

Autumn Semester, 2024

## GRADUATION THESIS

# **Non-circular Control Barrier Function Technique for Mobile Robots Collision Avoidance in Complex Environments**

ACADEMIC SUPERVISOR: SVININ Mikhail, LIU Wenbin

College of Information Science and Engineering  
Ritsumeikan University  
Graduation Research 3 (GD)

Name: JIA Shucheng  
Student ID: 26002104600  
Date: 2025/01/28

## **Abstract**

This study introduces a novel Super-Ellipse Control Barrier Function (SE-CBF) to enhance obstacle avoidance and navigation efficiency in mobile robots operating in complex environments. Unlike traditional CBF methods that rely on fixed safety distances, SE-CBF dynamically adjusts safety distance based on obstacle geometry and environmental conditions. Through simulation experiments and real-world validations, SE-CBF demonstrated superior performance in minimizing path redundancy, improving space utilization, and reducing task completion times in dynamic multi-agent scenarios. The proposed approach integrates with the Robot Operating System (ROS) and is validated using Raspberry Pi Mouse V3 robots. Key results indicate that SE-CBF outperforms conventional methods in adaptability, safety, and navigation efficiency. This study sets a foundation for scalable and efficient navigation strategies in autonomous systems, paving the way for applications in robotics, logistics, and intelligent transportation systems.

# Contents

<b>List of figures.....</b>	<b>3</b>
<b>1    Introduction.....</b>	<b>5</b>
1.1    Background.....	5
1.2    Literature Review .....	6
1.3    Problem statement .....	11
1.4    Research Contributions .....	11
<b>2    Design of a flexible obstacle avoidance Technique .....</b>	<b>14</b>
2.1    Control problem formalization .....	14
2.2    A flexible Technique Design.....	18
<b>3    Simulation Models and Experimental Platform .....</b>	<b>23</b>
3.1    Software Components .....	23
3.1.2    Gazebo Simulation Environment.....	24
3.2    Hardware Components .....	27
3.2.1    Raspberry Pi Mouse V3.....	27
3.3    System Architecture .....	30
<b>4    Simulation and Experimental Results.....</b>	<b>33</b>
4.1    Simulation Experiment Scenarios .....	33
4.2    Real-World Experimental.....	66
<b>5    Discussion and Conclusion .....</b>	<b>69</b>
<b>References .....</b>	<b>71</b>

# List of figures

Figure 1: CBF safety distance demonstration.....	16
Figure 2: Fixed safety distance limitations of CBF .....	17
Figure 3: Visualization of formula (8) .....	19
Figure 4: SE-CBF parameters shape.....	21
Figure 5: Gazebo multi-robots demonstrated .....	25
Figure 6: Topic tree of multi-robot simulator.....	26
Figure 7: Raspberry Pi Mouse mobile robot.....	28
Figure 8: Intel RealSense D435 Camera.....	29
Figure 9: ArTag Markers with Various IDs.....	29
Figure 10: Experimental Setup .....	31
Figure 11: System Architecture .....	32
Figure 12: Intersection Simulation with 8 robots .....	34
Figure 13: Snapshots for the CBF Intersection under simulation:.....	36
Figure 14: Trajectories path of 8 robots in the CBF intersection simulation .....	37
Figure 15: Trajectories of the x and y positions in the CBF intersection simulation .....	38
Figure 16: Snapshots for the SE-CBF Intersection under simulation:.....	39
Figure 17: Trajectories path of 8 robots in the SE-CBF intersection simulation .....	40
Figure 18: Trajectories of the x and y positions in the .....	41
Figure 19: Boxplot of Average Minimum Distance .....	43
Figure 20: Boxplot of Average Speed .....	44
Figure 21: Boxplot of Completion Time .....	46
Figure 22: Unknown area exploration .....	48
Figure 23: Snapshots for the Four Robots Result:.....	50
Figure 24: Trajectory map of 4 robots .....	51
Figure 25: Trajectories of 4 robots the x and y positions .....	51
Figure 26: Snapshots for the Eight Robots Result:.....	52
Figure 27: Trajectory map of 8 robots .....	53
Figure 28: Trajectories of 8 robots the x and y positions .....	55
Figure 29: Snapshots for the Twelve Robots Result: .....	56
Figure 30: Trajectory map of 12 robots .....	57
Figure 31: Trajectories of 12 robots the x and y positions .....	59

Figure 32: Task Completion Time .....	60
Figure 33: Average Path Length and Complexity .....	62
Figure 34 Minimum Distance .....	64
Figure 35: Snapshots for the SE-CBF Intersection under simulation:.....	66
Figure 36: Real-wrld Experiment Trajectories Map of four robots.....	67
Figure 37: Real-world Experiment Minimum Safety Distance .....	68

# **Chapter 1**

## **Introduction**

### **1.1 Background**

Since 2020, automation and robotics have begun to develop rapidly. These new and advanced technologies are constantly changing global industrial production and people's daily lives. The most representative examples of the use of robotics applications are autonomous vehicles, unmanned delivery systems, and intelligent service robots[1]. At the same time, Robots are also playing an increasingly important role in many fields such as logistics, medicine, agriculture, and urban construction. They are accelerating the process of the fourth industrial revolution. Because these technologies have helped improve production efficiency, reduce labour costs, and minimize safety risks. Ultimately, they could assist people in the pursuit of intelligent transformation in daily life. For example, in the field of warehousing and logistics, automated guided vehicles (AGVs) and autonomous mobile robots (AMRs) have greatly improved the efficiency of goods sorting and transportation[2][4].

Despite the significant progress mobile robots have made in various fields and their successful application in many real-world scenarios. There are still many areas where the autonomous navigation and obstacle and collision avoidance capabilities of mobile robots need improvement, especially in navigation and obstacle avoidance in complex dynamic environments. In real-life applications such as delivery robots, robots need to face many complex environments and situations. Robots need to move in crowded pedestrian flows, unexpected obstacles, and complex unstructured environments; at the same time, they have to avoid obstacles and reach their destination safely[2].

Therefore, Obstacle and Collision Avoidance is one of the essential and constantly evolving core technologies for the safe operation of mobile robots in dynamic environments[2]. Obstacle avoidance can guarantee that the robot detects obstacles in the environment in real-time and also ensures that the trajectory is dynamically adjusted during the path-planning process to avoid them. A good example of this phenomenon is found in warehousing and logistics. The implementation of Automated Guided Vehicles (AGVs) and Autonomous Mobile Robots (AMRs) has improved the efficiency of sorting and transportation of goods[4].

In recent years, engineers have developed a series of basic obstacle avoidance technologies to develop and improve different obstacle avoidance technologies. Among these different methods,

the Control Barrier Function (CBF) method, which has gained much attention in the last two years, has also been developed and improved. CBF defines a dynamic safety constraint distance between the robot and the obstacle or other agents. CBF helps the robot to avoid the risk of collision and adjust its trajectory in real-time based on their target and obstacle places[4]. In mathematical terms, when CBF achieves obstacle avoidance, it also assures high real-time performance, robustness of control systems, and stability and continuity of the robot's movement. This will actually enable the robot system to avoid obstacles in complex dynamic environments effectively. In fact, the introduction of CBF provides new solutions for mobile robot obstacle avoidance in complex environments.

However, there are also places that need improvement in the current CBF method. This method usually uses fixed safety distance constraints to prevent the robot from getting too close to dangerous areas. Sometimes, when facing irregularly shaped or dynamically changing obstacles, this strategy will reduce the robot's flexibility and space utilization efficiency. This research aims to improve and enhance the limitations of CBF and test the practicality and functionality of new CBF methods in different environments.

## 1.2 Literature Review

### 1.2.1 Obstacle and Collision Avoidance

Obstacle and collision avoidance is a necessary part of robotic control systems. It can ensure the safety, reliability, and functionality of mobile robots and autonomous navigation systems when running in unpredictable and complex environments. Obstacle detection and avoidance are key to autonomous operation. It affects and determines system performance and efficiency[5][6]. This capability plays a crucial role across various fields of industrial automation, logistics, autonomous vehicles, and drones. Efficient obstacle avoidance reduces the operational downtime of a system by ensuring higher reliability through curtailing accident risks. In other words, this can be considered to guarantee more extended service lives of a system by minimizing friction and abrasion caused by a possible collision[6]. Therefore, this calls for developing and implementing efficient obstacle avoidance mechanisms that ensure high performance and long-term viability for an autonomous system.

For instance, Lee et al.[5] propose an advanced robotic system that is designed to assist postal workers in navigating through complex urban environments. The research addresses the contribution of obstacle and collision avoidance technology in mobile robots to much-improved navigation within unstructured heavy-population and unpredictable urban environments. These technologies will be able to provide real-time course modifications and decrease the chances of incidents or discontinuation. A similar study by Meng et al.[6] presented an advanced collision

avoidance framework for mobile robots operating in dynamic environments. Their work has shown the importance of obstacle avoidance in AGVs and driverless trucks, where continuous navigation in dynamic environments is necessary. Without efficient detection and avoidance, automated systems cannot be truly effective or scalable.

Continuing from that, Cheng et al.[7] presented a design for a four-wheel adaptive robot positioning and navigation system based on the ROS. Most importantly, their study emphasized that in indoor settings with semi-structured environments, both obstacle avoidance and collision prevention are equally critical. They highlighted that mobile robots must adopt adaptive positioning and navigation strategies to interact with the real world. Effective obstacle avoidance thus becomes an indication of a system's productivity within such scenarios. Productivity, in most instances, promises huge flexibility in operations within a range of different industries. Therein, there exists a serious call to robustify the avoidance method along with the adaptive navigation to unleash all the potential in real dynamic situations using mobile robotic systems.

These studies together emphasize the critical advancements in obstacle and collision avoidance[7]. The importance of obstacle and collision avoidance in autonomous systems is well-established. In fact, with time, the various techniques developed for collision avoidance have literally evolved, reflecting the ever-growing complexity and demands posed by applications. Simple rule-based techniques have evolved into complex, adaptive algorithms. Early solutions used basic detection and avoidance tactics that worked in simple situations but failed in dynamic and complicated contexts. Recent work[5][6][7] has established more robust methods, including machine learning, real-time sensor fusion, and predictive modelling. These advancements improve previous methods and enable more efficient and durable autonomous systems.

Traditional approaches for obstacles and collision avoidance have been imperative in the development of autonomous navigation systems, providing the foundational framework for the design of the motion planning strategy for a mobile robot. Among these methods, the Artificial Potential Field (APF) is one of the most widely adopted approaches. In APF, obstacles are modelled as repulsive forces and targets as attractive forces, whereby a virtual force field created by these forces guides the motion of the robot[8][9]. However, it has the reputation of being plagued with local minima—when a robot gets trapped within a competitive compromise of forces. In their approach to an attempt at overcoming these detriments of APF, Meng et al. [6] introduced an enhanced version of the APF algorithm that dynamically changes potential fields and incorporates obstacle escape strategies to prevent the robot from stalling situations in complex environments. This improvement in navigation, in general, results in a 30% increase in navigation efficiency. Thus, it is an important improvement of autonomous systems in operation at a logistics or warehouse facility.

Another famous method is the Vector Field Histogram (VFH). VFH utilizes a histogram grid to determine obstacle-free regions around the robot[11]. VFH minimizes the amount of steering

angle for the determination of the best direction through the evaluation of obstacle densities. The application of this approach goes to such spheres as autonomous delivery robots.

Besides deterministic methods like APF and VFH, Genetic Algorithms (GA) become one of the strongest methods for obstacle avoidance and path optimization. Inspired by the principles of natural selection, GA repetitively evolves potential solutions to find optimal paths, such as minimizing travel distance and avoiding obstacles[12]. The evolution progression from classical methods like APF and VFH toward the incorporation of hybrid methods using GA exemplifies the ongoing drive toward even better obstacle avoidance approaches.

Autonomous navigation has traditionally relied on conventional methods for avoiding obstacles and collisions. While these traditional approaches are effective in simple and static scenarios, they often struggle to meet the demands of more dynamic and complex environments. Emerging techniques such as Control Barrier Functions (CBF) have become quite popular since they can overcome the deficiencies of traditional methods. CBF provides a radical step forward in obstacle-avoiding schemes by emphasizing real-time safety enforcement and adaptive control[4].

### **1.2.2 Introduction to Control Barrier Functions (CBF)**

For autonomous navigation systems, security is a crucial point. Control Barrier Functions (CBF) have been proposed as a key solution that provides safety mechanisms and improves both adaptability and robustness. CBF has been utilized as a tool to constrain dynamic systems for safety. CBF will allow the system to maintain a predefined safe distance from other objects, such as other agents or obstacles[13]. The safety requirements for CBF are directly included in the control procedure. In this way, the CBF changes system movement during the operation to achieve safe navigation and collision avoidance. The combination of control avoidance mechanisms and algorithms allows for accurate responses of the system. Moreover, Quadratic Programming (QP) enforces this mechanism. QP optimizes robots and autonomous systems to have a quick reflection of the obstacle[13][14].

Unlike traditional collision avoidance techniques such as Artificial Potential Fields (APF) or Vector Field Histograms (VFH), CBF provides safety guarantees by applying constraints at each control step. CBF guarantees the robot stays within safe operating bounds without the need for extensive global path planning[8][9][14]. In addition, a comparison of this approach with Genetic Algorithms (GA) reveals a notable outstanding feature. GA works as a global optimization method by improving solutions through evolutionary techniques such as selection, crossover, and mutation. Because this evolutionary process is uncertain, it might produce less straightforward or complex pathways. CBF employs a different strategy. It balances computational efficiency with adaptability. It is the more suitable method of real-time navigation[12]. However, despite its advantages, CBF

can sometimes be over-cautious, establishing a higher priority safety at the cost of optimal performance. This issue is similar to GA the overly conservative paths. While this trade-off may reduce efficiency, CBF is usually acceptable since safety is a top priority in autonomous navigation. The ability to balance safety, adaptability, and computational efficiency makes CBF a popular choice for solving the challenges of modern autonomous navigation in complex environments.

Various applications and successes of the CBF method demonstrate its value for ensuring the safety and adaptability of a single autonomous agent. It naturally scales to address more complicated multi-agent interaction scenarios. This progression is especially evident in the field of Multi-Agent Systems (MAS), where coordinated behaviour and collision avoidance become increasingly difficult as the number of agents grows[15].

Multi-agent systems (MAS) have become one of the most important fields of modern research due to their capacity for solving tasks. The fundamental idea behind MAS is that in order for agents to successfully accomplish common objectives, coordination and collaboration are essential. However, a significant challenge faced by MAS is that agents have to collaborate among agents while avoiding conflicts and achieving stability at the same time. Lan et al.[16] demonstrated the potential of multi-agent deep reinforcement learning (MADRL) in optimizing coordination within a multi-robot pick-and-place system. Their findings emphasized that promoting cooperative behaviours among robots can improve task efficiency, even in dynamic and unpredictable environments. Building on this concept, Schwartz et al.[17] extended the application of MADRL to connected autonomous vehicles. They proposed a safe multi-agent reinforcement learning framework and demonstrated that cooperation can improve safety and traffic conditions under difficult and unpredictable driving conditions. These investigations confirm MADRL's multifaceted benefits and MAS's limitations.

Reinforcement learning can improve coordination and efficiency, but MAS safety still also requires other techniques. In such scenarios, Control Barrier Functions (CBF) are highly useful for achieving safety and stability especially for navigation and avoiding collisions. CBF provides a mechanism for agents to maintain safe distances from other agents, and MAS can adjust paths to avoid obstacles. Tan et al.[18] proposed a distributed framework that combines CBF with QP to ensure guarantee collision avoidance for multi-agent systems in real-time. Expanding on this concept, researchers also developed a decentralized safe control framework using responsibility-associated CBF[18]. This technique seems promising in scenarios with a high density of agents that possess increased challenges for safety and stability control. These developments illustrate the more robust and efficient multi-agent systems in that they begin to reveal the supporting nature of MADRL and CBF in addressing the diverse problems experienced with MAS.

CBF's smooth integration into MAS develops scalable and efficient navigation strategies in complex environments. As the demand for autonomous multi-robot systems grows, CBF provides a mathematical way of ensuring safety with minimal performance compromise. All of these

improvements in CBF-based safety methods and multi-agent learning are making the way for the next wave of multi-agent systems that are strong and flexible.

### 1.2.3 Collision Avoidance Algorithms in Complex Environments

Obstacle avoidance in complex environments is a crucial challenge for autonomous systems, especially when applying CBF. The main problem is how to adapt to dynamic and irregular obstacles. The traditional methods of CBF cannot deal with the positions and shapes that change rapidly since their safety margins are fixed and cannot capture irregular geometries[20]. These limitations of CBF usually result in conservative or simplified avoidance strategies that are not suitable for an unpredictable environment.

In high-density environments, multiple agents navigating systems sometimes have to face some problems, such as path conflicts and inefficient use of space. Although CBFs are effective in ensuring safety, this technique may bring some issues such as bottlenecks and excessive detours. These challenges reduce system operation efficiency and increase the risk of deadlocks[21]. These constraints emphasize the need for a more flexible and adaptive CBF technique that can solve irregular barriers without losing computational efficiency or stability. To address this need, Extent-Compatible Control Barrier Functions (ECBF) provide a significant advancement that overcomes the physical constraints and limitations inherent in CBF models[22]. The ECBF mechanism considers and includes the physical dimensions and shape of agents in the safety framework. ECBF ensures that the whole volume of an agent stays within the safe set, unlike classical CBF, which only simply ensures that the centre point is safe. With these considerations integrated, ECBF enables better navigation and accurate obstacle avoidance in complex and dynamic situations. Execution of ECBF supplies the novel and efficient solution of a high-density multi-agent system problem.

Despite developments, ECBF and other collision avoidance techniques still face notable limitations. One major problem is low effectiveness when the technique works with irregularly shaped obstacles. To be detailed, ECBF and CBF struggle with directional constraints, where different safety distances This limitation restricts the adaptability of the CBF in an environment with non-uniform obstacles. This is because the CBF can only provide fixed constraints. Another challenge is the computational and algorithmic complexities associated with the sum of squares optimization, a modification in ECBF. These problems inhibit the real-time use of many algorithms, especially for use in higher dimensional or large-scale systems requiring decisions to be made quickly.

## 1.3 Problem statement

Advancements in collision avoidance methods, such as ECBF have significantly enhanced obstacle avoidance capabilities. However, several critical challenges remain unresolved, especially in complex and dynamic environments. Traditional CBF and even advanced ECBF approaches typically lack the ability to adapt effectively to irregular obstacles and rapidly changing conditions. Fixed safety margins utilized render them oblivious to changes that may be occurring in real-time. It is the nature of these systems that incorporates rigidity into a system, with the routes computed by a system usually being conservative. While this ensures there is safety across the system, it consequently leads to a reduction in autonomous system efficiency[24].

In multi-agent environments, the difficulties become far more pronounced. For example, in multi-agent interaction, maintaining fixed safety distances may result in poor coordination and inefficient utilization of the available space for all agents. This problem, in turn, negatively affects system path planning and operational efficiency. The risk of congestion in dense environments or of standstill in less crowded environments is bound to escalate. To address these issues, new CBF techniques are necessary, which incorporate adaptive safety distance that dynamically is changed based on the requirement. Additionally, new coordination strategies need to be developed to ensure safety while performing better during complex interactions among different agents.

## 1.4 Research Contributions

This study aims to propose and validate a new type of Control Barrier Function based on the concept of a Super-Ellipse. This novel control system is designed to greatly enhance the obstacle avoidance capabilities, navigation efficiency, and operational flexibility of mobile robots in complex environments. The method involves establishing a fluid, adaptive, high-resilience safety zone that makes obstacle avoidance more accurate and responsive. This innovation will dramatically improve the navigation performance of robotic systems. The specific objectives and contributions of the research can be summarized under three major aspects:

- **Dynamic Safety Distance Model**

This research proposes a new Super-Ellipse CBF (SE-CBF) method. This method allows for control over the geometric properties of the super-ellipse mathematical function by permitting different constraint axis lengths in various directions. More precisely, by adjusting the safety distance constraints along the major and minor axes, the system enables robots to change obstacle avoidance zones for complex-shaped obstacles. For instance, the new control method facilitates the

extension and relaxation of the safety distance for sparser or regularly shaped obstacle scenarios to maximize path space utilization. Conversely, when operating in scenarios that contain numerous complex-shaped or irregularly shaped obstacles or agents, the safety distance shrinks to fit the contours of the obstacles. The contour of the safe zone was specifically adjusted to match the shape of the irregular objects. This dynamic adjustment mechanism prevents excessive movements, achieves more accurate avoidance, and smoothly works within limited space. This improves the robot's navigation efficiency and adaptability to more intricate navigation environments.

- **Optimizing Space Utilization and Reducing Redundant Safety Zones**

Space utilization and reducing redundant safety areas is one of the key indicators of robot navigation performance. However, traditional CBF methods often lead to excessive redundant paths in multi-obstacle environments. This occurs because the safety distance remains constant in all directions, resulting in unnecessarily large safety zones even in directions where obstacle avoidance is not required.

The SE-CBF new concept introduced in this study dynamically adjusts the geometric shape and the safety zone. This new technique minimizes space waste and optimizes navigation performance. In dense areas, SE-CBF enables the robotic system to shift towards the obstacle while still ensuring safety by narrowing the safety zone on its shorter axis. In a wider area, SE-CBF enlarges the safety zone along the wider axis, ensuring a sufficient safe distance from multiple obstacles. This strategy reduces the possibility of collision while avoiding collision. The results show that robots achieve higher success rates when navigating through areas with multiple obstacles. Path planning efficiency is significantly improved, with path redundancy decreasing by 20% to 30%.

By focusing on spatial constraints in obstacle-dense environments, SE-CBF enhances the system's adaptability and safety. These advancements make it a robust solution for complex navigation scenarios, effectively addressing the limitations of traditional CBF methods.

- **Validating Method Effectiveness Through Simulation and Real-World Experiments**

To demonstrate the effectiveness of the proposed new technique, this study conducted comprehensive tests and validations of the SE-CBF method in both simulated environments and real-world scenarios. By comparing it with traditional CBF methods, the experiments evaluated its obstacle avoidance capability, navigation speed, and overall efficiency in complex environments. In the simulation experiments, two scenarios were used for comparison: a classic traffic intersection

and an unknown area exploration. These simulations tested SE-CBF's adaptability and performance advantages across different environments.

In the study of traffic intersections, this study focuses on verifying the completion efficiency and safety performance of SE-CBF compared with traditional CBF in dynamic multi-agent environments. Traffic intersections, as typical representatives of complex scenarios, include high-frequency path intersections, dynamic obstacles, and limited space, which impose stringent requirements on the real-time and reliability of obstacle avoidance algorithms. On the other hand, during the unknown area exploration experiment, traditional CBF methods faced operational limitations and could not run properly. This second scenario of the test focused primarily on evaluating SE-CBF's performance. The results showed that SE-CBF demonstrated superior navigation and obstacle avoidance capabilities in complex and unknown environments. For the real-world experiments, due to limitations inherent in the site and robot constraints, in this study only a simplified intersection test was conducted. The results clearly indicated that, compared to traditional CBF methods, SE-CBF exhibited higher safety and stability during obstacle avoidance.

The combined results from both the simulations and real-world tests show that SE-CBF theoretically expands the boundaries of Control Barrier Function applications and provides a practical and effective new solution for mobile robot navigation in complex environments. This method greatly enhances obstacle avoidance and path planning efficiency during challenging tasks, meeting diverse operational needs. These experiments successfully validated the feasibility and effectiveness of SE-CBF. The following section will detail introduce the SE-CBF technique.

