robot is equipped with a LiDAR. The depth detected by the LiDAR for each angle  $\theta_{\text{LiDAR}}$  is denoted as  $d_{\text{LiDAR}}$ . Each pair of depth and angle (polar) coordinates can be converted into its corresponding Cartesian coordinates. The conversion is expressed as:

$$\begin{cases} x_i = d_{i,\text{LiDAR}} \cos(\theta_{i,\text{LiDAR}}) \\ y_i = d_{i,\text{LiDAR}} \sin(\theta_{i,\text{LiDAR}}) \end{cases}, \text{ for } i \in [0, 2\pi] \tag{6}$$

Note that these coordinates do not contain the information about the obstacles. This paper utilizes the DBSCAN approach [14] to perform clustering for the data points detected by the LiDAR. In particular, the DBSCAN approach performs clustering based on the density of the detected data points. In the DBSCAN approach, the data points are categorized into three classes—kernel points, boundary points, and noise points. Two parameters  $\epsilon$  and  $n_{\text{min\_points}}$  are used to categorize a given data point. If  $\epsilon = r$ ,  $n_{\text{min\_points}} = 4$ , then point  $A$  is a kernel point, point  $B$  is a boundary point, and point  $C$  is a noise point (Fig. 1).

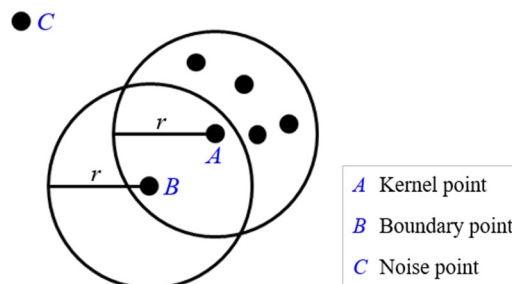


Fig. 1 Illustration of kernel points, boundary points, and noise points

For any given data point, define a circle with a radius, and its center is at this given data point. If the number of data points inside the defined circle is larger than  $n_{min\_points}$ , then this given data point belongs to the class of kernel points. Any data points that are not kernel points but are located inside a circle defined by any kernel points are called boundary points. For any data points that are neither kernel points nor boundary points, they are called noise points. For a kernel point  $P$ , all the data points that are located within the distance (radius)  $\in$  of  $P$  are called  $P$  reachability. For a kernel point  $P$ , if both data points  $P_1, P_2$  are  $P$  reachability, then data points  $P_1, P_2$  possess connectedness. As a result, one can define that any two data points inside the same cluster possess connectedness. Namely, if data point  $Q$  and data point  $P$  possess connectedness, they must be in the same cluster. Through the above clustering process, one can obtain the clustering results after several iterations.

Based on the clustering results of DBSCAN, the environment can be divided into several smaller regions containing obstacles. In the proposed obstacle avoidance algorithm, for each obstacle, an ellipse is used to enclose the obstacle and will be used to define the safety set of CBF. Firstly, find the minimum convex hull that encloses all the data points belonging to the obstacle, and the average  $p_{mean} = [x_{mean}, y_{mean}]$  of the data points enclosed by the convex hull is chosen as the center of the ellipse. Then, compute the covariance matrix  $\Sigma \in \mathbb{R}^2$ . Since this covariance matrix is symmetric, its eigenvectors are orthogonal to each other. Based on the eigenvalue  $\lambda_{max}$  with the largest modulus and its corresponding eigenvector  $v_{max}$ , one can compute the length  $a$  and the orientation  $v_{major}$  for the long axis of the ellipse. Similarly, according to the eigenvalue  $\lambda_{min}$  with the smallest modulus and its corresponding eigenvector  $v_{min}$ , one can compute length  $b$  and orientation  $v_{minor}$  for the short axis of the ellipse. As a result, one can find the MEE as described by:

$$\frac{(x' \cos(\phi) + y' \sin(\phi))^2}{a^2} + \frac{(y' \cos(\phi) - x' \sin(\phi))^2}{b^2} - 1 = 0 \quad (7)$$

where  $x' = x - x_{mean}$ ,  $y' = y - y_{mean}$ ,  $a = \sqrt{1.5\lambda_{max}}$ ,  $b = \sqrt{1.5\lambda_{min}}$ ,  $\phi = \tan^{-1}(v_{major,y}/v_{major,x})$ ;  $v_{major,x}$ ,  $v_{major,y}$  represent the  $x$  components and the  $y$  components of  $v_{major}$ , respectively, and the flowchart of the proposed approach is shown in Fig. 2.