# **Chapter 2**

## **Design of a flexible obstacle avoidance Technique**

### **2.1 Control problem formalization**

Control Barrier Function (CBF) is a mathematical tool used to ensure the safe operation of dynamic systems in complex environments. The core idea is to construct a mathematical constraint that keeps the system state within a predefined safe distance. By applying this fixed constraint, systems can effectively avoid getting too close to each other, preventing collisions. In specific applications, CBF monitors the distance between the system and obstacles or other agents. If the safe distance is about to be breached, CBF dynamically adjusts the control input and causes the system to slow down or alter its path, thereby mitigating collision risks. This approach gives the function the characteristic of maintaining a safety set, but in essence, it dynamically adapts to maintain the safety distance

In mobile robots and multi-agent systems, navigation and obstacle avoidance often need to operate in complex and unpredictable environments. Traditional path planning algorithms like A\* (A-Star Algorithm) or Dijkstra perform well in terms of path optimality, but they have significant limitations when dealing with dynamic obstacles[24]. These methods primarily rely on the assumption of a static environment and lack the ability to respond quickly to real-time changes. These algorithms struggle to meet strict real-time requirements for dynamic obstacle avoidance, making systems more susceptible to interference in complex environments.

Control Barrier Functions fill in the gaps left by traditional path-planning algorithms in obstacle avoidance. Once a navigation path is generated, CBF dynamically adjusts the control input of the robot or agent during movement. To ensure that the agent consistently satisfies safety constraints and avoids collisions with dynamic obstacles or other agents. This dynamic response mechanism allows robots to quickly adjust their trajectory when encountering sudden obstacles or path interference[4].

CBF is primarily applied in areas such as dynamic obstacle avoidance. When robots or multi-agent systems approach obstacles or other moving agents, CBF adjusts the control input in real-time and makes robots slow down or change paths automatically, effectively avoiding collision

risks[13]. CBF excels in region maintenance. By way of imposing constraints on the system state, it ensures that robots or vehicles remain within a predefined safe distance and area. For instance, in autonomous navigation systems, CBF ensures that vehicles strictly adhere to lane constraints and reduces the risk of deviation due to sudden disturbances or operational errors. Lowering the likelihood of traffic accidents. More importantly, CBF significantly enhances system robustness. In cases of actuator failure or drastic environmental changes, CBF can dynamically adjust control strategies to keep the system in a safe state. Loss of control or entry into dangerous areas are prevented and ensured smooth task execution.

### 2.1.1 Overview of Control Barrier Functions

Consider a control-affine system of the form:

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

where  $x \in R^n$  is the state,  $u \in U \subseteq R^m$  is the control input, and  $x \in D \subseteq R^n$  is the set of admissible inputs. The functions  $f: R^n \rightarrow R^n$  and  $g: R^n \rightarrow R^{n \times m}$  are assumed to be locally Lipschitz continuous. A set  $S \subseteq R^n$  is defined as forward controlled invariant with respect to the system (1) if, for every  $x_0 \in S$ , there exists a control signal  $u(t)$  such that  $x(t; t_0, x_0) \in S$  for all  $t \geq t_0$ , where  $x(t; t_0, x_0)$  represents the solution of system (1) at time  $t$  with the initial condition  $x_0 \in R^n$  at time  $t_0$ .

Consider control system (1) and a safe set defined by:

$$C = \{x \in R^n : h(x) \geq 0\}, \quad (2)$$

for a continuously differentiable function  $h: R^n \rightarrow R$  that has a relative degree one. The function  $h(x)$  is called a (zeroing) CBF if there exists a constant  $\gamma > 0$  such that:

$$\sup_{u \in U} [L_f h(x) + L_g h(x)u + \gamma h(x)] \geq 0, \quad (3)$$

Where  $L_f h(x) = \frac{\partial h}{\partial x} f(x)$  and  $L_g h(x) = \frac{\partial h}{\partial x} g(x)$  are the Lie derivatives of  $h(x)$  along the vector fields  $f(x)$  and  $g(x)$ , respectively, Given a CBF  $h(x)$ , the set of all control values that satisfy (3) for all  $x \in R^n$  is defined as:

$$K_{bf}(x) = \{u \in U : L_f h(x) + L_g h(x)u + \gamma h(x) \geq 0\}, \quad (4)$$

It was showed in [25] that any Lipschitz continuous technique  $u(x) \in K_{bf}(x)$  for ever  $x \in R^n$  will guarantee the forward invariance of  $\mathcal{C}$ . The demonstrably secure control law is derived by resolving an online quadratic programming (QP) problem that incorporates the control barrier condition as a constraint.

### 2.1.2 Fixed Safety Distances Lead to Inefficient Space Utilization

Traditional CBF enforces safety by constraining the minimum allowable distance between agents or between an agent and an obstacle. This constraint creates the impression of a virtual “safety” around each agent.

While this method ensures collision avoidance, it can lead to overly conservative behaviour, as agents are forced to maintain fixed minimum distances regardless of the actual size or shape of the obstacles they encounter. For instance, the robot is still required to keep the same minimum distance from the obstacle is small or irregularly shaped. As a result, the available space is not used to its fullest extent. This behaviour has been proven to guarantee collision-free navigation. This rigidity becomes particularly problematic in densely populated environments because it leads to unnecessary detours or even blockage of the path. Inefficiencies consume space and lower system efficiency and throughput. Mathematically, the fixed safety distance can be expressed as:

$$h(p_i, p_j) = \| p_i - p_j \|^2 - d_s^2, \quad (5)$$

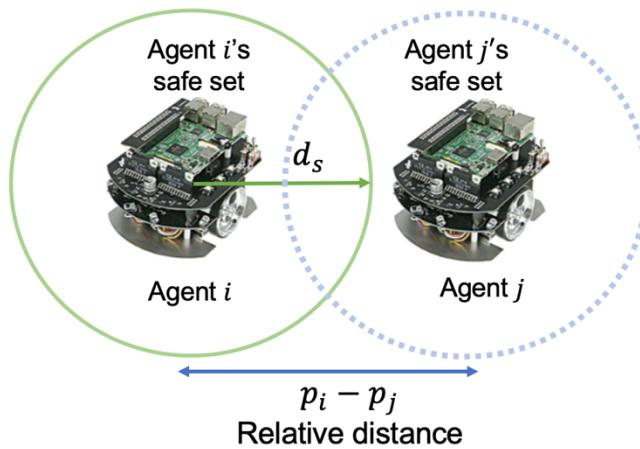


Figure 1: CBF safety distance demonstration

where  $d_s$  represents the constant safety distance. When robot  $p_i$  approaches an obstacle or another robot  $p_j$ . Whatever the size and shape of the obstacle, this constraint involves a minimum safety distance which will always be maintained. This will lead to unnecessary avoidance of obstacles and inefficient utilisation of the available free space.

### 2.1.3 Inability to Adapt to Complex or Irregular Obstacles

Traditional CBFs have difficulties dealing with irregular, dynamically changing obstacles. Safety sets for fixed pattern design are less effective in dealing with obstacle-laden terrain with irregular shielding shapes. The inability to reshape the safety set in real-time often forces robots to make frequent path adjustments. In this case, it might get stuck in local minima or avoid certain places altogether, which would make navigating less efficient. For instance, in environments with multiple irregular obstacles, the standard CBF formulation:

$$h(p_i) = \min_j(|p_i - p_j|^2 - d_s^2), \quad (6)$$

focuses only on the nearest obstacle without accounting for its shape. As a result, robots may unnecessarily avoid narrow passages or take longer detours around obstacles, limiting their ability to operate efficiently in complex environments like Figure 2 right side grey area.

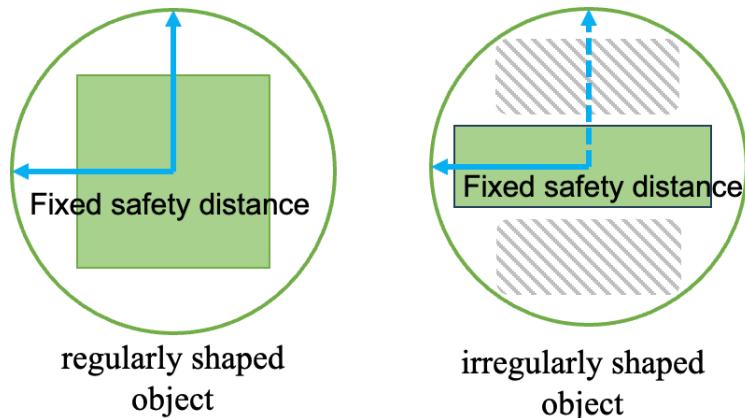


Figure 2: Fixed safety distance limitations of CBF

In practice, the effectiveness of CBF can be severely compromised by the failure of actuators or uncertainty regarding control input. Basic CBF methodologies presume there is perfect actuator dependability and the control inputs  $u$  can be perfectly commanded. In reality, however, there can

be actuation failures or faults leading robots to follow uncoordinated trajectories from their intended paths. This can be modelled by:

$$u_i = \theta_i \hat{u}_i, \quad (7)$$

where  $\theta_i \in [\alpha_i, 1]$  represents the actuator fault. When  $\theta_i = 0$ , the system becomes uncontrollable. Traditional CBFs often neglect this uncertainty. Sometimes, potential collisions or system failures occur when actuator performance is compromised.

## 2.2 A flexible Technique Design

In this section, a Super-Ellipse Control Barrier Function (SE-CBF) is proposed to address these challenges. The innovation improves the effectiveness and the range of adaptability of collision avoidance by optimally customising the boundary of safety around the potential collision regions or the other agents in the environment. The SE-CBF introduces a versatile mathematical model capable of representing a wide range of shapes, from circles to rectangles, depending on the requirements of the scenario. Through the utilization of the Super-Ellipse formulation, the safety barrier may be adjusted based on direction, allowing different safety distances in different conditions. This type of flexibility allows for the construction of more natural and efficient avoidance behaviour of the technique, creating avoidance paths with less non-productive distance, which increases performance on navigation tasks. The SE-CBF is a novel step towards the automation of safe efficient path planning in moving and complex enclosing environments.

### 2.2.1 Super-Ellipse CBF Formulation

A Super-Ellipse is a generalized form of an ellipse, capable of representing shapes ranging from circles and ellipses to rectangles and rounded squares, depending on the choice of parameters. The mathematical formulation of a Super-Ellipse is expressed as:

$$\left(\frac{x}{\alpha}\right)^n + \left(\frac{y}{\beta}\right)^n = 1, \quad (8)$$

where  $x$  and  $y$  represent the relative position of the agent with respect to the obstacle, and  $\alpha, \beta$  are scaling factors along the  $x$  and  $y$  axes. The parameter  $n$  controls how the border curves. When  $n = 2$ , the boundary forms a standard ellipse, while larger values of  $n$  produce a more rectangular profile, making the boundary more effective for elongated obstacles. This adjustment of  $n$  creates a spectrum of possible shapes and makes the system adapt to varying environmental constraints.

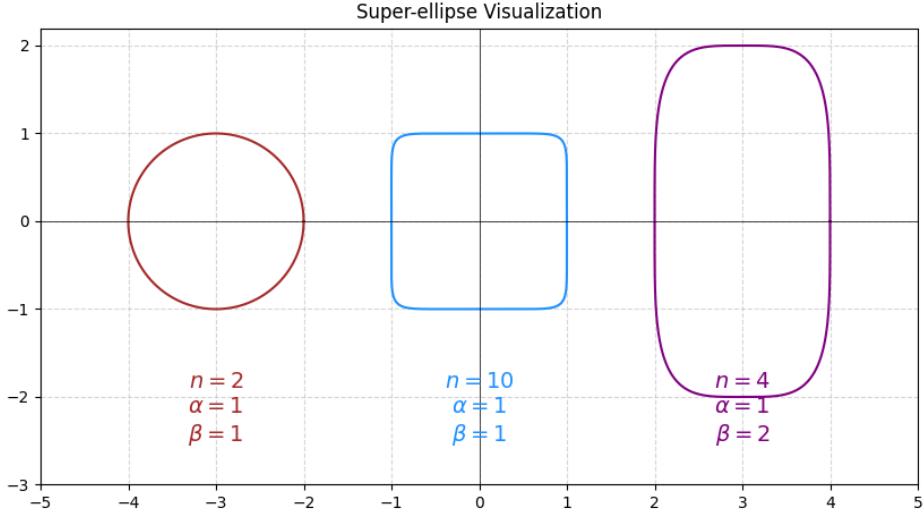


Figure 3: Visualization of formula (8)

To incorporate the Super-Ellipse into the CBF formulation for obstacle avoidance, the robot's position  $(Px, Py)$  relative to the centre of the obstacle  $(Ox, Oy)$  is considered. The resulting CBF is defined as:

$$h(x) = 1 - \left( \frac{(Px - Ox)}{\alpha} \right)^n - \left( \frac{(Py - Oy)}{\beta} \right)^n, \quad (9)$$

This formulation a safety boundary around the obstacle. The function  $h(x) \geq 0$  represents the safe set, where the robot remains at a non-colliding distance, while  $h(x) < 0$  signals a potential breach of the safety boundary. As the robot approaches obstacles, this restriction dynamically reshapes the border to meet its size and shape.

A key advantage of the Super-Ellipse lies in its capacity to introduce anisotropic scaling through the parameters  $\alpha$  and  $\beta$ . These scaling factors enable the safe set to stretch or contract independently along each axis, allowing the robot to maintain a larger safety distance in one direction. For example, when navigating narrow corridors,  $\alpha$  can be reduced and make closer

proximity along the horizontal axis. While  $\beta$  remains larger to preserve a safe buffer. Directional flexibility can help the robot and system navigate tight or congested spaces safely.

Furthermore, the curvature parameter  $n$  can be set dynamically in relation to the contour of the obstacle. For stretched obstacles, an increase of  $n$  is a yield boundary that approximates the rectangular shape. The approach envelope is more closely fitted to the shape of the obstacle. Conversely, smaller values of  $n$  produce smoother boundaries or ovals, which are more appropriate for circular or arbitrarily shaped obstacles instead. This adaptive approach avoids over-conservatism and improves space utilisation and path planning efficiency.

### 2.2.2 Dynamic Safety Distance: Formulation and Extension

The dynamic safety distance mechanism of SE-CBF enables the robotic system to flexibly avoid collisions. Unlike traditional CBF, which only applies fixed safety distance constraints, dynamic safety distance adapts to the shape, positioning, and size of obstacles in real-time. This flexibility is achieved through the combination of geometric transformations, rotation and scaling into the safety set. The dynamic safety distance creates a safety boundary that adapts to the demands of the task. This ensures that the robot can use the space saved in unimportant directions to avoid collisions.

To formalize this approach, the connection of the robot with the obstacle is defined by means of rotation and scaling matrices. The rotation matrix  $R(\theta)$  governs the orientation of the obstacle. This matrix is defined as:

$$R(\theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix}, \quad (10)$$

where  $\theta$  represents the obstacle's rotation angle relative to the global reference frame. By applying  $R(\theta)$  to the obstacle's bounding shape, the safety set dynamically aligns with the obstacle's orientation.

The dynamic safety distance is then expressed by the function:

$$h(x) = |p - R \cdot O|^2 - |R \cdot B|^2, \quad (11)$$

where  $p$  is the robot's position,  $O$  denotes the obstacle's center, and  $B$  represents a boundary point on the obstacle's surface. If  $h(x) \geq 0$ , the robot is within a safe region. Conversely, if the robot were to encroach upon the obstacle's safety margin, it would be considered an infraction.

Dynamic safety distances use rotational and scaling transformations to change the obstacle's safety border in real-time. Scaling ensures that the safety border matches the obstacle's size.

This is achieved using the scaling matrix  $S$  defined as:

$$S = \begin{bmatrix} a & 0 \\ 0 & b \end{bmatrix}, \quad (12)$$

where  $a$  and  $b$  are the obstacle's dimensions along the x and y axes. Rotation is applied using the matrix  $R(\theta)(10)$ , also the obstacle's boundary point  $B$  is calculated as:

$$B = R(\theta) \cdot S \cdot B_0, \quad (13)$$

where  $B_0$  represents the untransformed boundary point. Anisotropic scaling allows the safety set to stretch or contract independently along each axis. For example, in narrow corridors,  $a$  can be reduced to allow closer horizontal proximity, while  $b$  maintains a vertical buffer. This flexibility optimizes space usage while preserving safety.

By continuously updating  $R(\theta)$  and  $S$ , the system dynamically responds to changing environments and makes real-time collision avoidance without recalculating entire paths. This technique improves CBF flexibility and efficient operation in complex, dynamic situations and overcomes fixed-distance model limits.

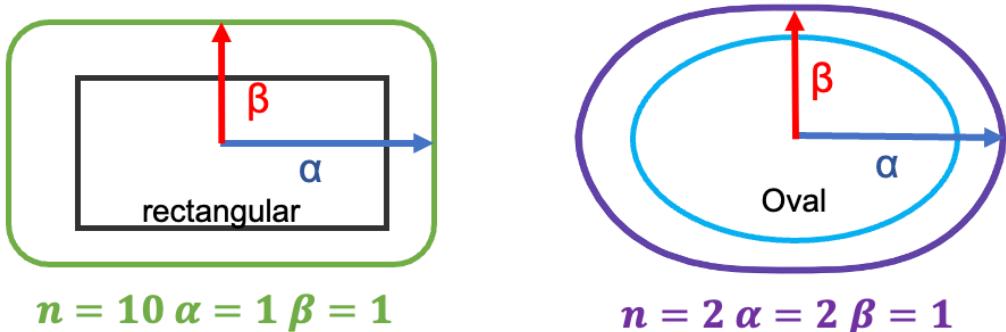


Figure 4: SE-CBF parameters shape

The Figure 4 illustrates different super-ellipse parameters shape the safety boundaries around obstacles. On the left, a rectangular obstacle is enclosed by a rounded rectangular boundary, created

with  $n = 10$ ,  $\alpha = 1$ , and  $\beta = 1$ . The high  $n$ -value sharpens the corners and makes it match the obstacle's shape. The blue and red arrows indicate the primary scaling directions along the x and y axes. On the right, an oval-shaped obstacle is surrounded by an elliptical boundary, using  $n = 2$ ,  $\alpha = 2$ , and  $\beta = 1$ . The lower  $n$ -value smooths the boundary, while the larger  $\alpha$  stretches it horizontally. The robot may move closer along one axis while maintaining vertical safety margins via anisotropic scaling.

These examples illustrate how the transformations of rotation and enlarging create flexible safety contours that support effective avoidance manoeuvres around obstacles of various shapes and positions. This kind of transformation improves spatial utilisation and guarantees effective collision avoidance in a more complicated space.

# Chapter 3

## Simulation Models and Experimental Platform

This section provides an explanation of the structure of the experimental system. The system under consideration is composed of hardware and software components that interact with one another, the robots, and the user.

### 3.1 Software Components

In this section, the software utilized in our system, including the middleware that controls the robots, the simulator for simulation, and the global coordination system, is described.

#### 3.1.1 Robot Operating System (ROS)

In recent years, the Robot Operating System (ROS) has become a significant framework in robotics as it provides a powerful and flexible environment for developing robotic applications of considerable complexity. As a meta-operating system, ROS is aimed at managing and coordinating communication between the hardware abstraction layer and the software integration. For example, mobile robots have a variety of sensors, actuators, and computing devices that have to work in an efficient manner to accomplish complex tasks. ROS serves as middleware that provides relief by operationalizing communication, control, and modularisation, enhancing the simplicity of the latter stage of the development cycle.

The ROS Core, as a framework, promotes the distributed architecture by separating robot software into parts known as nodes. Each node represents an individual process handling specific functions, such as sensor data processing, motor control, or navigation algorithms. Each node serves a specific purpose, and they all communicate by topic. The communication model employed utilizes a publish-subscribe model, which enables scalability, parallelism, and asynchronous message passing. Besides, services allow for synchronous interactions of requests and responses between the nodes. The ROS Master supervises all these activities, organizes the nodes into a

registry that allows the discovery of topics and services and makes sure that data is sent and synchronized between the nodes properly.

A key advantage of ROS is its support for multiple programming languages, primarily Python (rospy) and C++ (roscpp). The former facilitates rapid prototyping and testing and the latter performs efficient and resource-demanding tasks. This multifunctional allows developers to select the language that best suits their application needs.

ROS packages are the fundamental units of the system which framework the code, configuration files along with the libraries into reusable components. This organisation enables developers to reuse the components in a more efficient manner which enhances collaborative practices as software can be exchanged across projects. Package manifests are stored in XML format and contain the management of loose information and dependencies of the package, which enables integration and deployment in a straightforward way. Integration with simulation environments such as Gazebo makes it possible to perform virtual testing with realistic scenarios. This iterative process is crucial for further developing algorithms which need to be integrated with real robots

Additionally, ROS includes RViz which serves as a powerful visualization tool for real-time feedback on sensor data, robot states, and environmental models. Developers can analyze and debug the system using its interactive interface to better understand robot performance in simulation and physical trials.

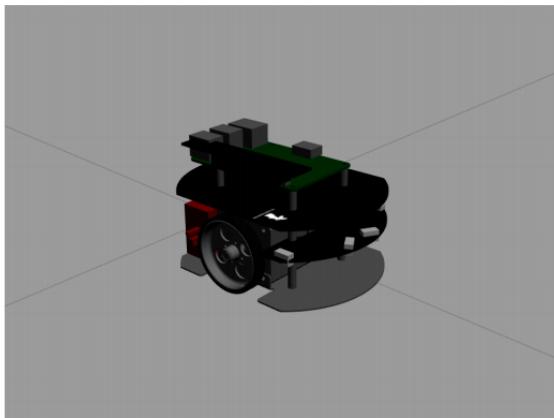
One of the significant advantages of ROS is its support for multiple programming languages, primarily Python (rospy) and C++ (roscpp). This versatility allows developers to choose the language that best suits their application needs and balance between rapid prototyping and computational efficiency. Python's simplicity accelerates development and testing, while C++ delivers the performance needed for resource-intensive applications. The modular construction of ROS coupled with the real-time functioning and high-level navigation and obstacle avoidance features make it one of the best robotics frameworks to use. The diverse libraries, tool sets and active community further enhance the speed of development and optimisation which in return causes rapid changes in robotic innovation and leads to the development of more sophisticated robotic systems.

### **3.1.2 Gazebo Simulation Environment**

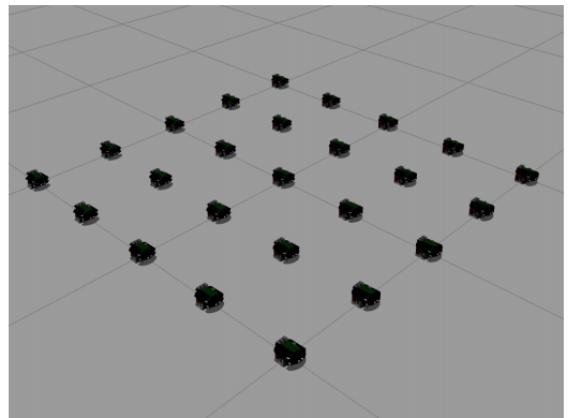
The modular architecture and efficient communication capabilities of ROS allow robotic systems to integrate different functions and pieces of hardware effortlessly. A key simulation tool – Gazebo, further allows for development and testing to be conducted in a more realistic virtual environment, which aids in the validation of algorithms during the simulation stage. This combination allows developers to do step-wise deployment to save on the costs and risks that are associated with hardware testing. In addition, Gazebo is useful in multi-robot system testing because of its ability

to collaborate, which proves important in studying complex systems in a controlled manner. Initially Gazebo developed by Willow Garage, Gazebo has evolved into one of the most essential tools in robotic simulation. It offers ROS developers a robust platform to design, test, and optimize robotic systems within virtual environments. With its user-friendly interface and extensive features, Gazebo is now essential for robotics researchers and engineers.

A major strength of Gazebo is its ability to handle complex robot models. Users are able to build virtual systems by joining joints with geometric shapes so simple mechanisms and even complex multi-degree of freedom robots can be simulated. This encourages rapid prototyping and algorithm testing as the design optimisations can be done in an iterative fashion with no physical hardware working constraints. For example, see Figure 5, which shows how Gazebo supports the simulation of up to 20 robots at once. This paradigm shift saves a lot of time and resources in physical experiments as they are cut down significantly. This ability is essential for large-scale testing and behaviour optimisation of robots in a wide range of controlled environments.



(a) Raspberry Pi Mouse V3 model



(b) Multiple Raspberry Pi robots in Gazebo

Figure 5: Gazebo multi-robots demonstrated

In addition, the Gazebo can digitally reproduce a variety of surroundings including indoor and outdoor settings, contributing to its potential usability. For the programmers, it is possible to create very detailed scenarios which include crossing harsh terrains and other different obstacles with varying levels of light. The use of Object-Oriented Graphics Rendering Engine (OGRE) allows Gazebo to build realistic 3D models and environments which makes it possible to simulate and accurately replicate the environment in a warehouse, city, forest and so on.

The integration of sensor simulation is very important when creating and configuring the robotic system so that a model can be built and put to use. This is where Gazebo excels beyond other simulation solutions because it allows the integration of LiDAR, camera, depth sensor and IMU. This feature supports full-fledged testing of multi-sensor fusion algorithm applications in mobile robotics, self-driven cars, and industrial robotics. From the perspective of the system

deployment, developers and operators can create all-inclusive scenarios that include benchmarks of the system's performance, increasing the trustworthiness of the system prior to deployment.

To correlate these features, it is worth mentioning that Gazebo is tightly integrated with ROS, which allows virtual robots to effectively communicate with ROS nodes. This integration supports developers in the entire simulation of robotic systems, allowing perception, planning, and control to take place in a singular environment. There is also support for regression testing, which guarantees the stability and performance of algorithms in the presence of changes to the environment. For example, developers may record robot locations via odometry topics, which can be utilized for a thorough evaluation of control systems. Communication interference in multi-robot simulation attempts is resolved in Gazebo by assigning every robot a distinct namespace, thus allowing effortless development of large-scale multi-robot systems. These features make Gazebo a preferred choice for scaling experiments to test multi-robot coordination and interaction.

One of the applications of Gazebo is demonstrated in the simulation of multi-robot systems aimed at testing the SE-CBF algorithm for obstacle avoidance in dynamic spaces. Researchers used Raspberry Pi Mouse V3 robots, modelled with the Unified Robot Description Format (URDF), to ensure consistency between simulation and real-world behaviour. These models are available on GitHub. The figure 6 below shows the coordinate transformation tree of a single Raspberry Pi Mouse V3 robot in the Gazebo simulation environment. This coordinate tree describes the relationships between the global pose reference frame (Odom), the robot body (base\_footprint), and the left and right wheels (left\_wheel and right\_wheel). This setup allowed accurate simulation of robot motion and debugging of control algorithms.

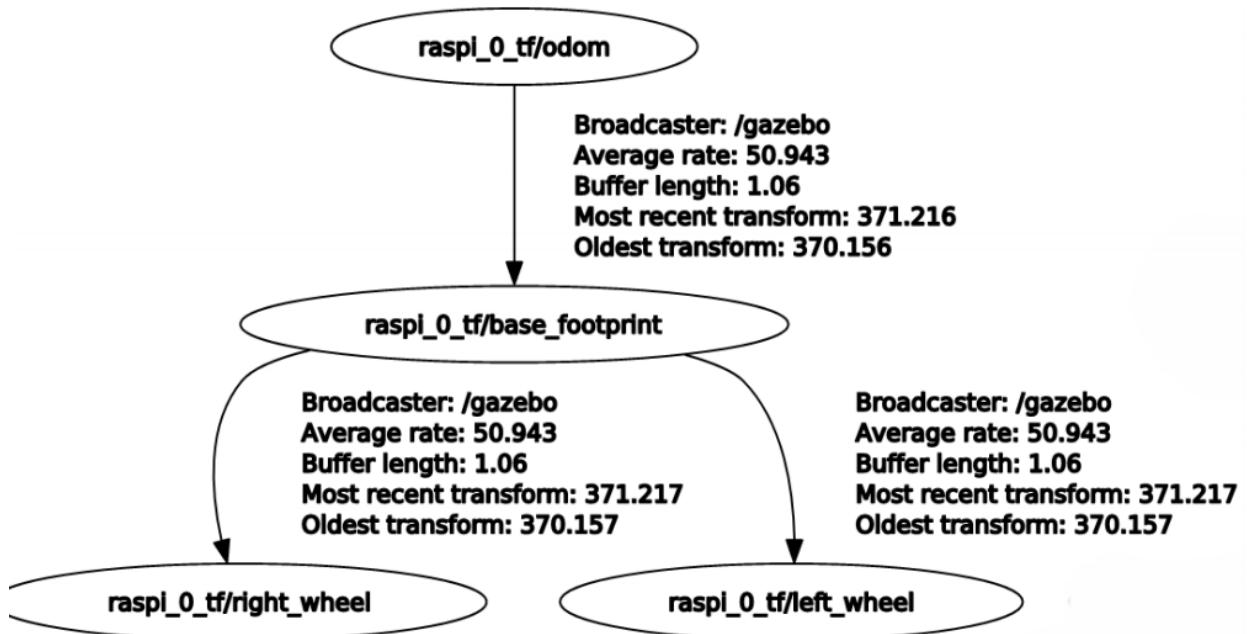


Figure 6: Topic tree of multi-robot simulator

Although the simulation is quite a valid environment for testing, its usage in the real world comes with issues, including odometry drift. The error rates of wheel encoders can over time build up and affect the precision of the position. To resolve this issue, actual systems often have a combination with global positioning system cameras or detached sensors which will be further discussed in the hardware part. These instruments provide high accuracy in tracking the system and its stability in physical spaces.

In conclusion, Gazebo provides an indispensable utility in the realm of robotic engineering due to its advanced hardware modelling capabilities, sensor simulations and the ability to integrate with ROS. From designing intricate robotic constructs to creating challenging real-life scenarios, Gazebo helps researchers in fine-tuning algorithms and verifying designs. It lowers the risk and the expenses associated with hardware testing, thus facilitating the development of target products in an iterative manner, and in so doing, rekindling the spirit of modern robotics.

## 3.2 Hardware Components

This section describes the hardware, including the robot used in the experiment and the camera for the position tracking system, in detail.

### 3.2.1 Raspberry Pi Mouse V3

This experiment utilized the Raspberry Pi Mouse V3 mobile robot, developed by RT Corporation, as a platform for multi-agent coordination and navigation experiments. Compact and flexible, the Raspberry Pi Mouse V3 features a control unit based on Raspberry Pi 3B+, which ensures efficient and stable operation of the robot without any interruptions. Moreover, the 1000mAh LiPo battery charges provide sufficient power for the robot so that it can operate continuously throughout an extended experiment. The sensor configuration of the robot comprises four ultrasonic and basic distance sensors that are designed for detecting obstacles. Sensors of this type are useful for relatively simple environments, but there are limitations in terms of range and accuracy in more sophisticated environments since the sensors only partially cover the circumferential area around the robot. Better detectors such as LiDAR, ultrasonic sensors, and even depth cameras can be seamlessly added to the improvements made to enhance the perception of the environment and assist with obstacle minimization.

The Raspberry Pi Mouse uses four-phase stepper motors that work independently on either side, allowing precise motion control through differential input. As a result of this design, the robot is capable of visually appealing fluid motion, whether it is travelling in a straight line or rotating in place; control and smooth movements are not an issue as the robot is easily able to manoeuvre and move. These experiments can be utilised with multiple robots in the same space to allow testing of

swarm intelligence and collaborative work between the multitude of robots. Furthermore, the compact design supports the deployment of multiple robots within a single experimental setting, facilitating studies in multi-robot collaboration and swarm intelligence. Figure 7 shows the detailed structure and components of the Raspberry Pi Mouse V3.

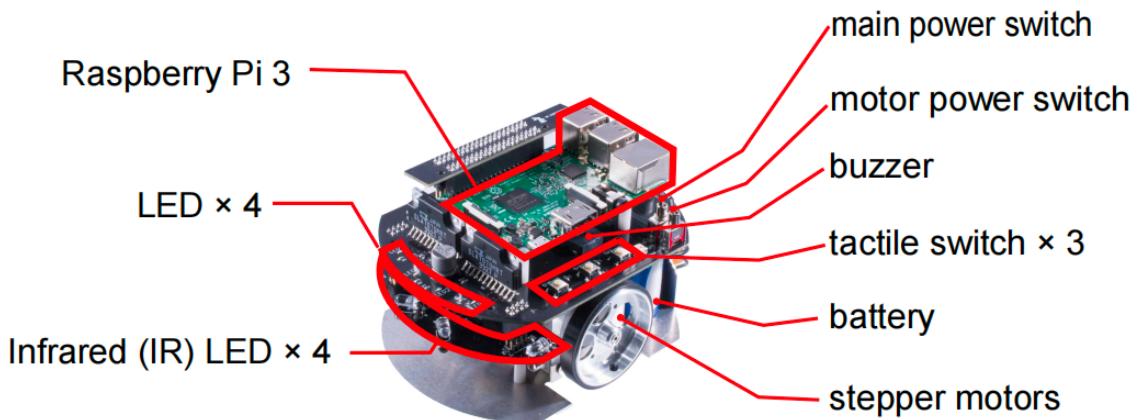


Figure 7: Raspberry Pi Mouse mobile robot

From the software's perspective, the Raspberry Pi Mouse comes with several open-source packages that are hosted on GitHub. This comprises device drivers created for ROS, which ensures that the robot could be used in a ROS system without any hindrances. With these drivers, researchers are able to administer the configuration and control of the Raspberry Pi Mouse within the ROS environment so they could be entitled to utilise already finished tools and libraries in ROS. Also, there exists a large number of example programmes implemented for the Raspberry Pi Mouse. Researchers are, therefore, able to quickly begin using their devices, lowering the needed effort required to develop a robot.

Raspberry Pi Mouse performs well in research and experiments and is widely used for educational purposes. Its intuitive structure and captivating software support make it easy to control mobile robots and use them as a learning tool. This is ideal for students who wish to see for themselves how mobile robots operate and the hardware and software required for its operation. Students and developers are provided with great scope for creativity and development due to the expandability and open-source architecture of the Raspberry Pi Mouse.

### **3.2.2 Intel RealSense D435 with ArTag tracking system**

In this experiment, as mentioned earlier, additional equipment is required to obtain the position information of the robot. We used a tracking system combining the Intel RealSense D435 depth camera with ArTag markers to achieve high-precision positioning and navigation of the mobile

robot. The Intel RealSense D435 is a powerful depth camera with excellent RGB image acquisition capabilities and broad compatibility. The camera is mounted on the ceiling of the experimental site and serves as a top-view camera (see Figure 8), tracking the ArTag marker attached to the top of the robot in real-time.



Figure 8: Intel RealSense D435 Camera

Besides, the Intel RealSense D435 camera is highly integrated with ROS, with its official software package already including functional components that support ArTag marker tracking. Additionally, the accompanying Intel RealSense SDK 2.0 provides comprehensive technical support for developers and offers wrapped interfaces for ROS, OpenCV, Python, and C/C++. While the Intel RealSense D435 excels in depth perception, this experiment primarily leverages its high-resolution RGB camera function to track ArTag markers placed on top of the robots. By utilizing high-precision RGB data streams, the experimental system can achieve stable marker recognition and real-time position monitoring in complex environments.

The D435 camera operates optimally at a working distance between 0.3 meters and 3 meters, which fully meets the experimental requirements for the distance between the camera and the markers. Its wide-angle field of view (FOV) further expands coverage, allowing the camera to comprehensively monitor the experimental site and ensure that multiple robots are effectively tracked simultaneously.

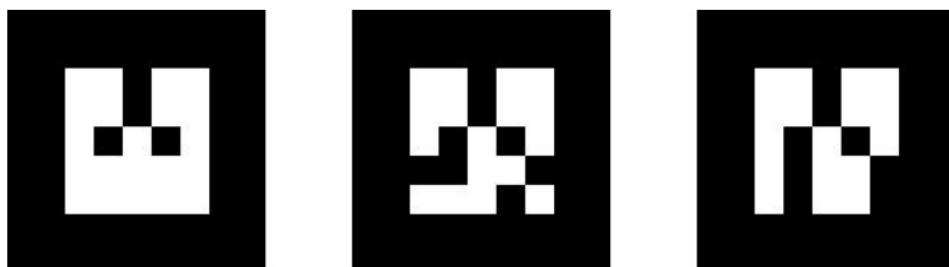


Figure 9: ArTag Markers with Various IDs

The ArTag system, an augmented reality (AR) marker technology, is widely used in virtual reality, robotic positioning, and object tracking. Its core advantage lies in its ability to calculate the

camera's position and orientation relative to physical markers in real-time. The computer system could precisely archive video tracking functionality. Specifically, in this experiment, each mobile robot is equipped with a unique ArTag marker on top (see Figure 9), allowing the overhead camera to independently recognize and track different robots.

The ArTag system employs the open-source ROS package called AR Track Alvar, which is maintained on the basis of the Alvar library. This library allows for the generation of ArTag markers of variable sizes and resolutions and facilitates real-time registration and tracking of either a single marker or a collection of them. Marker bundles were also used to improve the estimated pose accuracy by employing additional markers for a unit pose, thus collapsing those that are non-visible, leading to a robust system. Additionally, AR Track Alvar provided the advantage of easing configuration by estimating the markers' locations from an overhead camera.

In this experiment, the combination of the Intel RealSense D435 camera and the ArTag system presents an effective and inexpensive solution for indoor positioning tasks where GPS is not applicable. Camera-based systems are much more suitable, cheaper and flexible for positioning in a multi-robot system than technologies such as LiDAR, Ultra Wide Band (UWB) or WiFi-based positioning. Integrating the ArTag markers with the Intel RealSense D435 camera provides a way to precisely track the locations and orientations of robots during the experimentation process. Such a configuration allows efficient multi-robot cooperation, obstacle avoidance and other similar activities to be conducted in dynamic and complex settings.

### 3.3 System Architecture

In the real-world experimental section of this study, the robots operate within a  $2\text{m} \times 2\text{m}$  flat area to conduct precise behaviour tests and trajectory monitoring in a controlled environment. To ensure accurate robot position control, we designed a comprehensive tracking and control system to establish a stable feedback loop. It is possible to enable full process monitoring and real-time adjustment of robot movements.

The core control of the system is handled by a personal computer (PC) running Ubuntu 20.04 and ROS Noetic. This specific ROS version was chosen due to its full support for Python 3, ensuring high compatibility with the control algorithms and dependency libraries used in the experiment. This guarantees system stability and development efficiency. On the robot side, each robot is equipped with Ubuntu 18.04 Server Edition and runs ROS Melodic. Although the ROS versions on the robot and the main control PC differ, the system operates and communicates stably through ROS's multi-version compatibility mechanism. The entire system connects the robots and the main control PC to the same local area network (LAN) via a unified WiFi network. The LAN ensures low-latency data transmission and allows the control algorithm to compute robot positions and adjust trajectories in real-time on the PC. The PC can also wirelessly transmit control commands to the robots efficiently.

The camera used in the experiment is the Intel RealSense D435, which has already been highlighted for its impressive performance and sensing capabilities. In the experimental setup, the camera is mounted approximately 2.3 meters above the test area on the ceiling, secured by a universal mount, maintaining a vertical 90-degree downward view See the detail in Figure 10. This mounting configuration was refined through repeated adjustments and tests to ensure the camera fully covers the entire experimental area. By combining the camera with ArTag, the system achieves global monitoring, precise localization, and the clear capture of the position and movement trajectories of all robots during the experiment.

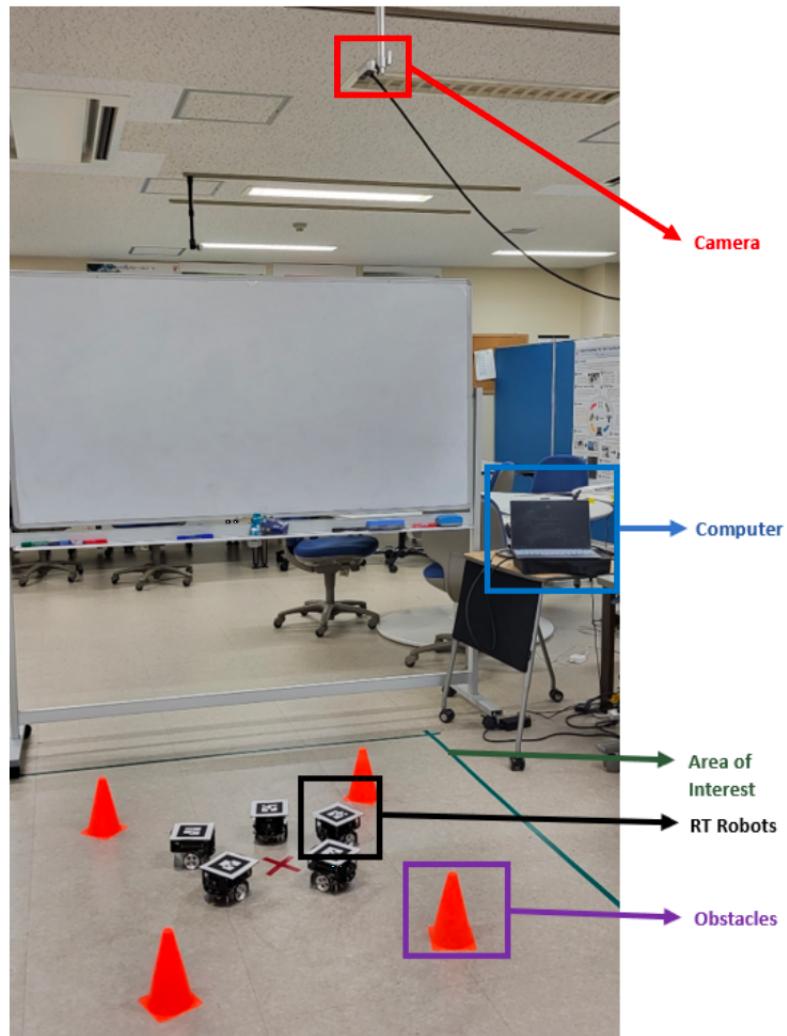


Figure 10: Experimental Setup

To ensure the effective operation of the camera tracking system, the experimental setup includes the installation of the ROS camera driver package and the AR Track Alvar package. These software packages detect and track the ArTag markers placed on top of the robots, providing real-time position feedback. In the experiment, the ArTag mark on the top of each robot is 7 cm in size and has a unique ID to ensure that the camera can accurately distinguish and track different robots.

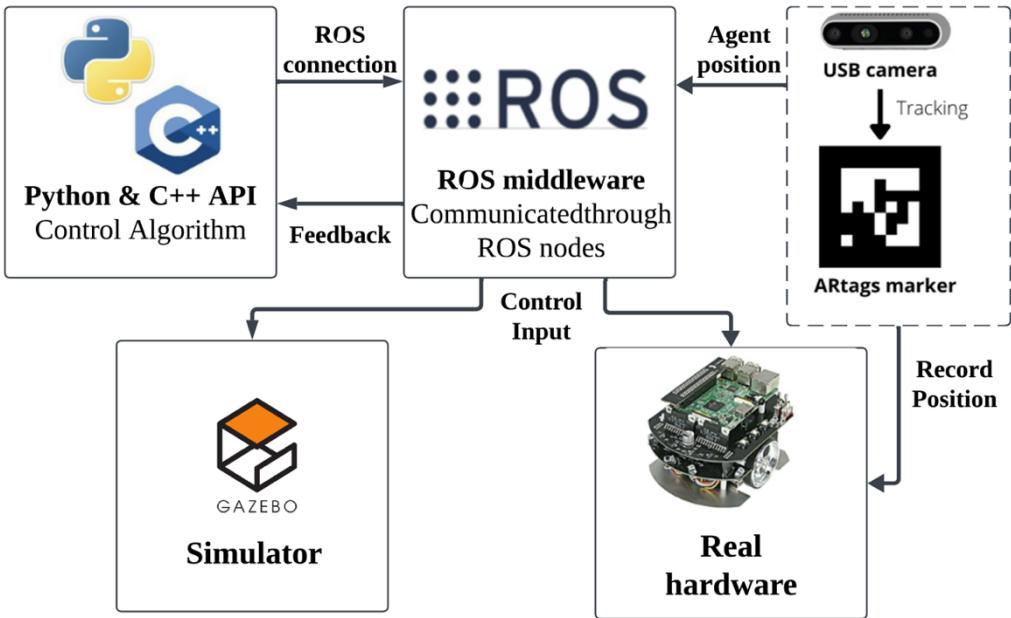


Figure 11: System Architecture

The system architecture, as illustrated in Figure 11, demonstrates the interaction between the camera, markers, and robots, as well as the flow of information throughout the experiment. The control input of each robot was obtained in real-time throughout the experiment, which made it possible to change them and control the robots' migration within the experimental space. An accurate account of the real-time position of each robot is input into the control algorithm and these are remotely delivered to the critical robots for accurate movement. It was especially emphasised that all the robot models should always be within the field of vision of the camera, as this enabled continuous tracking and effective feedback control.

To enhance the experiment's robustness and reliability, real-time data recording and monitoring tools were implemented. These tools facilitated the real-time elimination of inefficient actions as they monitored specific criteria which ensured efficiency in experiments such as position accuracy, movement efficiency and obstacle avoidance performance. The post-examination analysis of experimental data aided in observing control algorithms during task performance in comparison to the desired results, which provided researchers with information on weaknesses of algorithms, thus making it possible to enhance the performance of the system. This integrated mechanism of feedback and data recording constantly provided a reliable way of conducting experiments and formed the basis for refining algorithms and improving the practical side.

# Chapter 4

## Simulation and Experimental Results

This section details the experiments conducted to validate the obstacle avoidance and navigation capabilities of the proposed SE-CBF technique in complex environments. The evaluation was presented with ROS/Gazebo simulations and also physical experiments relating to the evaluation of the technique with regards to path planning and obstacle avoidance in multi-agent systems. During the simulation phase, two scenarios were constructed: a traffic intersection and an unknown area exploration. Following simulation validation, the SE-CBF technique was deployed on the Raspberry Pi Mouse V3 robot for real-world testing. These experiments assessed its ability to handle sensor noise, mechanical inaccuracies, and environmental disturbances. The experiments aimed to determine the effects of sensor noise, mechanical errors, and environmental noise. In the first scenario, experiments focused on obstacle avoidance and autonomous dynamic adaptation. In the second scenario, the focus shifts to testing the stability and effectiveness of the technique performance target validation in more complicated scenarios. The performance of the SE-CBF technique was tested under different conditions with further integration of simulations and experiments into this dual-validation approach with respect to the SE-CBF technique. The results substantiate the technique's mobility and efficiency, confirming its use in complex and dynamic environments.

### 4.1 Simulation Experiment Scenarios

This section presents two scenarios that showcase distinct applications of autonomous systems in complex environments: managing autonomous intersections and exploring unknown areas.