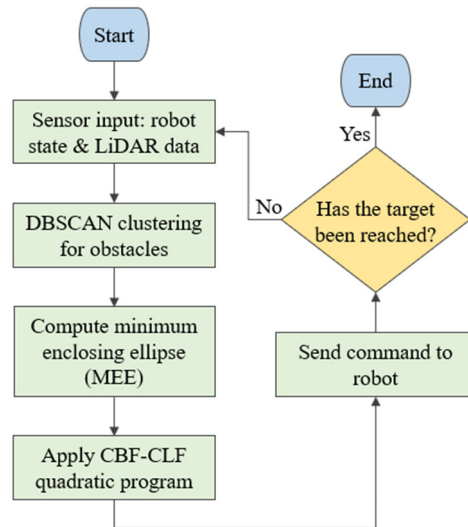


Fig. 2 Flowchart of the proposed approach

## 5. Simulation and Experimental Results

To validate the effectiveness of the proposed DBSCAN-MEE-CBF approach, both simulation and real-world experiments were conducted. The results from both evaluations demonstrate that the proposed approach successfully reduces the redundant unsafe area and improves the feasibility of motion planning. The details of these evaluations are presented in the following sections.

### 5.1. Simulation

TurtleBot3 Burger is used as the experimental platform in this paper. The state vector is denoted as  $x = [X, Y, \theta]^T \in \mathbb{R}^3$ , where  $X$  and  $Y$  are the Cartesian coordinates of the TurtleBot3 Burger and  $\theta$  is its orientation. The state equation is given as:

$$\dot{x} = \begin{bmatrix} \dot{X} \\ \dot{Y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} v \cos(\theta) \\ v \sin(\theta) \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \omega \quad (8)$$

In the simulation, the translation velocity command  $v$  is a constant, while the angular velocity command  $\omega$  is time-varying.

#### 5.1.1. The original CBF-CLF obstacle avoidance method

In the simulation, the initial state  $x_0 = [X_0, Y_0, \theta_0]^T = [0.0, 0.0, 0.0]^T$ , the goal state (i.e., destination) is  $x_{goal} = [X_{goal}, Y_{goal}, \theta_{goal}]^T = [-1.2, 1.2, 0.0]^T$ , where the units of  $X$  and  $Y$  are meters (m), and the unit for  $\theta$  is radian (rad). To verify the effectiveness of the CBF obstacle avoidance method, the continuously differentiable function  $h(x)$  is constructed to include the square of the distance, the derivative of the square of the distance, and the square of the safety distance.

$$h(x) = (X - X_{obs})^2 + (Y - Y_{obs})^2 - d_{safe}^2 + 2v \cos(\theta)(X - X_{obs}) + 2v \sin(\theta)(Y - Y_{obs}) \quad (9)$$

where  $X$ ,  $Y$ , and  $\theta$  are system states;  $X_{obs}$ ,  $Y_{obs}$  are the coordinates of the center of the obstacle; and  $d_{safe}$  is the pre-defined safety distance. In addition, one can use the CLF to ensure that the system state converges to the goal state. The CLF is defined as:

$$V(x) = (X - X_{goal})^2 + (Y - Y_{goal})^2 \quad (10)$$

Reformulating the above CBF and CLF yields a quadratic program.

$$\begin{aligned} \min_{u, \delta} & \frac{1}{2} (u - u_{ref})^T H (u - u_{ref}) + p \delta^2 \\ \text{s.t.} & L_f V(x) + L_g V(x) u + c_1 V(x) \leq \delta \\ & L_f h(x) + L_g h(x) u + c_2 h(x) \geq 0 \end{aligned} \quad (11)$$

The environment used in the simulation is illustrated in Fig. 3. Three rectangular obstacles consist of many detected data points. In the original CBF-CLF quadratic program, circles are used to construct the model for obstacles. Fig. 4 shows the simulation results for the original CBF-CLF quadratic program. Clearly, the new trajectory generated by the original CBF-CLF quadratic program can avoid obstacles and reach the goal state safely. In addition, the values of CBF for three obstacles shown in Fig. 5(a) are all positive, indicating that all the system states are in the safety set. In addition, the value of CLF shown in Fig. 5(b) converges exponentially. Fig. 6(a) shows the clustering results of the detected data points using the DBSCAN method. The MEE for these three obstacles obtained using Eq. (7) is shown in Fig. 6(b). Based on the obtained MEE, one can define the CBF with a new  $h(x)$ . It is given by:

$$\begin{aligned} h(x) = & \frac{\left[ (X - X_{obs}) \cos(\phi) + (Y - Y_{obs}) \sin(\phi) \right]^2}{(a + d_{safe})^2} + \frac{\left[ -(X - X_{obs}) \sin(\phi) + (Y - Y_{obs}) \cos(\phi) \right]^2}{(b + d_{safe})^2} \\ & + \frac{2 \left[ (X - X_{obs}) \cos(\phi) + (Y - Y_{obs}) \sin(\phi) \right] (v \cos(\theta) \cos(\phi) + v \sin(\theta) \sin(\phi))}{(a + d_{safe})^2} \\ & + \frac{2 \left[ -(X - X_{obs}) \sin(\phi) + (Y - Y_{obs}) \cos(\phi) \right] (-v \cos(\theta) \sin(\phi) + v \sin(\theta) \cos(\phi))}{(b + d_{safe})^2} - 1 \end{aligned} \quad (12)$$

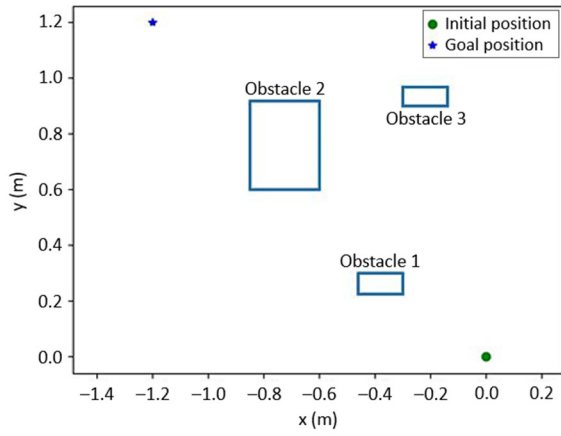


Fig. 3 Three rectangular obstacles were used in the simulation

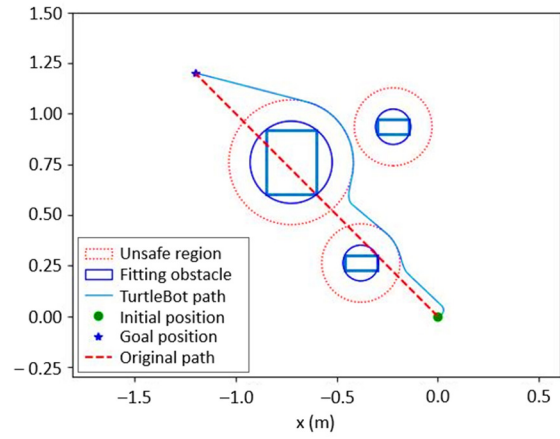
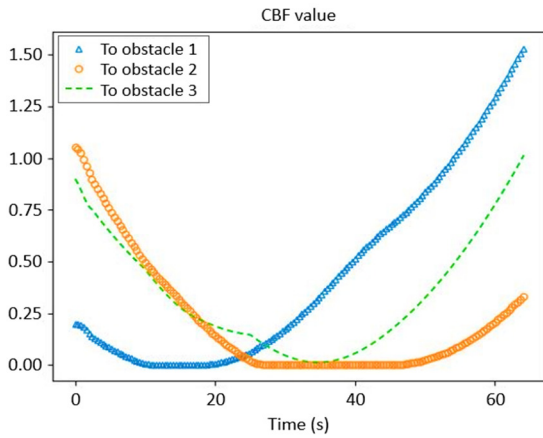
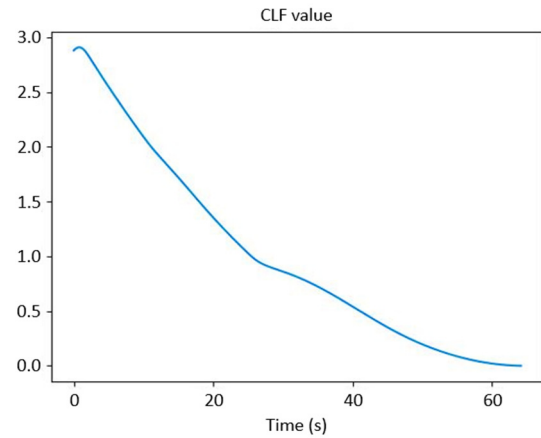