#### 4.1.1 Autonomous intersection

The first intersection experiment compares the obstacle avoidance performance and practicality of the Control Barrier Function (CBF) with the proposed Super-Ellipse Control Barrier Function (SE-CBF). The objective is to test whether SE-CBF can improve the efficiency of obstacle avoidance and path planning in a complex multi-agent system.

In multi-agent intersection situations, MAS faces a critical challenge that balances safety with space utilization. Agents must navigate densely populated areas while avoiding collisions, minimising path lengths, and reducing traffic congestion. Traditional CBF often struggles and has space waste problems in these scenarios. SE-CBF is designed and assumed to solve these problems of CBF. As introduced before, SE-CBF provides greater adaptability with a dynamic safety distance constraints mechanism. SE-CBF is supposed to make the MAS have smoother navigation and better performance in this experiment.

To test this hypothesis, the experiment tests how SE-CBF can improve roundabout junction traffic speeds and reduce congestion to balance safety and performance. The target of SE-CBF is to increase system effectiveness at difficult junctions by enabling agents to move as fluidly as possible while still maintaining proper spatial separation [26].

A typical two-lane roundabout intersection is recreated in a ROS/Gazebo simulation environment to test these capabilities. This setup includes four entry and exit points, each equipped with yield signs and traffic lines to simulate real-world traffic rules, as shown in Figure 12. In the simulation, the center of the roundabout features a green landscaped obstacle area, while the outer edge is marked by a red boundary line to clearly define the roundabout's limits. Each entry lane leading into the roundabout is marked with dashed and solid lines, indicating different priority paths to reflect actual traffic environments as accurately as possible.

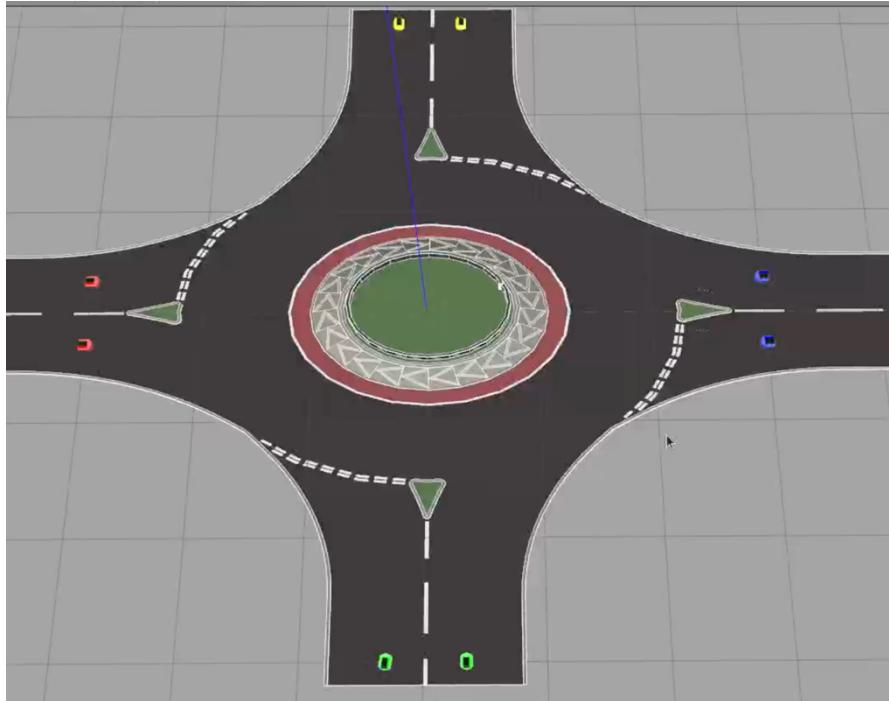


Figure 12: Intersection Simulation with 8 robots

The simulation experiment takes place at a typical four-way circular intersection. During the experiment, eight robots enter the intersection from four directions. The objective of the robots is

to complete a cross-intersection manoeuvre—each robot must traverse the roundabout and exit in the opposite direction. To create a dynamic and complex multi-agent interaction scenario, the robots are organized into groups. Specifically, two same color robots are assigned to the same group and located in the same direction. This setup is designed to simulate a complex multi-agent interaction scenario. During the experiment, the robots autonomously generate their own plans and routes in real-time. The two techniques, CBF and SE-CBF, are required to avoid potential collisions while maintaining smooth movement.

To ensure a fair comparison, the experiment is conducted under identical environmental and initial conditions for both the conventional CBF and SE-CBF techniques. Key performance metrics which will be introduced later—such as obstacle avoidance success rate, path length, and task completion time—are recorded and analyzed. A distinguishing feature is that the type of technique is the only variable being tested. One group of robots operates using the conventional CBF, while the other employs the SE-CBF. Both setups maintain consistent environmental conditions. For both tools, the parameters are also consistent, such as the robot's moving speed, the angular velocity and the minimum safety distance between agents. The comparison focuses on how each technique defines and utilizes these safety distances with the obstacle. The experiment provides a clear and unbiased evaluation of the potential advantages of the SE-CBF over the conventional CBF by isolating the technique as the solitary variable.

CBF use fixed safety distances, which create a circular safety zone around the robot. Though effective, this simplistic model overlooks the fact that space management is rarely optimized at roundabouts or space-constrained corridors. The uniform margin of circumvention may incite poor path planning and, subsequently, traffic flow reduction and holdups due to diversified safety zone overlaps that act as constrictions.

SE-CBF, on the other hand, introduces super-ellipse geometry to dynamically adjust the safety distance and safety set, addressing the limitations of conventional CBF. SE-CBF adapts the safety distance based on environmental density. Specifically, In the longitudinal (travel) direction, SE-CBF applies a longer safety distance, ensuring that the robot can pass quickly through intersections or narrow pathways. In the lateral (avoidance) direction, the safety zone is shortened. This different safety distance can make the safety set perfectly match the middle obstacle shape and allow multiple robots to move closer together for more efficient obstacle avoidance.

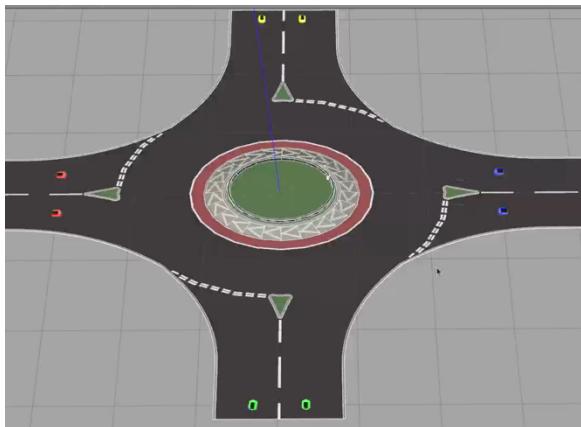
To comprehensively evaluate the performance of both obstacle avoidance methods, the experiment focuses on three key performance indicators[27] [28]:

- **Minimum distance** -This refers to the average minimum distance maintained between the robot and obstacles or other intelligent bodies throughout the entire operation process. It reflects the safety and reliability of each method in complex interactive environments.

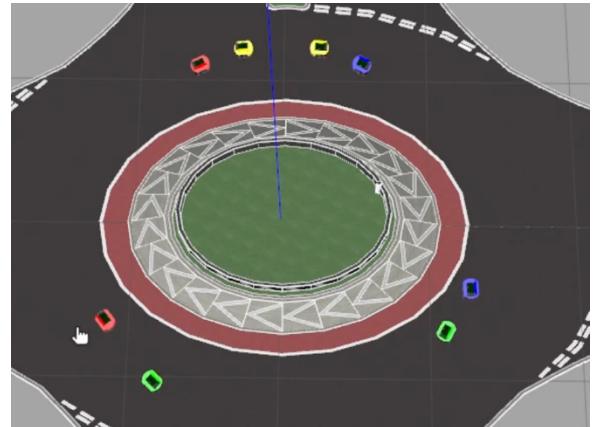
- **Travel Speed** -The average speed of the robot while crossing the intersection, used to assess the smoothness of path planning and identify slowdowns or stalls caused by avoidance maneuvers.
- **Passing Time** -The time taken for each robot to traverse the intersection, comparing the efficiency of conventional CBF and SE-CBF in facilitating intersection crossings and quantifying the practical impact on path planning.

#### 4.1.2 Autonomous intersection results

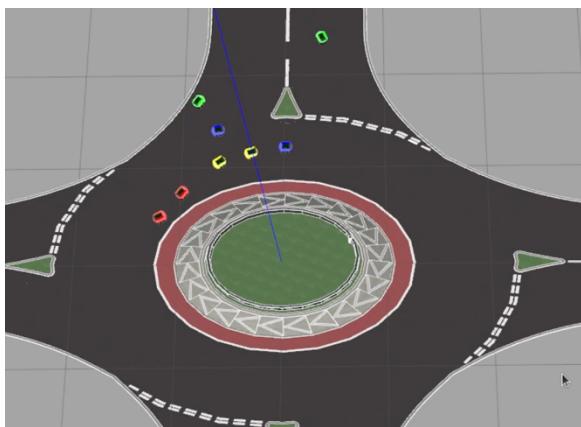
**CBF simulation result:** Figure 13 shows the keyframes for the different stages of the simulation process.



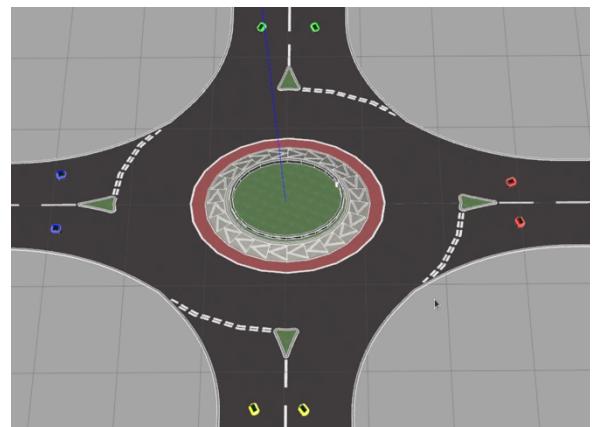
(a)  $T = 0s$



(b)  $T = 40s$



(c)  $T = 70s$



(d)  $T = 110s$

Figure 13: Snapshots for the CBF Intersection under simulation:

(a)  $T = 0s$ , (b)  $T = 40s$ , (c)  $T = 70s$ , (d)  $T = 110s$

Figure 13(a) shows the initial state of the simulation at 0 seconds, displaying the starting positions and distribution of multiple robots near the roundabout intersection. Robots of different colors represent different path-planning tasks. All robots are stationary at their respective entrances waiting for the simulation to start.

Figure 13(b) captures the simulation at 40 seconds. By this time, some robots have begun to enter the roundabout. The traditional CBF control mechanism ensures that the robots maintain a fixed safe distance while driving to avoid potential collisions. Therefore, the robots have to re-plan the path while maintaining a safe distance. It can be seen that some robots have to take a more conservative path or stop and wait for other vehicles to pass due to lack of space. This leads to a decrease in the overall efficiency of the robots.

Figure 13(c) illustrates the state of the simulation at 70 seconds, approximately halfway through the process. It can be observed that the robots dynamically adjust their paths. Among them, the robots with enough space on the outside pass one by one first, leaving enough space for the remaining cars. Although four robots can only move slowly or stop due to mutual influence and the constraints of surrounding obstacles, no collision occurs.

Figure 13(d) shows a snapshot of the simulation at the end of the time. All robots have successfully passed through the intersection, which proves the effectiveness of the traditional CBF method in multi-robot obstacle avoidance tasks. However, the time required for some robots to pass through the intersection has increased significantly, which reflects the limitations of the fixed safety distance method in complex intersection scenarios.

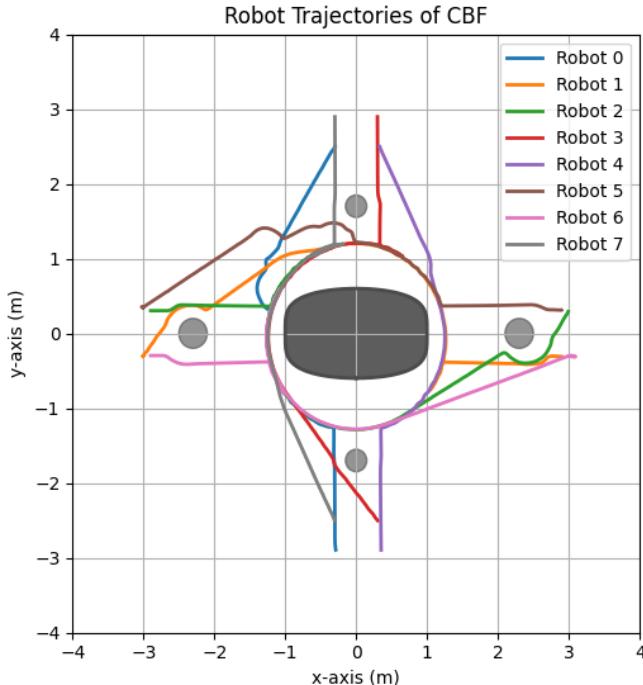


Figure 14: Trajectories path of 8 robots in the CBF intersection simulation

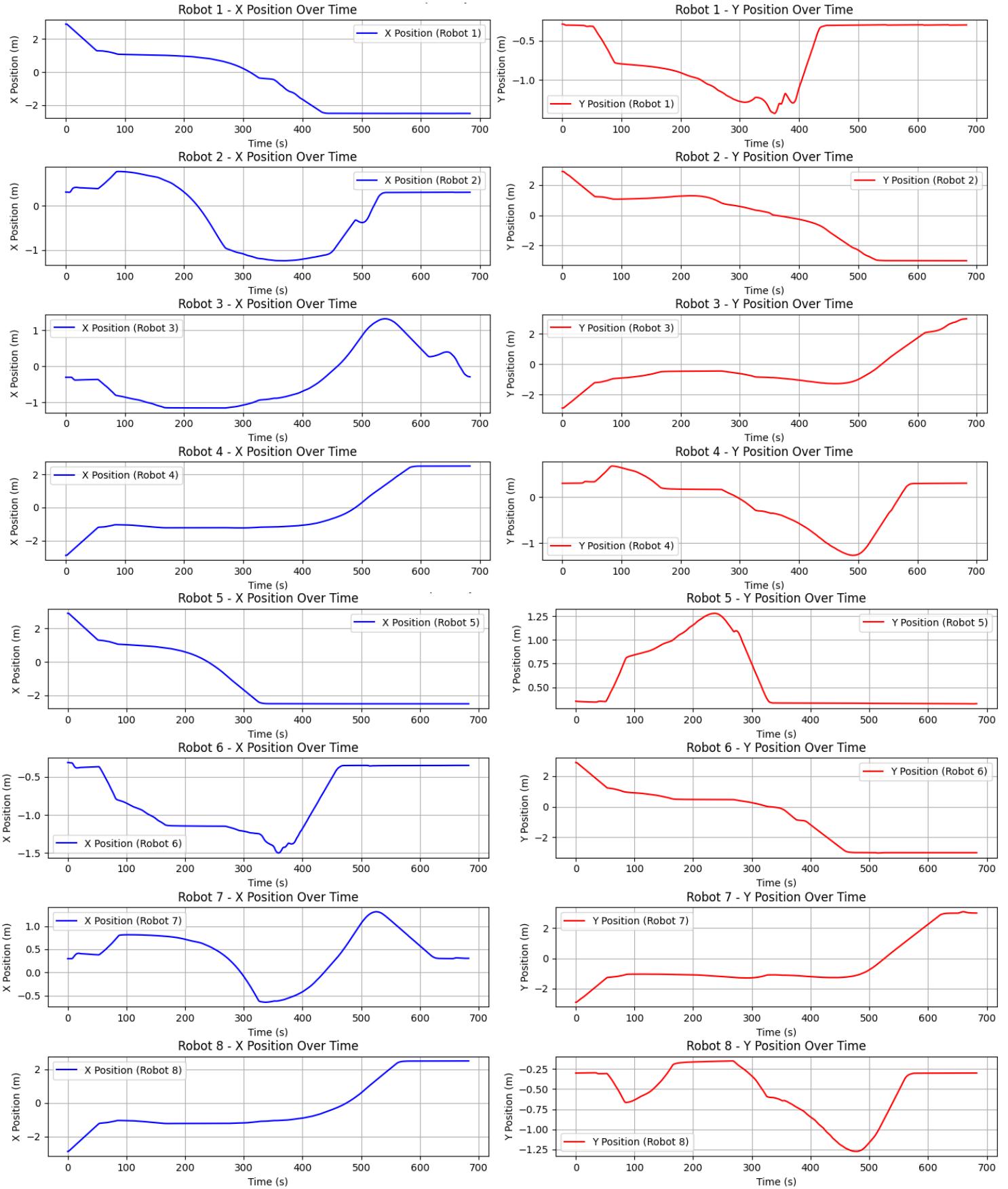


Figure 15: Trajectories of the x and y positions in the CBF intersection simulation

In addition, Figure 14 shows the robot trajectory diagram, which analyzes the movement trajectory of each robot from a bird's-eye view. The trajectory colors correspond to robot IDs, with the center area representing the intersection and the grey circular region indicating static obstacles. It can be observed that the robot path avoids the obstacle, but due to the fixed safety distance constraint, some of the paths deviate significantly from the center, resulting in too much space not being used effectively.

Figure 15 illustrates the X and Y position trajectories of eight robots over time. Each row represents an individual robot, with the left column showing the X position and the right column showing the Y position. The trajectories reveal distinct movement patterns for each robot, highlighting how their positions evolve dynamically in response to the environment. Smooth transitions in some trajectories suggest effective navigation, while abrupt changes in others may indicate obstacle avoidance or interactions with nearby robots. Overall, the data showcases the adherence to safety constraints, as robots maintain safe distances and avoid overlaps, reflecting the control algorithm's performance.

**SE-CBF simulation result:** Figure 16 shows the keyframes for the different stages of the SE-CBF simulation process.

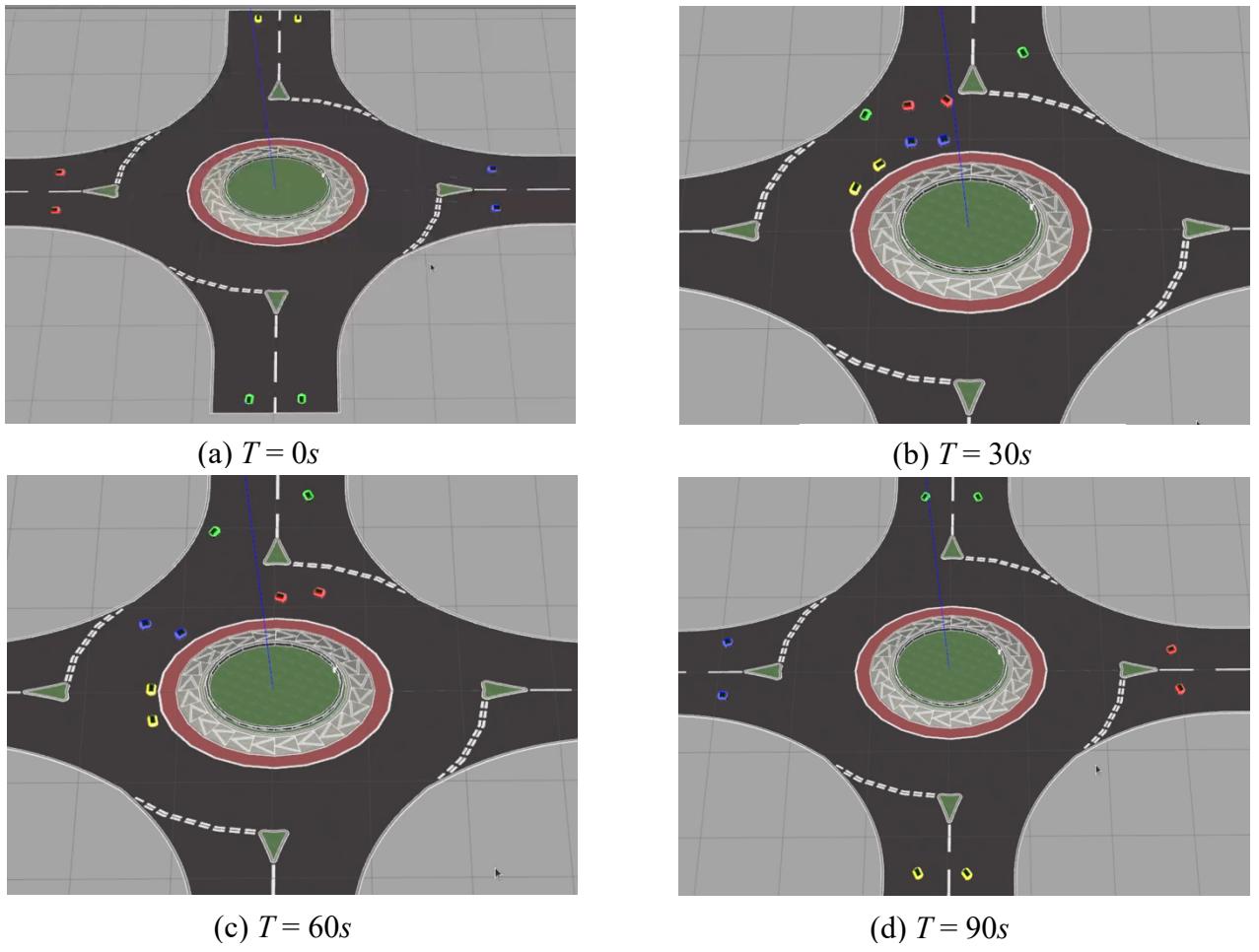


Figure 16: Snapshots for the SE-CBF Intersection under simulation:

(a)  $T = 0s$ , (b)  $T = 30s$ , (c)  $T = 60s$ , (d)  $T = 90s$

Figure 16(a) shows the initial state of the simulation at 0 seconds, which is identical to the start of the CBF simulation. Multiple robots are standing by outside the roundabout. The robot colors correspond to the numbers, and they are distributed at the entrances to the intersection.

Figure 16(b) depicts the state at 30 seconds when the robots have already begun entering the intersection—earlier than in the CBF scenario. Unlike the traditional CBF method, SE-CBF allows robots to maintain a more flexible safety distance. When facing the elliptical greenbelt obstacle in the center, SE-CBF maintains a longer safety distance along the x-axis while reducing it along the y-axis. As seen in the image, this provides the robots with more space to maneuver around each other. Even though robots still need to maintain a minimum distance, no congestion or halting occurs, unlike the potential blockages seen with CBF.

Figure 16(c) shows that some robots have successfully crossed the intersection within 60 seconds, faster than with CBF. The remaining robots have also completed their collision avoidance maneuvers with other agents and are beginning to proceed in pairs toward their target destinations.

Figure 16(d) illustrates the end of the simulation, demonstrating that the SE-CBF method completes the intersection crossing in just 90 seconds, significantly faster than the CBF method. All robots successfully navigate through the intersection, validating SE-CBF’s superiority in complex traffic scenarios.