Fig. 4 Simulation results of the original CBF-CLF quadratic program

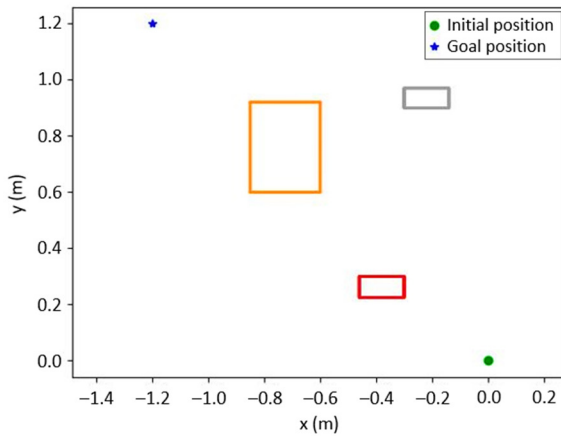


(a) Value of CBF

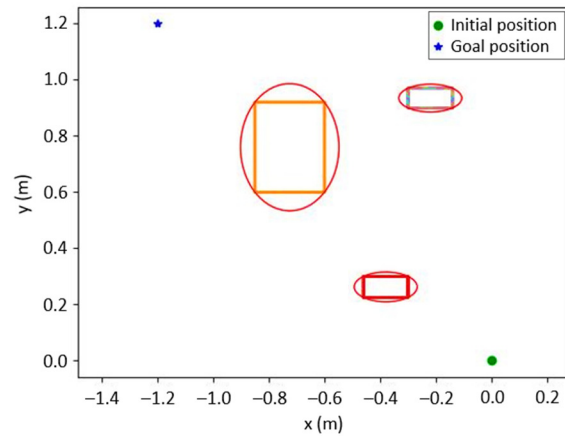


(b) Value of CLF

Fig. 5 Simulation results of the original CBF-CLF quadratic program



(a) Results of data point clustering using DBSCAN



(b) MEE Obtained from Eq. (7)

Fig. 6 Results of DBSCAN clustering and MEE

5.1.2. The proposed DBSCAN-MEE-CBF obstacle avoidance method

Fig. 7 shows the simulation results for the proposed DBSCAN-MEE-CBF obstacle avoidance method. The new trajectory can avoid obstacles and reach the goal state safely. Moreover, compared with the circles used to model the obstacles in the original CBF-CLF quadratic program, the ellipses used to model the obstacles in the proposed method fit more closely with the shape of the obstacle. In addition, the values of CBF for the three obstacles shown in Fig. 8(a) are all positive, indicating that all the system states are within the safety set. In addition, the value of CLF shown in Fig. 8(b) converges exponentially.

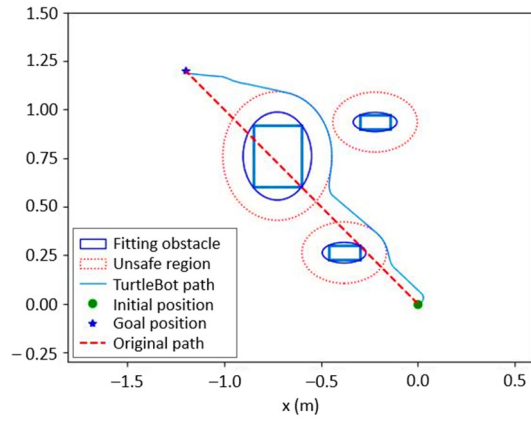
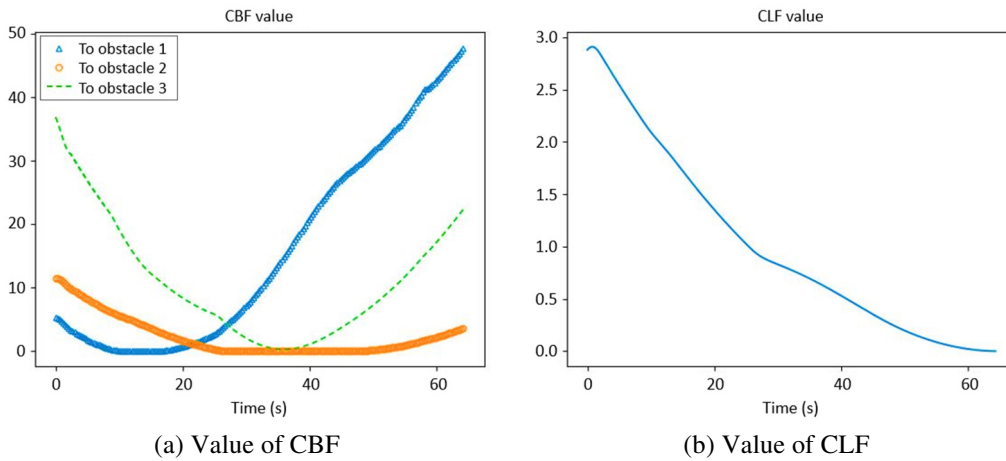