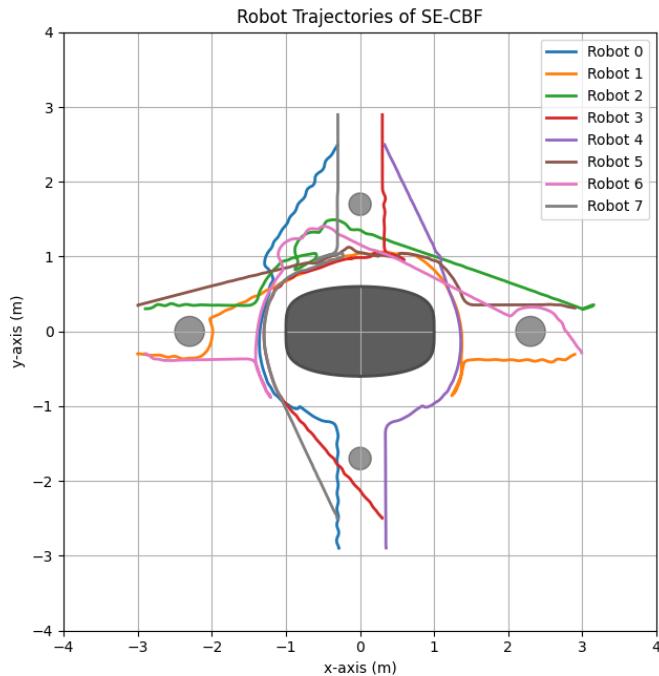


Figure 17: Trajectories path of 8 robots in the SE-CBF intersection simulation

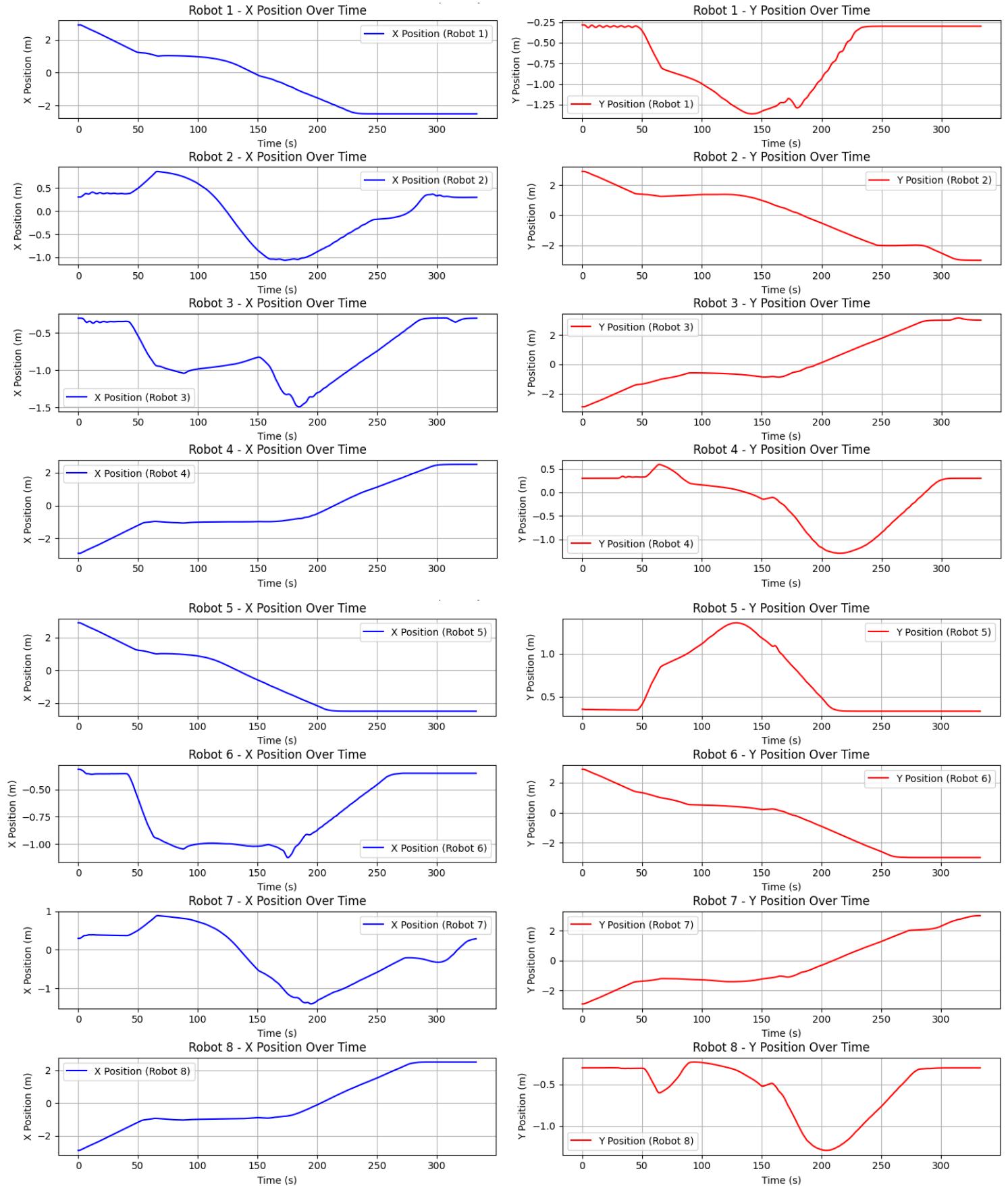


Figure 18: Trajectories of the x and y positions in the SE-CBF intersection simulation

Figure 17 presents the robot trajectories under SE-CBF control. The paths indicate that the robots' trajectories are closer to the center of the intersection, and their routes around static obstacles appear smoother compared to traditional CBF. The dynamic adjustment of safety distances through super-ellipse CBF allows robots to maneuver flexibly based on actual distances, avoiding the path redundancy caused by fixed safety distances. Some robots in the figure can navigate near and closer to the center obstacle. It means that robots would make full use of the intersection space, which improves traffic efficiency. SE-CBF reduces the safety distance along the minor axis, creating more available space for robots to pass quickly through the intersection.

Figure 18 showcases the X and Y position trajectories of eight robots using the SE-CBF. Each robot's movement is visualized with its X position on the left and Y position on the right. The trajectories indicate smoother and more centralized paths compared to traditional methods, reflecting improved flexibility in maintaining safe distances. The super-ellipse function allows for more efficient use of space, with robots exhibiting less deviation from optimal paths, particularly near obstacles. This demonstrates the effectiveness of SE-CBF in enhancing path planning and spatial utilization while maintaining safety constraints.

Overall, both CBF and SE-CBF methods demonstrate high safety levels for obstacle avoidance. The next sections will compare and evaluate CBF and SE-CBF experimental data and findings to investigate their practical performance and optimization impacts in complicated situations.

#### 4.1.3 Autonomous intersection analysis

Following the completion of the experiments, a comprehensive analysis and comparison of the results were conducted based on three core performance indicators: task completion time, path complexity, and obstacle avoidance success rate. To ensure data reliability and minimize the impact of random deviations, each experiment was repeated 15 times for both the CBF and SE-CBF methods.

The collected data was processed and visualized using boxplots, which are a statistical way to summarize and compare the distributions of results across multiple experimental trials. Boxplots can show central tendencies, such as medians, and provide insights into data variability through measures like interquartile ranges and potential outliers. This method of visualization allowed for a clear, straightforward comparison of the two obstacle avoidance techniques, emphasizing their respective strengths and weaknesses.

The specific experimental results are outlined and discussed in the subsequent sections, providing a more granular view of the performance indicators and the implications of the observed differences.

## Average minimum distance

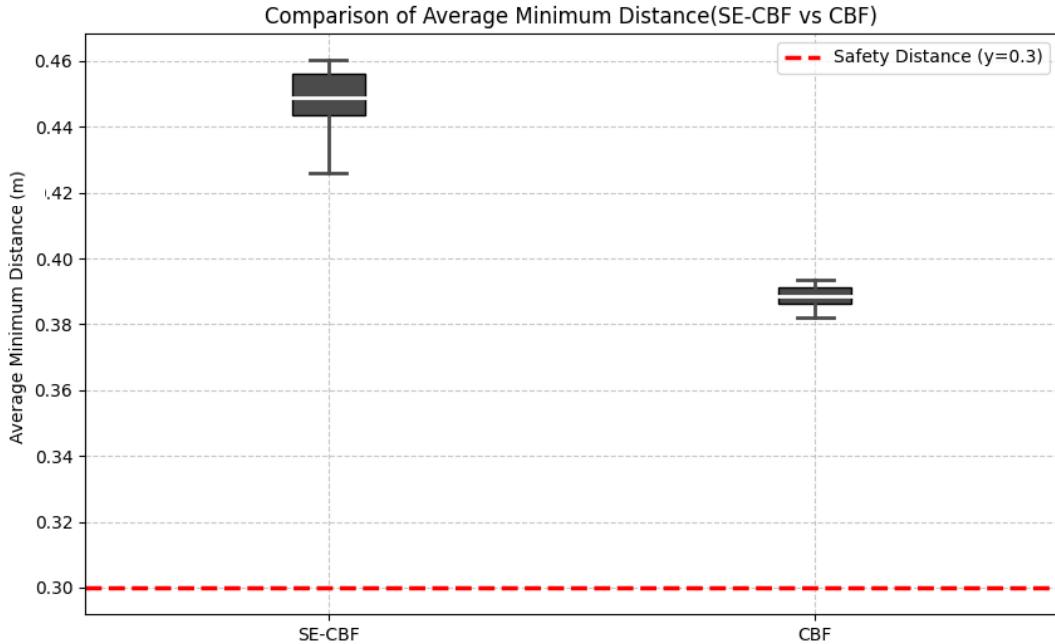


Figure 19: Boxplot of Average Minimum Distance

Figure 19 compares SE-CBF and CBF in terms of the critical metric “Average minimum distance.” The horizontal x-axis represents the two control methods (SE-CBF and CBF), while the vertical y-axis shows the average minimum distance between the robots and obstacles or other robots during the experiment. A red dashed line at  $y = 0.3$  indicates the minimum allowable safety distance threshold during obstacle avoidance, serving as a reference for evaluating the performance of both methods.

“Average minimum distance” is a core indicator of a robot’s obstacle avoidance ability and path planning safety. It refers to the average minimum distance maintained between the robot and obstacles throughout the entire operation process. This indicator of minimum distance reflects the safety space maintained by the robot in movement in an environment where objects with which the robot may potentially collide are in constant motion. The larger the value, the more buffer space the robot leaves during obstacle avoidance and the higher the safety. Compared with only examining the instantaneous minimum distance or the result of a single obstacle avoidance, the average minimum distance can more comprehensively evaluate the stability and obstacle avoidance effect of the robot throughout the entire path planning process, thereby avoiding interference with the results by chance or extreme situations.

This boxplot analysis shows that the plot for SE-CBF is significantly higher than that of CBF, indicating that SE-CBF consistently maintained a larger minimum distance throughout the experiment. The median value for SE-CBF is approximately 0.45, with the interquartile range (IQR) between 0.44 and 0.46. The data is tightly clustered and well above the safety distance threshold.

In contrast, the median value for CBF is around 0.385, with the IQR between 0.38 and 0.39. Both technique methods' data are above the safety threshold, which means that both techniques can keep MAS safe in the complete intersection. SE-CBF demonstrates superior performance across all quartiles and mean values, consistently achieving greater distances from obstacles. The stable and centralized distribution of SE-CBF data and higher average values show higher reliability. In comparison, 15 times all CBF data are lower than SE-CBF data, suggesting a higher collision risk under the same conditions.

The red dashed line ( $y = 0.3$ ) presented in the figure represents the safety threshold necessary for MAS safety running in the intersection. Even though both CBF and SE-CBF methods maintain distance between multi-agents above this safety threshold, SE-CBF keeps greater minimum distances compared to CBF. This indicates that SE-CBF provides better performance by maintaining a safer distance from obstacles and other agents. The result means that the adaptation of SE-CBF helps robots navigate and optimize path planning and collision avoidance. SE-CBF is more effective and trustworthy in avoiding obstacles in restricted and complicated situations.

## Average Speed

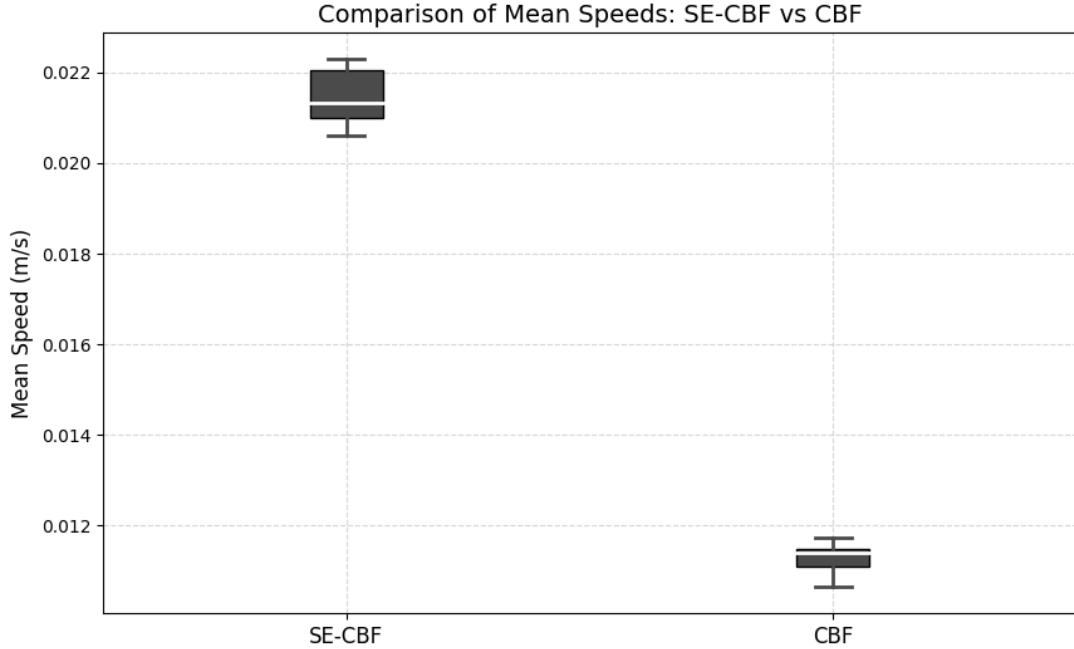


Figure 20: Boxplot of Average Speed

Figure 20 presents a comparison of the "Average speed" between SE-CBF and CBF during the experiment. The x-axis shows two control methods SE-CBF and CBF, while the y-axis shows the average speed in meters per second (m/s). To ensure the reliability of the data and avoid interference from outliers, the boxplot has been filtered to remove outliers.

"Average speed" is a key metric for evaluating the overall movement efficiency of the robots during the experiment. It indicates the relative speed of robots during obstacle avoidance and path-planning tasks. This metric serves as a direct evaluation of task completion efficiency. For example, if the average speed were high, then it can be anticipated that there would be little time lost or spent by unnecessary movements. Path planning would be done more efficiently, and obstacle avoidance measures would be more successful. This comparison metric of average speed accurately depicts the effectiveness of various obstacle-avoiding maneuvers.

From the Figure 20, it shows that SE-CBF plot is on the left, with a median speed of approximately 0.0215 m/s, and IQR is between 0.0205 m/s and 0.022 m/s. The speed is higher and the distribution is concentrated. In contrast, CBF plot is on the right, with a median speed of around 0.0115 m/s, and an IQR between 0.011 m/s and 0.012 m/s. The overall speed for CBF is significantly lower than SE-CBF, and the distribution is similarly concentrated.

The experimental results clearly demonstrate that under the same conditions, the SE-CBF technique has a higher robot movement speed performance. SE-CBF reduces unnecessary delays and detours and maintains safe distances. This advantage is particularly crucial in complex environments or multi-robot systems. This adaptability is particularly beneficial in complex environments or multi-robot systems, where higher speeds directly contribute to shorter task completion times and enhanced overall system efficiency.

In contrast, the CBF technique, which is constrained by fixed safety distances, often forces robots to slow down or take frequent detours during obstacle avoidance. The fixed distances significantly inhibit movement speed and make CBF inefficient in mobile environments. While the CBF technique is conservative and prioritizes safety, it does claim consistent performance despite the constant speed and the lowered average speed of the robot. That highlights once again the minimum performance of CBF.

In general, SE-CBF is more accomplished in ensuring safety and also surpasses the traditional CBF in efficiency during the planning and execution of the path. These boxplot data results further emphasize that there is an SE-CBF advantage regarding speed during each experimental test case. These findings demonstrate how SE-CBF achieves greater robot speed and task completion efficiency while still being safe, making it much more pragmatic and actionable in sophisticated dynamic situations. This provides sufficient proof for the wider use of SE-CBF in the real world, where efficiency and safety are equally critical.

## Completion Time

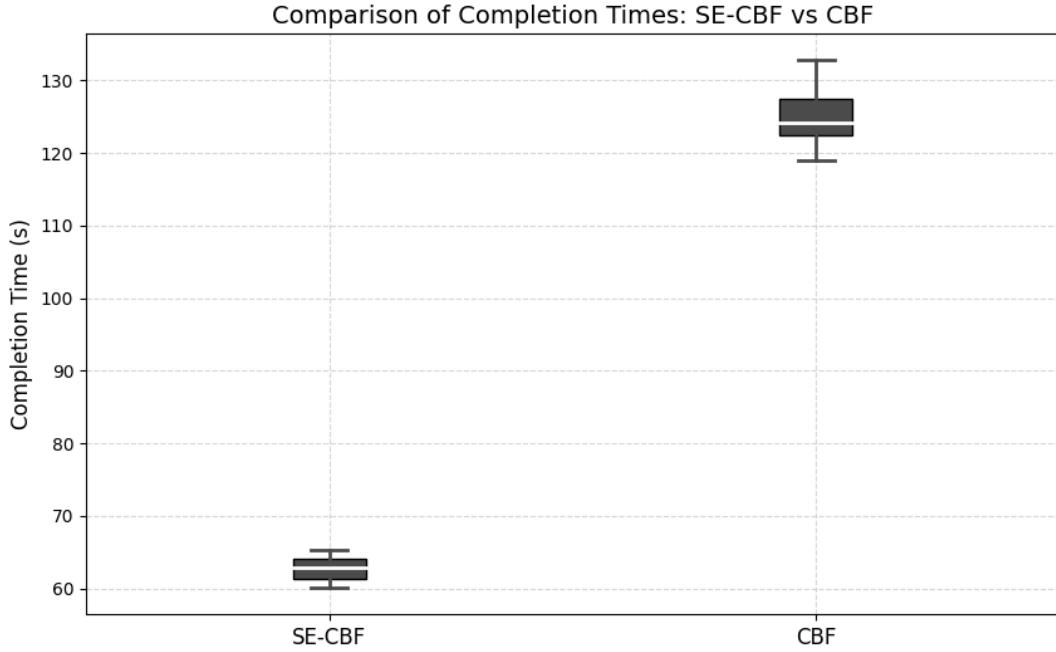


Figure 21: Boxplot of Completion Time

The Figure 21 illustrates the Boxplot comparison of the metric "Completion Time" between SE-CBF and CBF during the experiment. The horizontal x-axis represents the two control methods (SE-CBF and CBF), while the y-axis indicates the time required to complete the task (in seconds).

"Completion Time" is a key indicator for evaluating the execution efficiency of a robotic system. It is measured by the total time required for robots to complete specific tasks. This metric is particularly important as shorter completion times indicate that robots can efficiently plan paths, navigate obstacles, and reach their goals with minimal delays. This is especially important in multi-robot systems and collaborative tasks, where interactions and complex environments increase the complexity.

From the Figure 21, it shows that the SE-CBF plot is positioned on the left, with a median of approximately 62 seconds and IQR is between 60 and 65 seconds. The short range and tightly distributed indicate that the completion time is low and high stability in the experimental results. In contrast, the CBF plot is on the right, with a median of around 123 seconds and an IQR between 120 and 130 seconds. The overall task completion time for CBF is significantly higher than that of SE-CBF, with a wider data distribution and greater variability.

The Boxplot and data show the advantage of SE-CBF in terms of task completion time. SE-CBF provides more flexible path planning and better collision avoidance functions. Robots can move at higher speeds and avoid obstacles at the same time during the task (see Figure 20 and

Figure 21). Furthermore, with the higher speed, the complete tasks time is significantly reduced too. SE-CBF method demonstrates superior performance with shorter, more consistent task completion times. The smaller IQR ranges in the boxplot of SE-CBF indicate higher effectiveness and stability. Conversely, the CBF boxplot shows a wider IQR time distribution, demonstrating greater variability. This variability is a result of the CBF technique's fixed safety constraints mechanism, which frequently forces robots to slow down or take unnecessary detours during obstacle avoidance. This mismatch emphasizes CBF's limitations in this experiment since it shows its incapacity to balance reliability and efficiency.

The experiment results reveal that SE-CBF can reduce task completion time while maintaining robust obstacle avoidance capabilities. This efficiency advantage makes SE-CBF a better solution for operational and reliability techniques for MAS in complex environments. Although the conventional CBF method achieves task completion, its longer and more variable task completion times would make it less desirable for experiments where high efficiency and high stability are of concern.

The comparative analysis of the three major parameters from the experimental results suggests that SE-CBF has a number of benefits over the traditional CBF in a multi-robot interaction case. The hypothesis that SE-CBF makes MAS perform better in traffic intersections would be proved. First, in regard to the average minimum distance, SE-CBF now has better dependability and advanced safety guarantee. Second, in the case of average speed, SE-CBF caused fewer instances of breakdown or stalling when there were obstacles in the way. Last, in terms of task completion time, SE-CBF significantly reduces the time required for robots to traverse intersections. SE-CBF optimally balances safety, efficiency, and fluency for the multi-agent in the complex environments task.

#### **4.1.4 Unknown area exploration**

This is the second simulation experiment of the study, designed to simulate robots performing autonomous exploration and obstacle avoidance tasks in complex environments. Such conditions are attempted to be simulated by providing a set of obstacles that are distributed irregularly over the area, some regions having a high density of obstacles while others being more sparse[29]. These obstacles, which include rectangular prisms, cylinders, cubes, and ellipsoids, are randomly placed throughout the scene. This creates a diverse and complex navigation environment. Such a layout is critical for reflecting the unpredictable and random environments robots often encounter in real-world applications. These and many challenges are fully represented for testing purposes in robotic performance under real-world conditions, including the one presented in Figure 22.

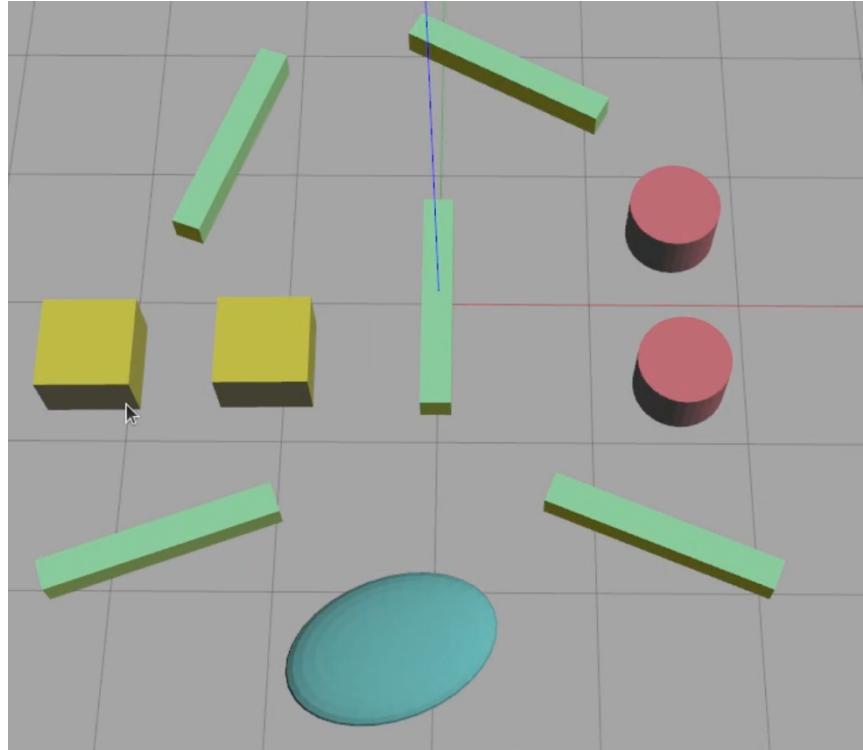


Figure 22: Unknown area exploration

The experiment is designed to evaluate the performance of SE-CBF Functionality and practicality with different numbers of robots in a complex environment. Specifically, the focus is on analyzing how the number of robots impacts key performance metrics, task completion time, path complexity, and obstacle avoidance success rates. The experiment plans to reveal the collaborative multi-agent systems' efficiency in complex environments by increasing the number of robots step by step.