Fig. 7 Obstacle avoidance in the X-Y plane using DBSCAN-MEE-CBF



(a) Value of CBF

(b) Value of CLF

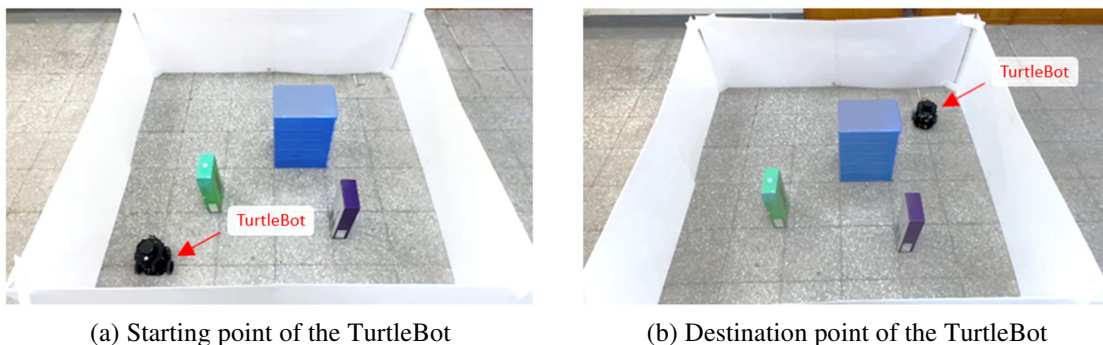
Fig. 8 DBSCAN-MEE-CBF obstacle avoidance

5.2. Experimental results

In the experiment, Python is used to solve the quadratic program, and ROS is used to broadcast the translation velocity command and rotational velocity command to the TurtleBot3 Burger. In addition, for safety concerns, an additional 1.5-centimeter distance will be added to the obstacle models for all methods tested in the experiment.

5.2.1. The original CBF-CLF quadratic program obstacle avoidance method

In this experiment, the original CBF-CLF quadratic program is used to avoid obstacles. Fig. 9 shows the image sequences of the TurtleBot moving from the starting point (Fig. 9(a)) to the destination point (Fig. 9(b)). Fig. 10 shows the new trajectory generated by the original CBF-CLF quadratic program obstacle avoidance method. The new trajectory can successfully avoid obstacles and reach the goal state safely. The values of CBF for three obstacles shown in Fig. 11(a) are all positive, indicating that all the system states are in the safety set. In addition, the value of CLF shown in Fig. 11(b) converges exponentially.



(a) Starting point of the TurtleBot

(b) Destination point of the TurtleBot

Fig. 9 Comparison of TurtleBot trajectories: original CBF-CLF vs. proposed DBSCAN-MEE-CBF

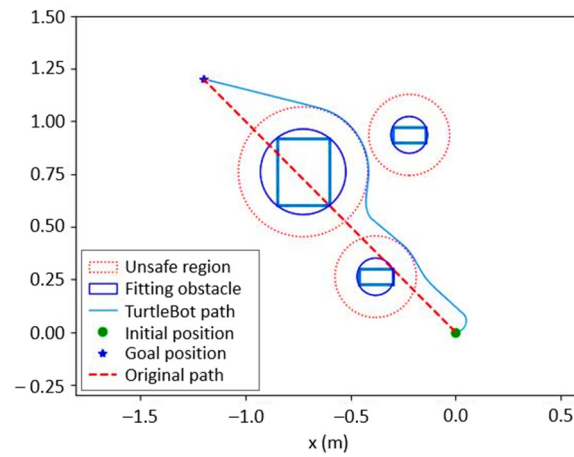


Fig. 10 X-Y plane trajectory using the original CBF-CLF quadratic program

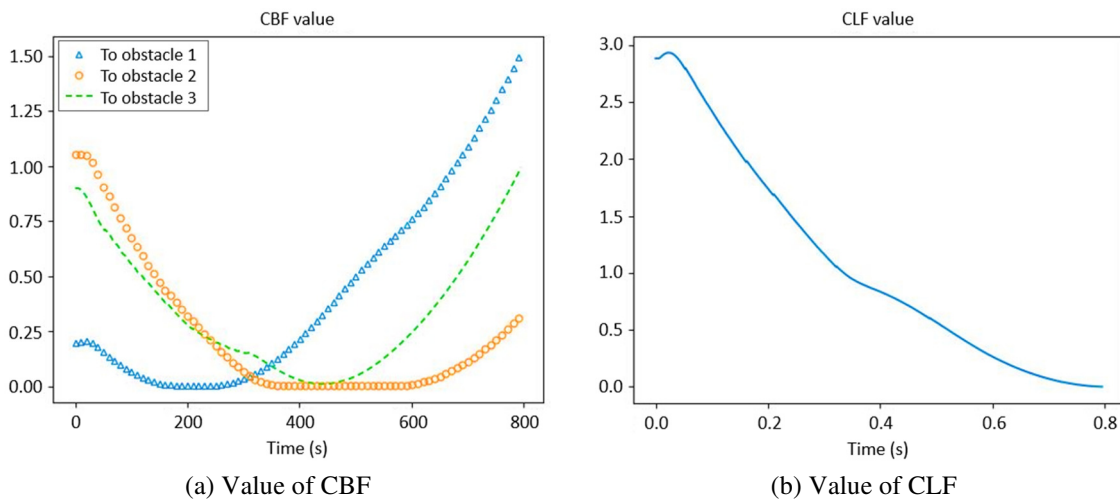


Fig. 11 Experimental results of the original CBF-CLF quadratic program

5.2.2. The proposed DBSCAN-MEE-CBF obstacle avoidance method

In this experiment, the proposed DBSCAN-MEE-CBF obstacle avoidance method is used to avoid obstacles. Fig. 12 shows the new trajectory generated by the proposed DBSCAN-MEE-CBF obstacle avoidance method. Unlike enclosing circles, which are used in Fig. 10, enclosing ellipses are adopted in Fig. 12. As a result, the size of the unsafe area is reduced, and the new trajectory can successfully avoid obstacles and reach the goal state safely. The values of CBF for three obstacles shown in Fig. 13(a) are all positive, indicating that all the system states are within the safety set. In addition, the value of CLF shown in Fig. 13(b) converges exponentially.

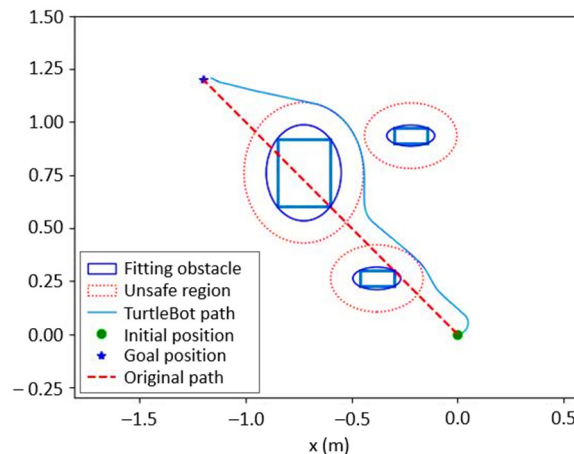


Fig. 12 X-Y plane trajectory using the proposed DBSCAN-MEE-CBF approach

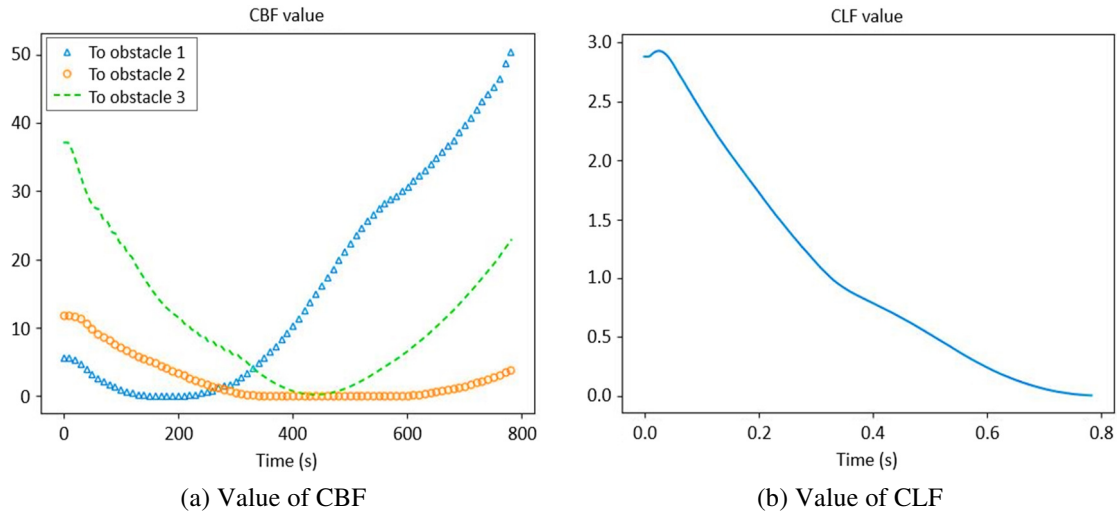


Fig. 13 Experimental results of the proposed DBSCAN-MEE-CBF obstacle avoidance method

5.2.3. Comparison of experimental results

As indicated by Table 2, the actual curve length for the original CBF-CLF quadratic program is longer than the actual curve length for the proposed DBSCAN-MEE-CBF obstacle avoidance method. Table 3 shows a comparison of the area of the enclosing circle used in the original CBF-CLF quadratic program and the area of the enclosing ellipse used in the proposed DBSCAN-MEE-CBF obstacle avoidance method without considering the extra safety distance. Table 4 shows a comparison of the area of the enclosing circle used in the original CBF-CLF quadratic program and the area of the enclosing ellipse used in the proposed DBSCAN-MEE-CBF obstacle avoidance method by considering the extra safety distance.