To achieve this, the experiment tests groups of 4, 8, and 12 robots step-by-step. At the beginning of each task, robots are evenly distributed around the scene to ensure uniform starting conditions. Once the task begins, they enter the exploration area from four directions. The goal of the robots is to navigate through dense obstacle zones and reach designated target positions. This configuration enables a stepwise assessment of system performance at different robot densities and the growing complexity of the environment. Throughout the task, robots must keep a safe distance, navigate around obstacles, and gradually complete the exploration of the environment.

The most important point of the experiment is safety. Each robot is required to maintain a fixed safety radius during exploration to prevent path overlap or navigation conflicts. This requirement ensures stable multi-robot operation and mitigates potential collisions, unlike traditional fixed-radius CBF, which often faces limitations in complex dynamic environments, such as encountering

path blockages or restricted exploration in dense areas. The purpose of this experiment is to test whether SE-CBF overcomes these challenges or still faces similar constraints.

Notably, this experiment does not aim to directly compare SE-CBF with conventional fixed-radius CBF. Preliminary tests have shown that s CBF fails to work in this experimental setup, regardless of the number of robots. CBF lacks the capability to manage or adapt to the complexity of dense environments. In this regard, the experiment is shifted to analyze the influence of the robot population in a controlled dynamic environment on the performance of the entire system. In particular, this research seeks to expose the increasing number of robots step by step to demonstrate how the SE-CBF can perform at a high level where multi-agent systems must work rather energetically and cooperatively. This research attempts to show the significance of SE-CBF in addressing the substantial variability in dense areas.

To comprehensively evaluate SE-CBF's obstacle avoidance performance and path planning capabilities in dynamic environments, the experiment is guided by the following key performance metrics[30] [31]:

- **Task Completion Time** – Records the total time required for different numbers of robots to complete scene exploration and coverage tasks. This metric helps analyze the relationship between the number of robots and task efficiency, determining if increasing the number of robots significantly shortens task duration.
- **Path Length and Complexity** – Evaluate the generated path trajectories during exploration, assessing overall path length and complexity. This helps detect redundant detours or overlapping paths, reflecting the efficiency of path planning and multi-robot collaboration.
- **Minimum distance:** This records the minimum distance between the robot and obstacles or other robots in a dynamic obstacle environment as an important indicator of obstacle avoidance performance.

#### 4.1.5 Unknown area exploration Result

**Four Robots Result:** Figure 23 shows the keyframes for the different stages of the 4 robots simulation process.

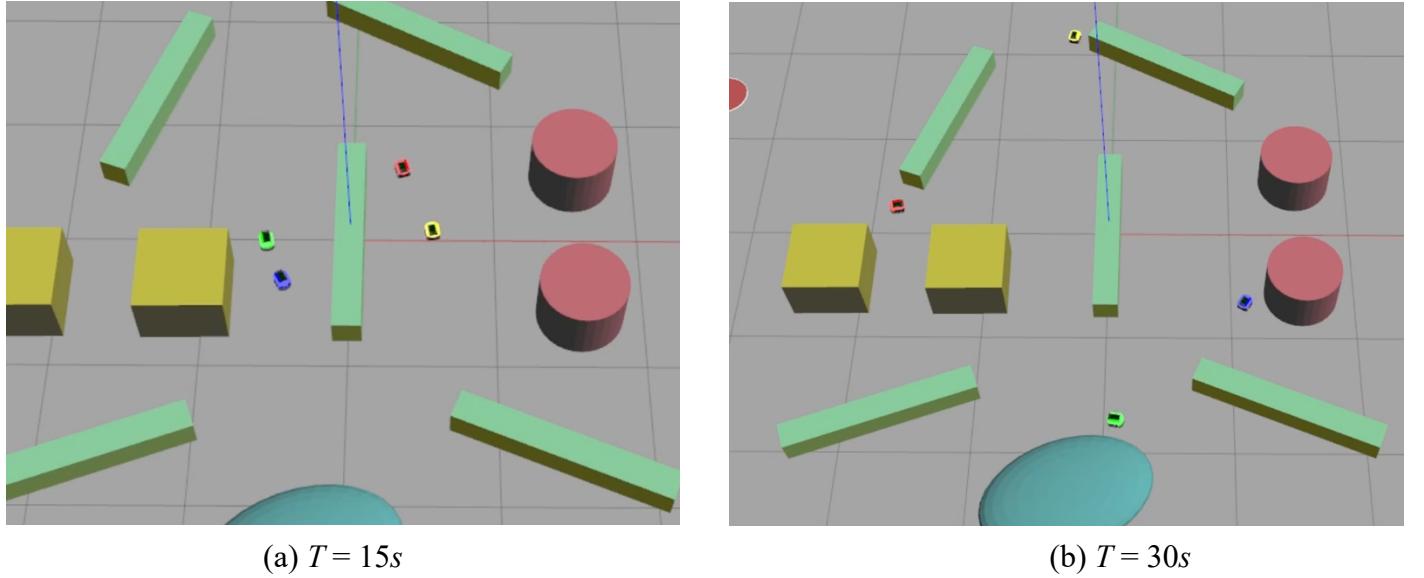


Figure 23: Snapshots for the Four Robots Result:  
 (a)  $T = 15s$ , (b)  $T = 30s$

Figure 23(a) shows the simulation at 15 seconds, where four robots have already entered the complex obstacle environment. Robots of different colors represent different path planning tasks. The robots are evenly distributed and have begun their exploration tasks while actively avoiding obstacles.

Figure 23(b) depicts the simulation at 30 seconds. By this point, the robots have explored most areas and are preparing to exit the obstacle zone. It can be observed that during the obstacle avoidance and path planning processes, the robots dynamically adjust their directions, gradually maneuvering around obstacles located at the center of the scene. The obstacles include rectangular prisms, cylinders, and ellipsoids, densely distributed to increase the complexity of path planning. The robot paths are clearly visible, with some detours reflecting the impact of the complex environment on exploration efficiency.

Figure 24 and Figure 25 depict the trajectories of four robots during exploration in an unknown area. The robots' paths, color-coded by IDs, navigate around obstacles represented by different geometric shapes (green: rectangular prisms, red: cylinders, yellow: cubes, blue: ellipsoids). Figure 24 provides a top-down view, showcasing efficient obstacle avoidance and adaptive navigation, while Figure 25 illustrates the X and Y positions over time, highlighting dynamic adjustments to

the environment. Together, the figures demonstrate the effectiveness of the control algorithms in achieving safe and comprehensive exploration.

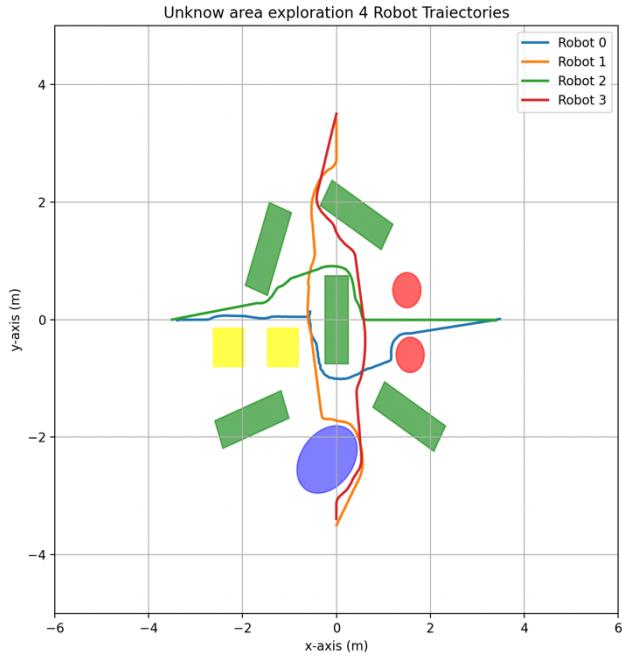


Figure 24: Trajectory map of 4 robots

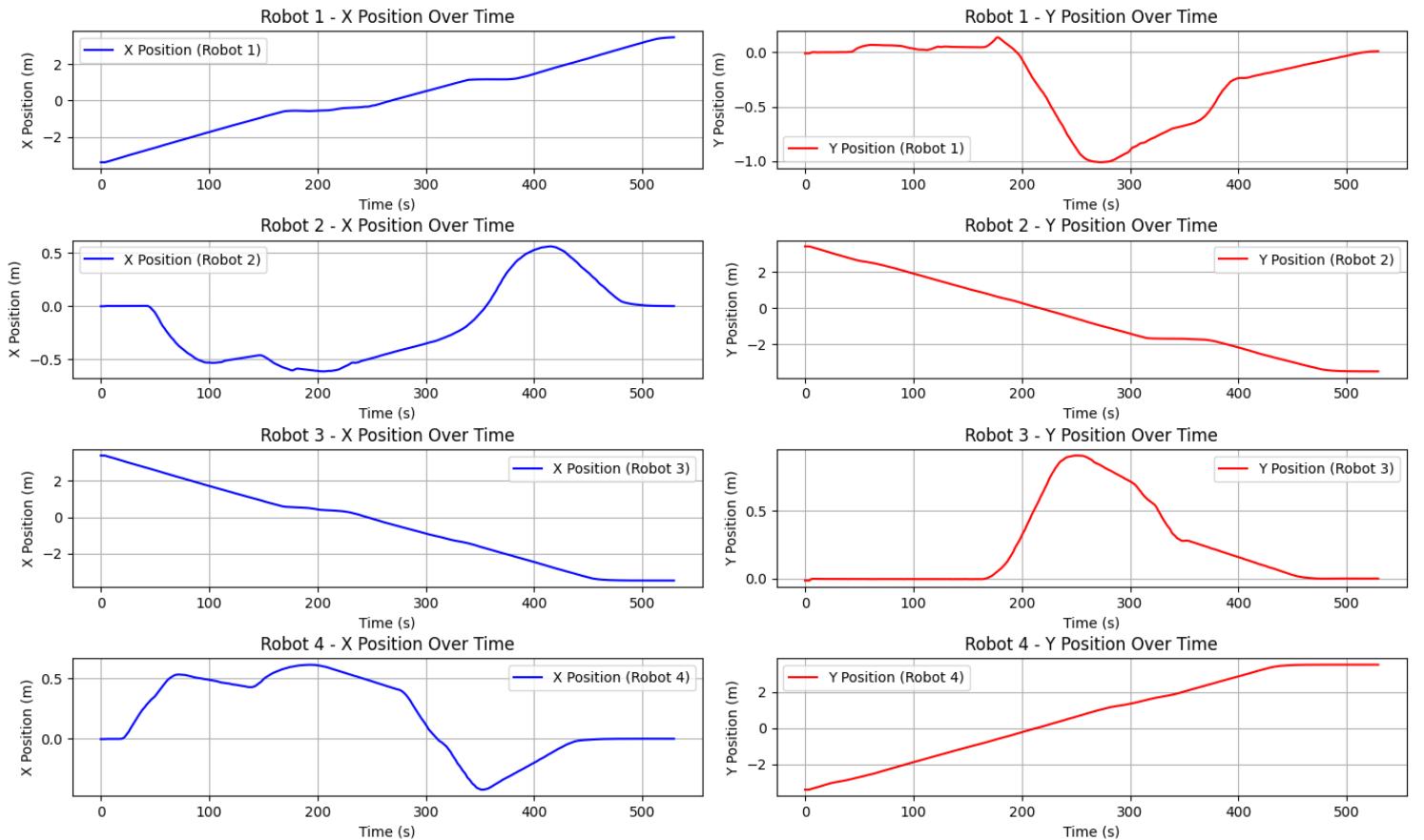


Figure 25: Trajectories of 4 robots in the x and y positions

The robot paths navigate around the obstacles, demonstrating strong obstacle avoidance capabilities. In areas with dense obstacles, significant detours can be observed, reflecting the robots' path-planning strategies in dynamic environments. Due to the preset safety distances, some robots adopt more conservative paths during obstacle avoidance, resulting in longer path lengths. However, no collisions occurred, validating the effectiveness and robustness of the SE-CBF method in complex scenarios. The simulation results indicate that the four robots were able to successfully complete the exploration tasks in complex environments. However, path complexity was high, and some robots took longer to complete their paths due to detours. By increasing the number of robots, the experiment can further investigate the impact of multi-robot collaboration on exploration efficiency.

**Eight Robots Result:** Figure 26 shows the keyframes for the different stages of the 8 robots simulation process.

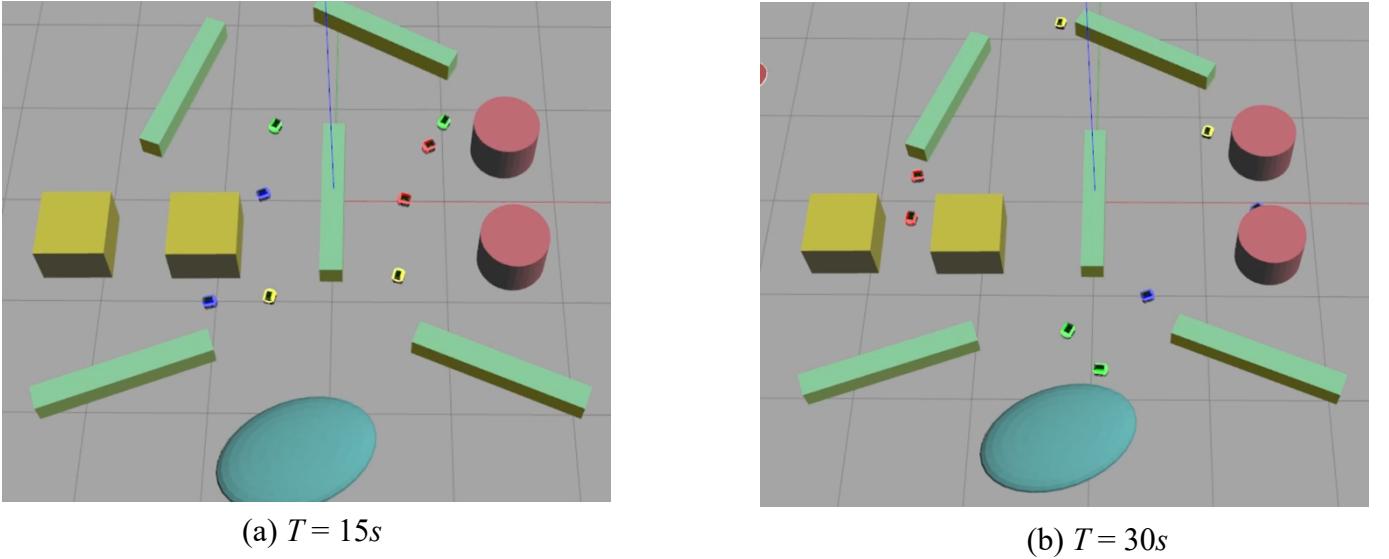


Figure 26: Snapshots for the Eight Robots Result:

(a)  $T = 15s$ , (b)  $T = 30s$

Figure 26(a) shows the distribution of the eight robots entering the complex obstacle environment at 15 seconds in the simulation experiment. Each robot is represented by a different color, reflecting the path-planning tasks they are performing. As the robots gradually move closer to the center of the obstacle area, local areas become congested and the trajectories of the robots begin to intersect with each other. This complex interaction triggers dynamic path re-planning. Robots need to continue exploring the space around the obstacles while avoiding collisions with each other. In addition, it can be observed that some robots try to avoid obstacles by detouring or adjusting their

angles, while others choose to pass through the path-dense area with higher priority, which further verifies the real-time responsiveness of the algorithm in complex environments.

Figure 26(b) shows the situation after 30 seconds when the eight robots have completed their path-planning tasks. It can be seen clearly from the figure that robots of the same color gradually converge and form groups during the exploration process. This means they have effectively covered the dense obstacle area and successfully completed the tasks within the area. After crossing the obstacles, the robots gradually exit the complex environment in different directions and enter the next stage of path tracking. It should be noted that during this process, the robots continue to dynamically adjust their trajectories to avoid possible secondary collisions. The smooth and diverse paths indicate that the system has strong environmental adaptability and robustness. In addition, the robots maintain a reasonable distance apart, which shows that the path-planning algorithm takes safety and space utilization into account while ensuring efficiency.

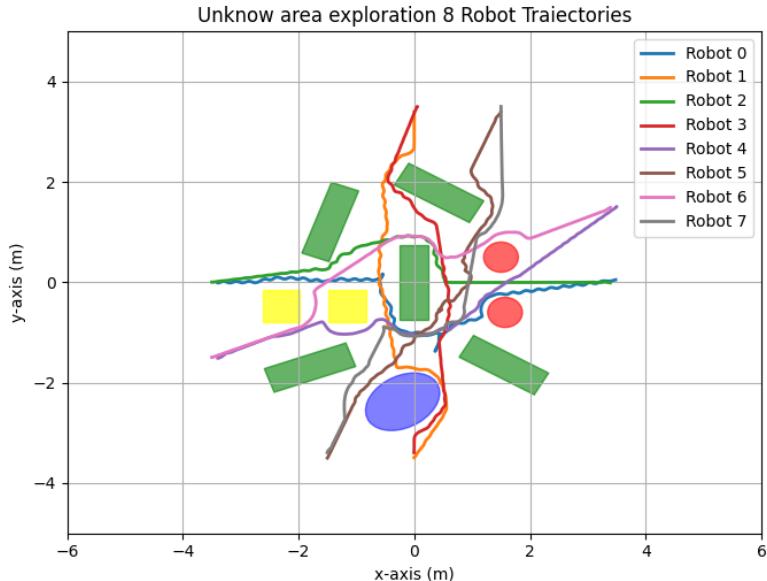
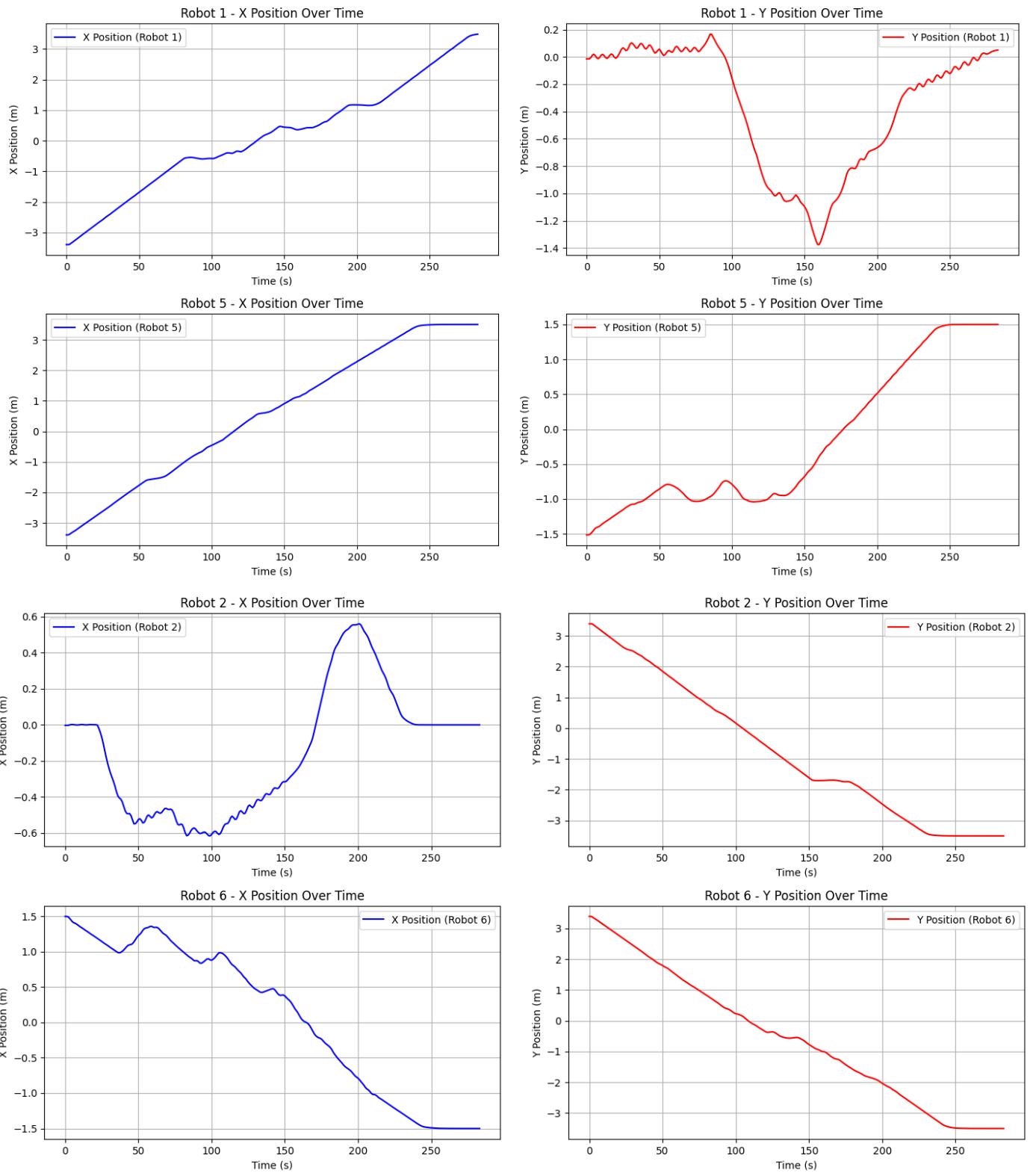


Figure 27: Trajectory map of 8 robots

Figure 27 provides a robot trajectory map, offering a top-down view of the movement paths of all eight robots. The colors of the trajectories correspond to the robots' IDs. The central area of the scene is densely packed with obstacles, and the robot paths navigate around these obstacles. In the trajectory map, it is noticeable that some robots exhibit significant deviations in their paths during obstacle avoidance. This indicates that the robots re-planned their exploration routes in the complex environment. Although these detours increased the overall travel distance, they successfully avoided potential collisions.