Table 2 Actual curve length

Method	Length (m)
Original CBF-CLF quadratic program	1.95962927
Proposed DBSCAN-MEE-CBF obstacle avoidance method	1.923764028

Table 3 Enclosing area analysis: original CBF-CLF vs. proposed DBSCAN-MEE-CBF

	Enclosing circle	Enclosing ellipse	Comparison between the enclosing circle and the enclosing ellipse	
	Area (m <sup>2</sup> )	Area (m <sup>2</sup> )	Area difference (m <sup>2</sup> )	Percentage of area difference (%)
Obstacle 1	0.024550164	0.018855005	0.005695159	23.2
Obstacle 2	0.129461892	0.125658986	0.003802906	2.937
Obstacle 3	0.023942989	0.017592874	0.006350115	26.52

Table 4 Area comparison between enclosing circle and enclosing ellipse in obstacle avoidance methods

	Enclosing circle	Enclosing ellipse	Comparison between the enclosing circle and the enclosing ellipse	
	Area (m <sup>2</sup> )	Area (m <sup>2</sup> )	Area difference (m <sup>2</sup> )	Percentage of area difference (%)
Obstacle 1	0.117506749	0.108311789	0.00919496	7.825
Obstacle 2	0.298024045	0.293247445	0.0047766	1.602
Obstacle 3	0.116173866	0.105878397	0.010295469	8.86

Both Table 3 and Table 4 indicate that the proposed DBSCAN-MEE-CBF obstacle avoidance method yields a smaller area for enclosing obstacles. Namely, the size of the redundant unsafe area of obstacles can be reduced. Therefore, the mobile robot can have a larger safe space to move in, so the likelihood of finding a feasible solution will increase. The purpose of the proposed DBSCAN-MEE-CBF method is to provide a safer space for the robot during navigation compared with that obtained

from the conventional enclosing circle approach. Additionally, compared with traditional methods [N3], the use of CBF in this paper enables the robot to successfully avoid obstacles while ensuring that it does not enter restricted areas. This makes the approach not only applicable to TurtleBot3 but also extendable to other robots.

## 6. Conclusion and Future Work

This paper presented the DBSCAN-MEE-CBF approach, which integrates DBSCAN clustering with the CBF-CLF quadratic program to determine the MEE for obstacle representation. This method aimed to reduce the redundant unsafe area in obstacle avoidance. A TurtleBot running on ROS was used as the simulation and experimental platform to validate the effectiveness of the proposed approach. Both simulation and experimental results confirmed that, compared with the original CBF-CLF quadratic program, the proposed DBSCAN-MEE-CBF approach can effectively reduce the redundant unsafe area of obstacles, thereby achieving safer and more feasible navigation. The main findings are summarized as follows:

- (1) The DBSCAN-MEE-CBF approach reduces the redundant unsafe area around obstacles, providing a larger feasible space for navigation.
- (2) Compared with the traditional CBF-CLF method, the proposed approach alleviates unnecessary safety constraints and improves path feasibility.
- (3) Both simulation and experimental results validated the effectiveness and practicality of the proposed method for obstacle avoidance.

In addition to these conclusions, several directions for future research are highlighted. The method could be extended to handle dynamic obstacles with real-time updates to obstacle boundaries for applications in complex environments. Furthermore, integrating reinforcement learning or other AI-driven decision-making techniques with DBSCAN-MEE-CBF could enhance its adaptability, enabling autonomous robots to learn optimal navigation strategies under varying environmental conditions.

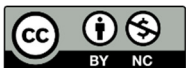
## Conflicts of Interest

The authors declare no conflict of interest.

## References

- [1] S. Sotnik and V. Lyashenko, "Modern Industrial Robotics Industry," *International Journal of Academic Engineering Research*, vol. 6, no. 1, pp. 37-46, 2022.
- [2] Y. K. Hwang and N. Ahuja, "A Potential Field Approach to Path Planning," *IEEE Transactions on Robotics and Automation*, vol. 8, no. 1, pp. 23-32, 1992.
- [3] C. Zong, Q. Du, J. Chen, Y. Shan, Y. Wu, and Zhida Sha, "A Design of Three-Dimensional Spatial Path Planning Algorithm Based on Vector Field Histogram\*," *Sensors*, vol. 24, no. 17, article no. 5647, 2024.
- [4] S. Koley, "Role of Fluid Dynamics in Infectious Disease Transmission: Insights from COVID-19 and Other Pathogens," *Trends in Sciences*, vol. 21, no. 8, article no. 8287, 2024.
- [5] A. N. A. Rafai, N Adzhar, and N. I. Jaini, "A Review on Path Planning and Obstacle Avoidance Algorithms for Autonomous Mobile Robots," *Journal of Robotics*, vol. 2022, no. 1, article no. 2538220, 2022.
- [6] H. I. Lin and M. F. Hsieh, "Robotic Arm Path Planning Based on Three-Dimensional Artificial Potential Field," *18th International Conference on Control, Automation and Systems*, pp. 740-745, 2018.
- [7] J. Chen, X. Zhang, X. Peng, D. Xu, and J. Peng, "Efficient Routing for Multi-AGV Based on Optimized Ant-Agent," *Computers & Industrial Engineering*, vol. 167, article no. 108042, 2022.
- [8] C. Duan and S. Li, "Research on AGV Route Planning Based on Improved Artificial Potential Field Algorithm," *Proceedings of the 3rd Asia-Pacific Conference on Image Processing, Electronics and Computers*, pp. 338-342, 2022.
- [9] R. Szczepanski, T. Tarczewski, and K. Erwinski, "Energy Efficient Local Path Planning Algorithm Based on Predictive Artificial Potential Field," *IEEE Access*, vol. 10, pp. 39729-39742, 2022.

- [10] T. Xu, H. Zhou, S. Tan, Z. Li, X. Ju, and Y. Peng, "Mechanical Arm Obstacle Avoidance Path Planning Based on Improved Artificial Potential Field Method," *Industrial Robot*, vol. 49, no. 2, pp. 271-279, 2022.
- [11] Z. Fang and X. Liang, "Intelligent Obstacle Avoidance Path Planning Method for Picking Manipulator Combined with Artificial Potential Field Method," *Industrial Robot*, vol. 49, no. 5, pp. 835-850, 2022.
- [12] X. Xia, T. Li, S. Sang, Y. Cheng, H. Ma, Q. Zhang, et al., "Path Planning for Obstacle Avoidance of Robot Arm Based on Improved Potential Field Method," *Sensors*, vol. 23, no. 7, article no. 3754, 2023.
- [13] J. Borenstein and Y. Koren, "The Vector Field Histogram-Fast Obstacle Avoidance for Mobile Robots," *IEEE Transactions on Robotics and Automation*, vol. 7, no. 3, pp. 278-288, 1991.
- [14] I. Ulrich and J. Borenstein, "VFH/sup \*/: Local Obstacle Avoidance with Look-Ahead Verification," *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, vol. 3, pp. 2505-2511, 2000.
- [15] T. Xu, "Recent Advances in Rapidly-Exploring Random Tree: A Review," *Heliyon*, vol. 10, no. 11, article no. e32451, 2024.
- [16] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control Barrier Function Based Quadratic Programs for Safety Critical Systems," *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 3861-3876, 2017.
- [17] E. A. Basso and K. Y. Pettersen, "Task-Priority Control of Redundant Robotic Systems Using Control Lyapunov and Control Barrier Function Based Quadratic Programs," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 9037-9044, 2020.
- [18] E. Schubert, J. Sander, M. Ester, H. P. Kriegel, and X. Xu, "DBSCAN Revisited, Revisited: Why and How You Should (Still) Use DBSCAN," *ACM Transactions on Database Systems*, vol. 42, no. 3, pp. 1-21, 2017.
- [19] I. Jang and H. J. Kim, "Safe Control for Navigation in Cluttered Space Using Multiple Lyapunov-Based Control Barrier Functions," *IEEE Robotics and Automation Letters*, vol. 9, no. 3, pp. 2056-2063, 2024.
- [20] H. Dai, C. Jiang, H. Zhang, and A. Clark, "Verification and Synthesis of Compatible Control Lyapunov and Control Barrier Functions," *IEEE 63rd Conference on Decision and Control*, pp. 8178-8185, 2024.



Copyright© by the authors. Licensee TAETI, Taiwan. This article is an open-access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY-NC) license (<https://creativecommons.org/licenses/by-nc/4.0/>).