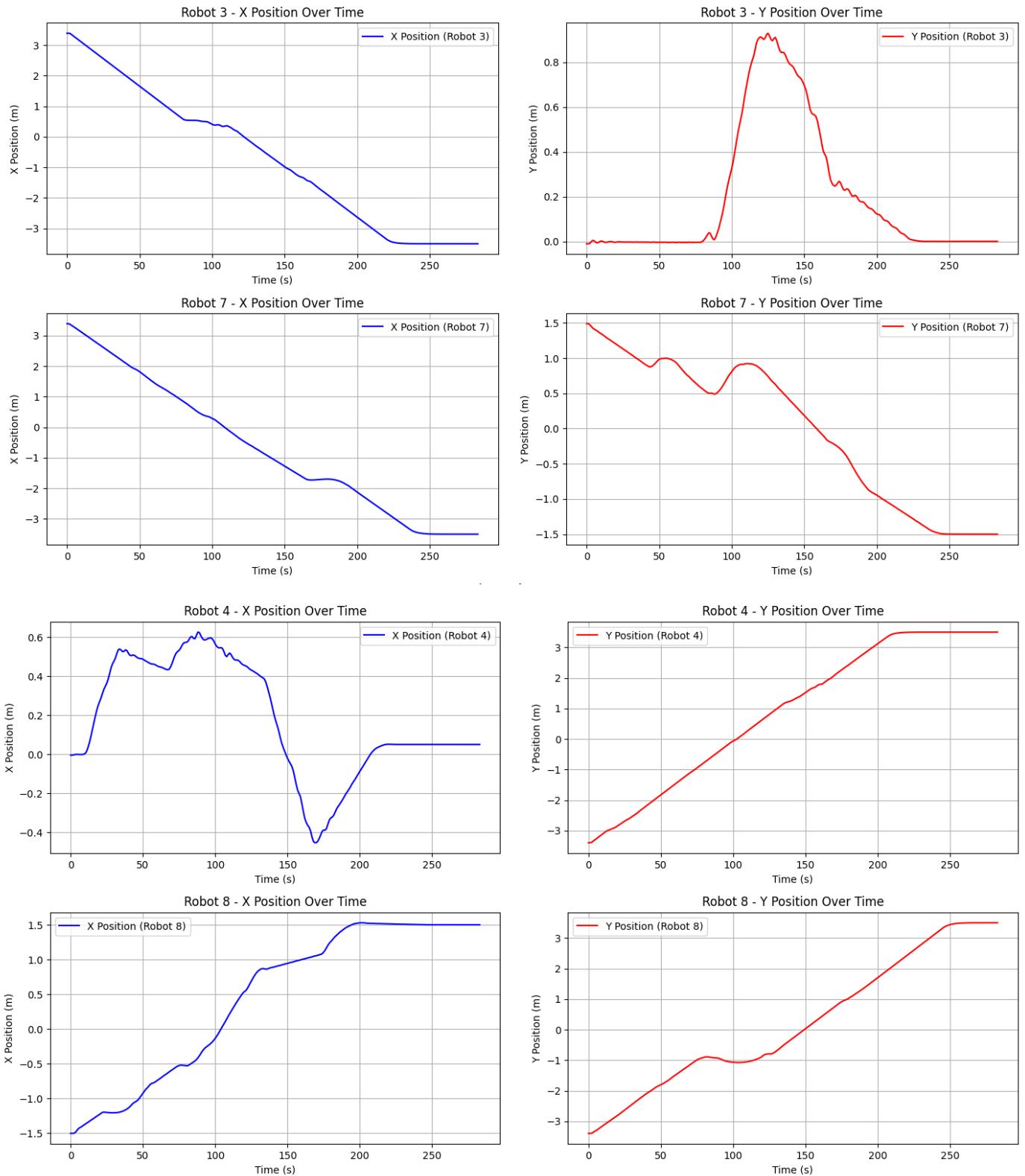


Figure 28: Trajectories of 8 robots the x and y positions

Figure 28 illustrates the trajectories of eight robots during an exploration task, grouped based on their starting and ending positions and represented by different colors. Each figure corresponds

to two robots that share similar movement patterns and trajectory characteristics. The trajectories are color-coded for clarity: blue, red, green, and yellow. Robots with similar starting points and destinations are grouped together, making it easier to observe their behavior and compare their navigation strategies. This grouping highlights the consistency of movements within the same group and differences across groups, providing insights into the control algorithms' effectiveness in guiding robots through the environment.

During the collaborative multi-robot exploration, overlapping paths were observed, but no collisions occurred, demonstrating the effectiveness of the dynamic path adjustment and obstacle avoidance mechanisms. The simulation results show that eight robots can efficiently cover the scene area while avoiding collisions. However, the complexity of path planning is higher compared to the four-robot configuration. From the path diagram, people could find that Due to the increased number of robots, path overlap occurs in certain areas, suggesting that multi-robot systems in high-density environments may face potential path congestion risks.

**Twelve Robots Result:** Figure 29 shows the keyframes for the different stages of the 12 robots simulation process.

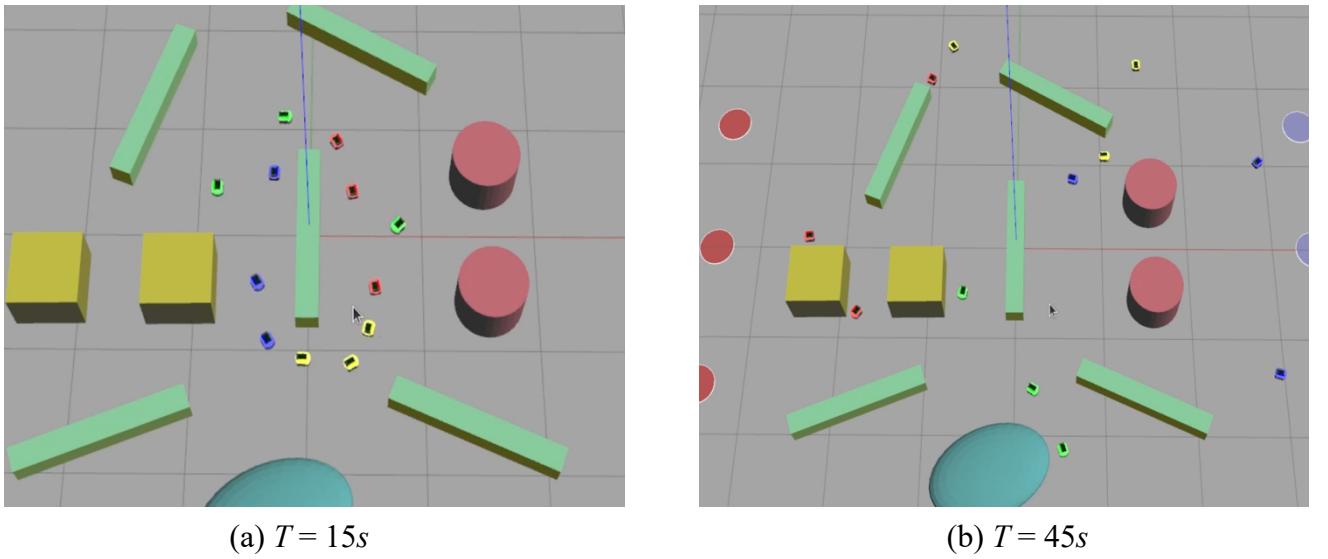


Figure 29: Snapshots for the Twelve Robots Result:  
 (a)  $T = 15s$ , (b)  $T = 30s$

Figure 29(a) shows the distribution of the 12 robots entering the complex obstacle environment at the 15-second mark in the simulation experiment. As the number of robots increases, the space in the obstacle area becomes more crowded, and there are frequent path interactions between the robots. As can be seen in the figure, the robots in the central area almost simultaneously made path corrections and rotated as a whole in a counterclockwise direction. This synchronized movement effectively mitigates potential collision risks and highlights the algorithm's capability for

coordinated scheduling in complex environments. Notably, as the robots gradually approach their respective target positions, they peel off from the rotation path one by one, navigating precisely toward their destination areas. This demonstrates the accuracy and coordination of the path-planning process. Meanwhile, robots on the periphery autonomously adjust their paths in the more open space, skillfully avoiding obstacles while maintaining a safe distance from the robots in the central area. This prevents excessive path overlap, reducing the likelihood of traffic congestion.

Figure 29(b) illustrates the state at 45 seconds, by which time the twelve robots have largely completed their exploration of the dense obstacle region and are beginning to exit the area along different paths. The algorithm's adaptability and path diversity in complex environments is illustrated by the fact that some robots opt to navigate through narrow passages between obstacles in areas with a greater number of obstacles. However, compared to the eight-robot experiment, the increased number of robots and obstacles makes it difficult for robots of the same color to continue exploring in groups of three. The figure clearly shows that robots of the same color exhibit a more dispersed distribution during exploration. This dispersed navigation pattern improves environment coverage and reduces the risk of path overlap and congestion. As a result, the robot team can complete tasks more efficiently in complex environments, minimizing the likelihood of collisions and path crossings. This further ensures the safety and stability of the overall system, even when operating with a large-scale robot fleet.

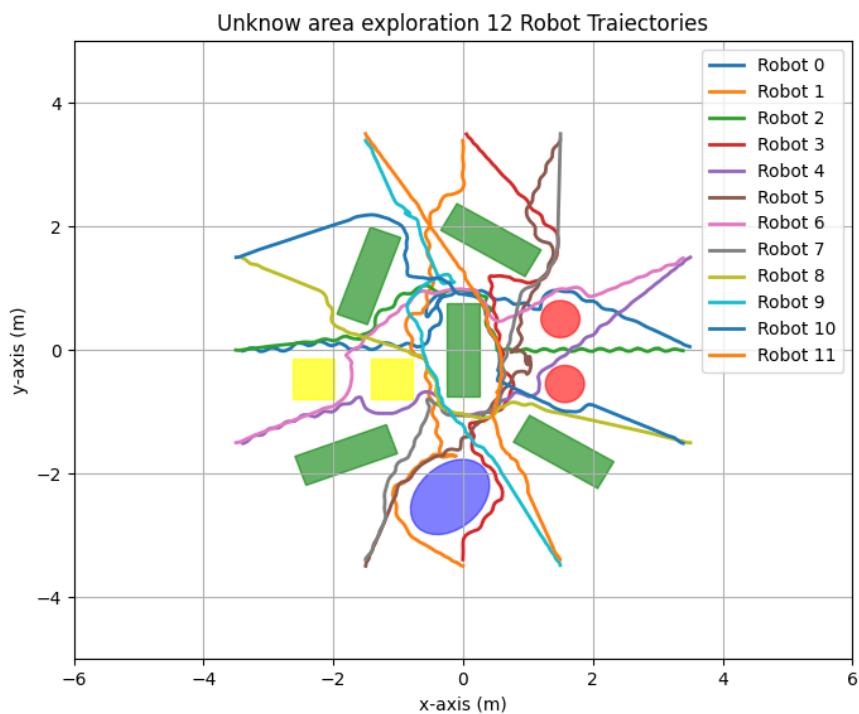
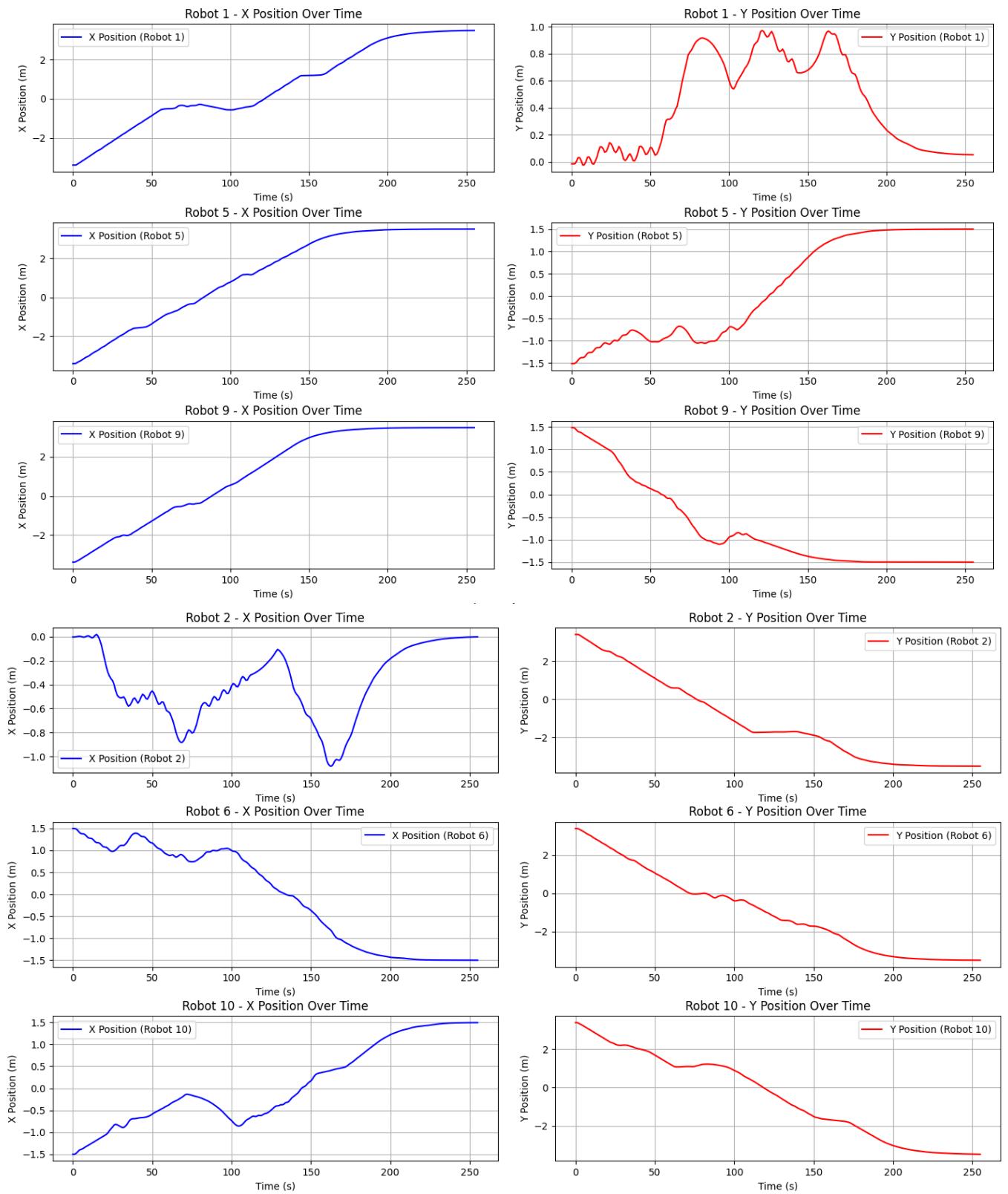


Figure 30: Trajectory map of 12 robots



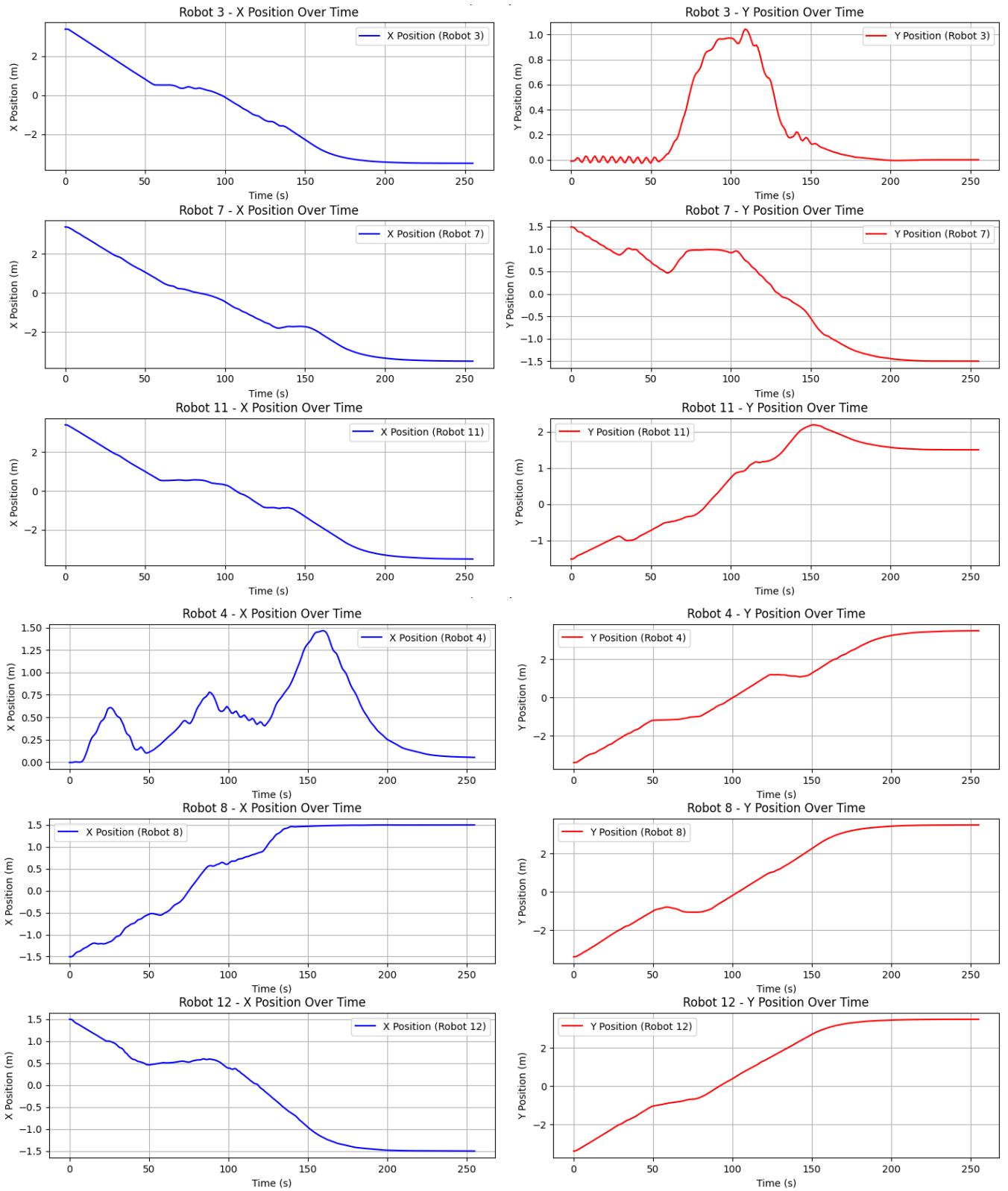


Figure 31: Trajectories of 12 robots the x and y positions

Figure 30 provides a robot trajectory map from a top-down perspective, showcasing the movement paths of the twelve robots. The colors of the trajectories correspond to the IDs of individual robots. The central area is densely packed with obstacles, and the robots' paths weave

around them, demonstrating path diversity and detours. During the collaborative multi-robot exploration, overlapping and intersecting paths are noticeable, especially in the dense obstacle region where path crossings are significant.

Figures 31 depict the trajectories of 12 robots during their exploration tasks, grouped similarly to the 8-robot case. The robots are categorized into four groups, with each group consisting of three robots that share similar starting and ending positions. The grouping is based on trajectory patterns and destination alignment, allowing for a clear comparison of navigation strategies within and across groups. Each group's movement patterns highlight the consistency of the control algorithms, demonstrating their ability to guide multiple robots efficiently in diverse exploration scenarios. This approach maintains a logical structure and ensures clarity when analyzing the behavior of larger robot teams.

As the number of robots increases, some opt for longer detours to avoid collisions with other robots and obstacles, reflecting the flexibility of path planning in complex environments.

The simulation results indicate that the twelve robots effectively complete the exploration tasks in complex environments. Although paths intersect frequently, no collisions occur, highlighting the effectiveness of obstacle avoidance and path planning mechanisms. However, the larger number of robots results in significant path overlap and detours, indicating that in high-density multi-robot collaborative environments, the complexity of path planning and overall navigation time increases substantially.

#### 4.1.1 Unknown area exploration Analysis

##### Task Completion Time

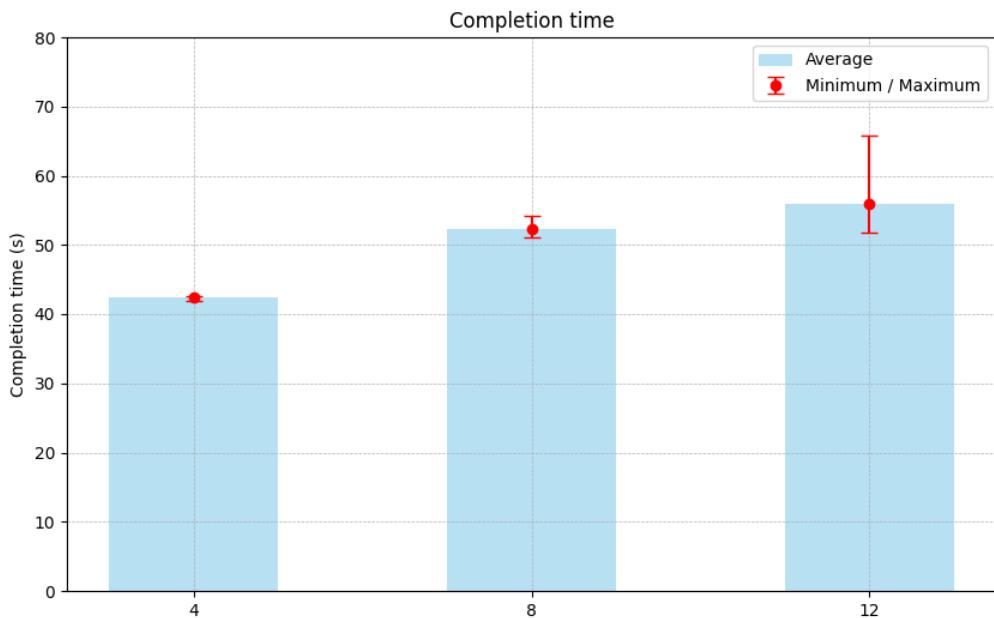


Figure 32: Task Completion Time

Figure 32 presents a comparison of task "Completion Time" for different numbers of robots navigating a complex terrain. The horizontal x-axis represents the number of robots (4, 8, and 12), while the vertical y-axis indicates the time required to complete the task (in seconds). The chart combines bar plots with error bars, visually displaying the average completion time alongside the range of maximum and minimum values for each configuration.

The "Completion Time" comparison is used to evaluate the efficiency and scalability of the system as the number of robots increases, highlighting the impact of robot density on task performance and identifying potential bottlenecks in complex environments.

From Figure 32, it can be observed that task completion time increases gradually as the number of robots increases: With 4 robots, the average completion time is approximately 42 seconds, with minimal difference between the maximum and minimum values. This indicates a stable system performance with little fluctuation in completion times. When the number of robots increases to 8, the average completion time rises to about 52 seconds; it shows a slight difference between the maximum and minimum values and reflects a modest increase in variability. For 12 robots, the average completion time is 55 seconds. The maximum completion time is around 70 seconds, while the minimum drops below 50 seconds. This huge IQR indicates the increase in system uncertainty as robot interactions and path competition intensify.

This trend illustrates that as the number of robots increases, the collaborative complexity rate of the MAS and SE-CBF also increases. Congested paths become a key limiting factor, and problems such as collision avoidance and path planning become tougher. SE-CBF method has demonstrated its effectiveness and success in all experiments. SE-CBF has higher reliability and supports multi-robot collaboration in complex scenarios. On the other hand, the results emphasize the importance of managing the number of robots in complex experiments. When the numbers of robots increase, it is true that task coverage and efficiency can increase and maximize collaborative benefits. However, excessive numbers of robots might lead to path competition, system uncertainty, and instability. Task complexity, completion time, and system stability must balance for best performance. Multi-robot systems need this equilibrium for efficiency and stability.

The experiments were conducted with configurations ranging from 4 to 12 robots. In every scenario, all tasks were completed successfully. This consistent success highlights the strong performance of the SE-CBF method. Additionally, it demonstrates the robustness of SE-CBF when applied to multi-robot systems. Even with 12 robots, where the complexity of path planning and obstacle avoidance was significantly higher, SE-CBF still kept the system security and optimized paths. Consequently, SE-CBF demonstrates its reliability and scalability as a solution for large-scale multi-robot systems.

## Path Length and Complexity

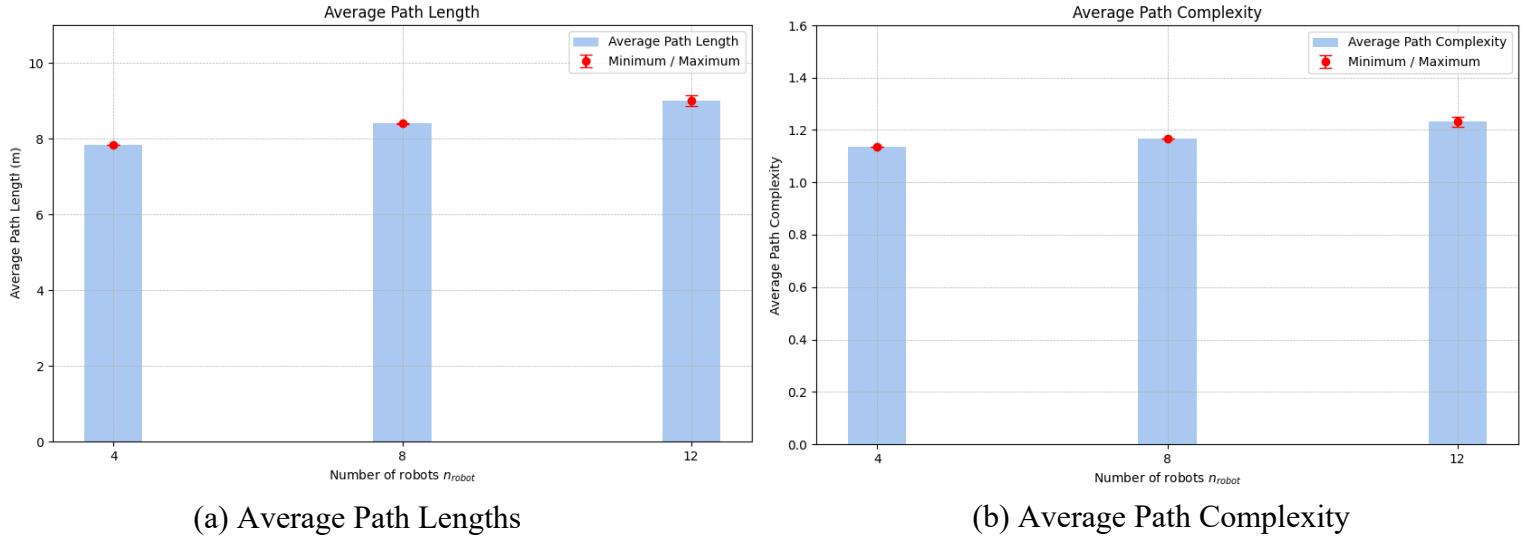


Figure 33: Average Path Length and Complexity

In experiment analysis, the average path length and path complexity offer a comprehensive way to evaluate the performance of the path planning technique. These two metrics provide complementary insights: shorter path lengths indicate higher efficiency in task completion, while lower path complexity reflects smoother trajectories and lower implementation costs. Together, these two metrics reflect how the technique and system balance the space used for navigation and avoidance.

Figure 33(a) compares the "Average Path Lengths" of different numbers of robots (4, 8, and 12) navigating a complex terrain. The horizontal x-axis represents the number of robots, while the vertical y-axis indicates the path length. The bar chart displays the average path length, red diamond points represent the minimum and maximum path lengths, and the error bars show the range of variation.

The "Average Path Length" is the average of the path lengths required for each robot to complete the task in multiple experiments. It is used to measure the overall efficiency of the system's path planning. It reflects the distance moved by the robot in completing the task. The lower the average value, the more optimized and efficient the path planning.

From Figure 33(a), it is clearly illustrated that the path length increases gradually with the number of robots: With 4 robots, the average path length is approximately 7.8, with the minimum and maximum path lengths nearly overlapping, indicating minimal fluctuation and highly stable path planning in a low-density environment. When the number of robots increases to 8, the average path length rises to about 8.4, and the error range widens slightly. With 12 robots, the average path length reaches nearly 9.0, with a further increase in the error range, reflecting a significant rise in uncertainty for path length in high-density conditions.

This trend is primarily due to the dynamic obstacle avoidance requirements of multi-robot systems in complex terrains. As robots interact more frequently and navigate overlapping paths, the number of detours increases, resulting in longer overall paths. However, the growth in path length is relatively gradual, demonstrating that the SE-CBF method effectively reduces path redundancy and maintains high path-planning efficiency.

Figure 33(b) compares the "Average Path Complexity" for different numbers of robots (4, 8, and 12) navigating the complex terrain. The horizontal x-axis represents the number of robots, while the vertical y-axis indicates the path complexity. The bar chart shows the average path complexity, red diamond points represent the minimum and maximum path complexities, and the red bars illustrate the range of variation.

$$\text{Path Complexity} = \frac{\text{Straight Line Distance}}{\text{Total Path Length}}$$

"Average Path Complexity" is an important indicator of the degree of trajectory change during the path-planning process of a robot. To calculate the "Average Path Complexity," the path complexity for each robot is determined as the ratio of the total path length to the straight-line distance between the robot's start and endpoints. For each robot, the total path length is computed by summing the Euclidean distances between consecutive trajectory points, while the straight-line distance is the direct Euclidean distance between the start and endpoints. The average path complexity is then obtained by taking the mean of the path complexities across all robots in the group. It reflects the smoothness and continuity of the path. The higher the path complexity, the more bends and adjustments there are in the path, and the more complex the trajectory planned by the system. At low complexity, the robot path is relatively straight and smooth, while high complexity may increase the time and energy consumption of the robot performing the task.

The figure shows a gradual increase in path complexity as the number of robots rises. With 4 robots, the path complexity is approximately 1.1, and the minimum and maximum values are nearly identical. This indicates low and stable complexity in a low-density environment. When the number of robots increases to 8, the path complexity rises slightly to about 1.15, accompanied by a modest expansion in the error range. For 12 robots, the path complexity reaches 1.22, and the IQR values increase. As the numbers of agents increase, the system becomes more unpredictable.

The observed increase in path complexity corresponds to the robotic system and SE-CBF frequent adjust path and avoid collision. As the density of robots grows, the system must plan more excess trajectories to navigate around obstacles and prevent collisions between agents. This adjustment naturally leads to greater irregularity and unpredictability in their paths. Despite this, as the complexity of experiments increases, SE-CBF also assists the system in completing tasks and

experiments. This indirectly reflects the powerful capability of SE-CBF in path planning, optimization and collision avoidance.

## Minimum distance

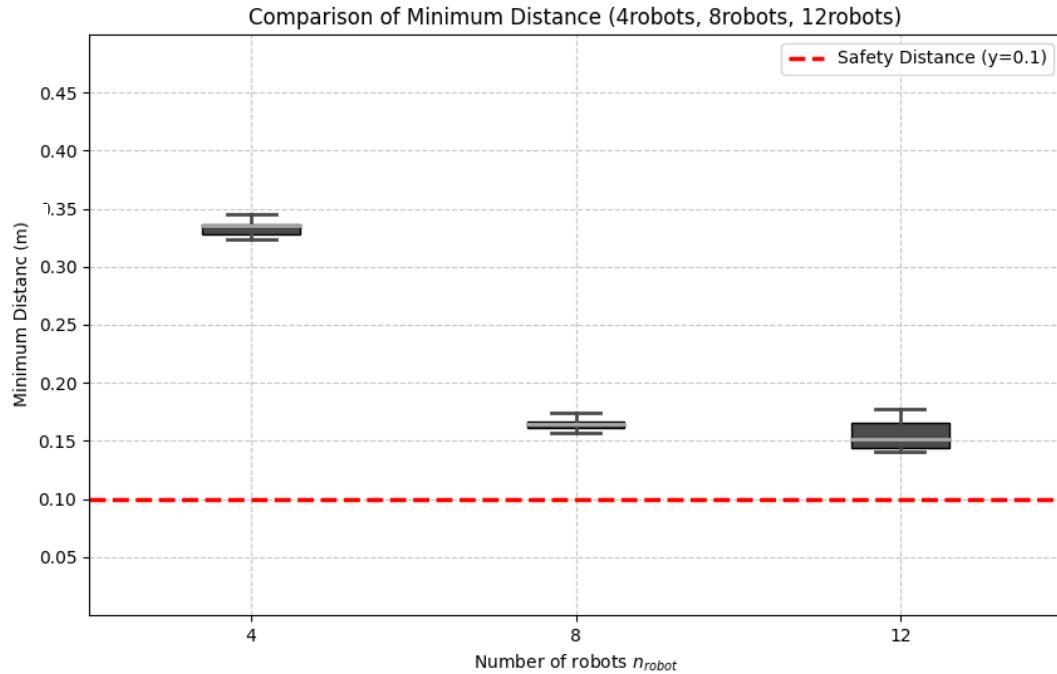


Figure 34: Minimum Distance

Figure 34 uses a boxplot to compare the "Minimum Distance" for different numbers of robots (4, 8, and 12). The horizontal x-axis represents the number of robots, while the vertical y-axis indicates the minimum distance between robots or between robots and obstacles. The chart uses a boxplot to display the distribution of minimum distances, with a red dashed line marking the safety distance threshold ( $y = 0.1$ ).

The "Minimum Distance" metric is a key indicator for assessing system safety in the experiments. It evaluates the effectiveness of the obstacle avoidance algorithm under extreme conditions. This metric is of great importance since it provides a close relative measure of the system's capability to keep a safe distance from the obstacles and other robots during the experiment. The minimum distance data can assist the researcher in measuring the robustness of the control strategy and the collision prevention capabilities of the technique. Furthermore, the minimum distance data might show the domain of the algorithm which requires improvement.

From Figure 34, it can be observed that as the number of robots increases, the minimum distance shows a gradual decline. All data remains above the safety distance threshold and reflects the experiment's success. With 4 robots, the minimum distance ranges between 0.32 and 0.34, highly over the safety distance threshold. This indicates that the agent keeps a large and stable distance from each other and obstacles and high security. As the number of robots increases to 8,

the minimum distance decreases to a range of 0.15 to 0.18. This data is lower than the data of minimum distance for 4 robots, but it is still above the safety distance threshold. The median minimum distance drops to approximately 0.17, and the range of data expands. The IQR reflects an increase in variability for 8 robots. Next, when the number of robots further increases to 12, the median minimum distance decreases to around 0.15, with a broader data distribution range. Some experimental results are close to the safety distance threshold. However, the lowest value consistently remains above 0.1, demonstrating that the system can maintain basic safety even with 12 numbers of robots.

The trend in Figure 34 reveals that as the number of robots increases, the system faces greater challenges in obstacle avoidance. This is particularly evident in the scenario with 12 robots, where the complexity of path planning and collision avoidance increases. The underlying reason for this phenomenon is the increasing constraints on available space. As more robots work within an experiment, the available space can be used for movement reduces. Consequently, the system and technique need to use limited space to avoid collision and navigate. This challenge is reflected in the diagram, where the minimum distance between agents and obstacles decreases as the number of robots rises. This reduction in distance signifies the increase in competition and emphasizes the need for provably safe and stable controls in such congested environments.

Despite the decrease in available space, the results of the experimentation have performed above the safety distance threshold which indicates the accuracy of the strategy employed for obstacle avoidance. This reveals the dependability of the SE-CBF technique in handling multi-robot obstacle avoidance tasks. The experimental results strongly validate the SE-CBF method's superior obstacle avoidance capability in multi-robot systems.

The experimental results comparative analysis given demonstrates the most important features that SE-CBF brings to multi-agent systems. In terms of task completion time, SE-CBF significantly reduces the period required for robots to complete complex tasks. This improvement proves the capacity to optimize operational efficiency for multi-agents under complex situations. Moreover, SE-CBF effectively minimizes path complexity and average path length, providing smoother and more efficient trajectories for the system. It proves that SE-CBF reduces unnecessary delay and maximizes space utilization. Furthermore, based on the minimum distance data, SE-CBF can guarantee the safest distance between all the robots and the surrounding obstacles. This guarantees sufficient collision protection, enhancing the safety of the system. According to these three metrics data analyses, the study states that SE-CBF offers a flexible and robust solution for multi-agent systems in complex environments. This makes it possible to overcome the challenges imposed by the previous method.

## 4.2 Experimental Validation in Real-World Environments

In addition to simulation studies, the research uses real-world experiments to test the effectiveness of the SE-CBF method. For this purpose, a one-way, one-lane intersection scenario was designed, as shown in Figure 35. The experimental setup included four robots, each vehicle entering the intersection from one of four directions. The experiment required all robots to navigate through the intersection simultaneously, reaching their target while avoiding collisions. A rectangular obstacle was placed at the center of the intersection to increase the complexity of the task. The objective was for the robots to successfully navigate around the obstacle, minimize unnecessary space usage, and reach their designated goals without collisions.

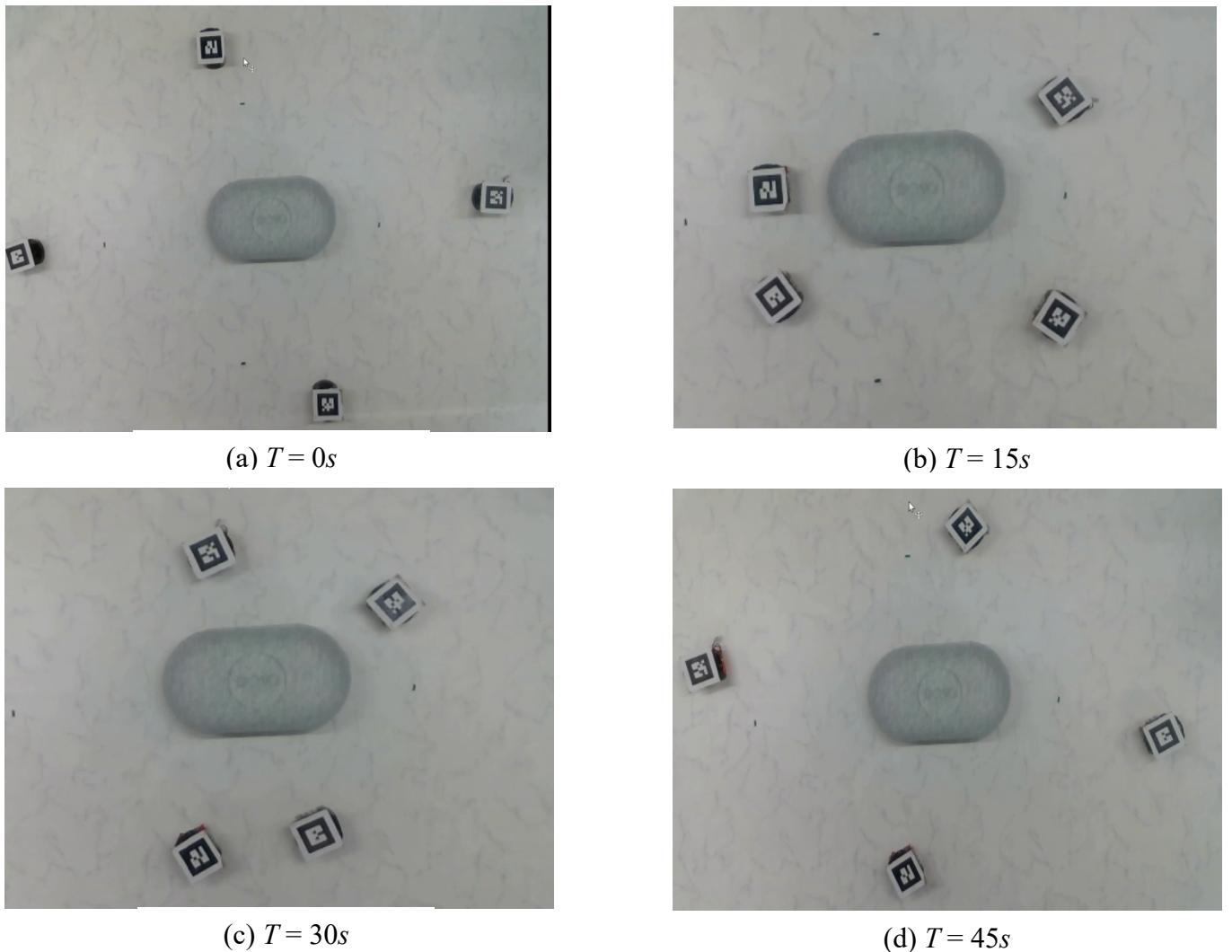


Figure 35: Snapshots for the SE-CBF Intersection under simulation:  
(a)  $T = 0s$ , (b)  $T = 15s$ , (c)  $T = 30s$ , (d)  $T = 45s$

During the experiment, four robots performed obstacle avoidance and path planning. Figure 35(a) ( $T = 0s$ ) shows the initial state of the experiment, where the four robots are evenly distributed

around the experiment's four directions. At this point, the robots are positioned at their preset starting locations, ready to enter the real-world intersection experiment.

As the experiment starts, Figure 35(b) ( $T = 15\text{s}$ ) captures the robots beginning to move. The robots start avoiding the obstacle and move toward their respective targets. During this phase, they maintain appropriate spacing and begin to move to the opposite side.

In the mid-phase of the experiment, as shown in Figure 35(c) ( $T = 30\text{s}$ ), the robots close the edges of the obstacle area. The robots continuously adjust their positions in real-time. Despite nearing the central obstacle, the robots maintain safe distances, effectively reducing collision risks.

Finally, Figure 35(d) ( $T = 45\text{s}$ ) illustrates the final state of the experiment. At this stage, all robots successfully navigate around the central obstacle and reach their target. The robots are now dispersed around the obstacle, and the obstacle avoidance process has been completed smoothly. Notably, no collisions occurred throughout the experiment.

This experimental result validates the SE-CBF's effectiveness and safety. It demonstrates that the SE-CBF can enable multi-robot systems to achieve autonomous obstacle avoidance and path planning in real-world applications.

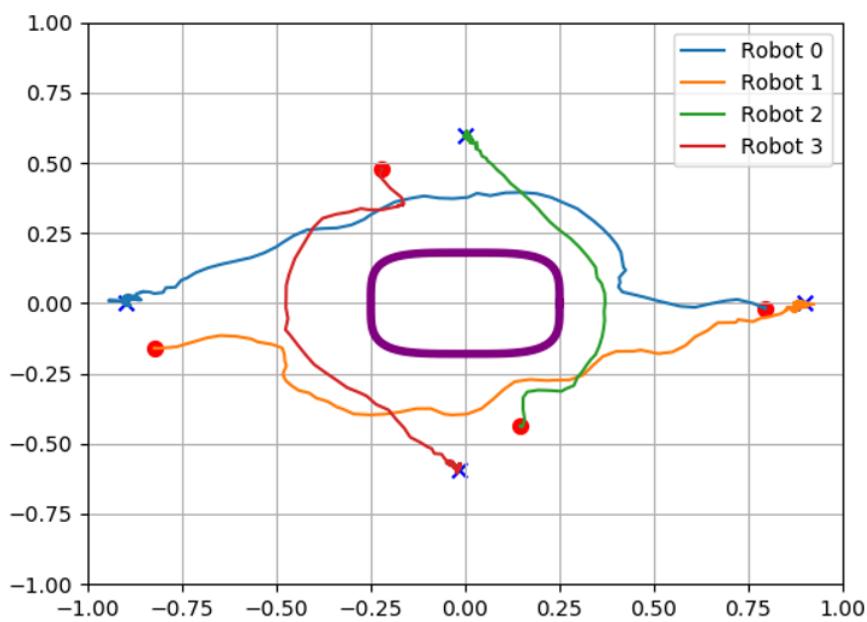


Figure 36: Real-wrold Experiment Trajectories Map of four robots

Figure 36 illustrates the trajectories of four robots during the obstacle avoidance task. The purple area represents the central obstacle, while the colored curves (blue, orange, green, and red) depict the movement paths of the robots. The starting points of the trajectories are marked with circles, and the endpoints are indicated by "x" symbols. The figure clearly shows that the robots successfully navigated around the obstacle, avoided collisions, and reached their designated target positions. The pictorial view in Figure 36 illustrates a successful implementation of obstacle

avoidance path planning. Each robot adopts a unique path around the obstacle, which they consistently manage throughout the task. Even while working in a communal space, the robots steer clear of each other, which shows efficiency in dealing with multi-agent system collisions.

Such an outcome proves the control algorithm used with the multi-robot system to be practically reliable and robust for the implementation of obstacle avoidance tasks in more complex environments. It also demonstrates the algorithm's ability of safe and efficient navigation in an environment which makes it suitable for multi robot applications in the real world.

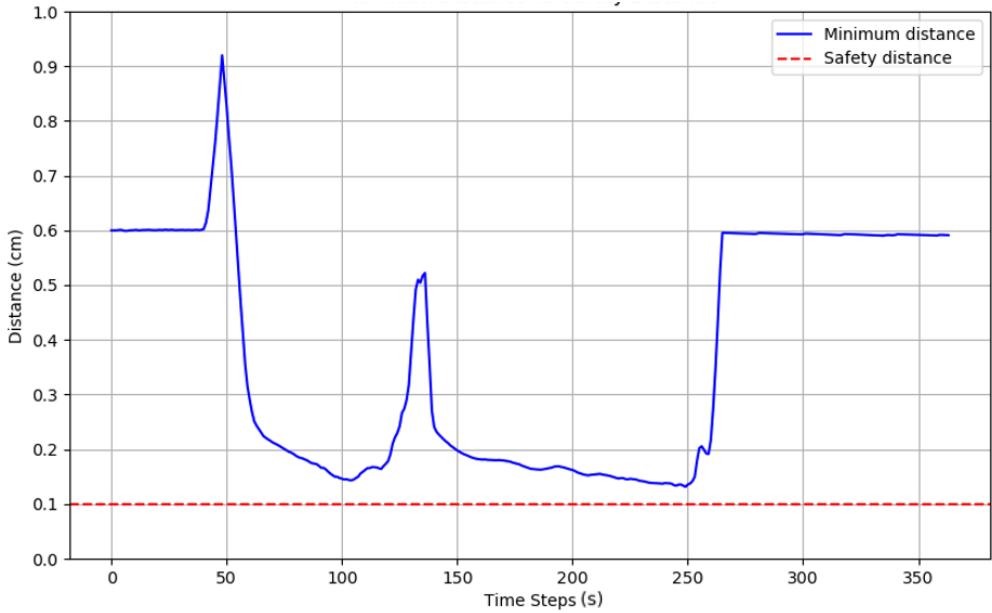


Figure 37: Real-world Experiment Minimum Safety Distance

Figure 37 illustrates the variation in minimum distances between robots overtime during the experiment. The vertical y-axis represents the distance (in centimeters), while the horizontal x-axis denotes the time steps. The blue curve tracks the minimum distance recorded among all robots at each time step, and the red dashed line marks the safety distance threshold (0.1 cm).

From the figure, it is evident that the minimum distance consistently remains above the red safety line throughout the experiment. Although slight decreases in the minimum distance are observed midway through the experiment, these dips never breach the safety threshold. This trend highlights the robustness of the designed control strategy, which effectively maintains safe distances between robots even during moments of increased spatial complexity. Overall, the results confirm that the control strategy ensures collision-free operation at all times, providing a reliable framework for managing robot interactions in dynamic and high-density environments.

# Chapter 5

## Discussion and Conclusion

This study proposed a novel Super-Ellipse Control Barrier Function (SE-CBF) to address the limitations of traditional Control Barrier Functions (CBF) in obstacle avoidance and path planning for mobile robots. The SE-CBF model includes dynamic safety distance that adapts to irregular obstacle shapes. The new SE-CBF makes the robotic system work in complex environments. This chapter discusses the key findings and evaluates the effectiveness of the SE-CBF, its limitations, and future research directions.

The experimental results from simulations and real-world tests validate the performance advantages of SE-CBF over traditional CBF:

- **Enhanced Adaptability:** SE-CBF dynamically adjusts the safety distance based on obstacle shape. This flexibility significantly improves MAS's path planning.
- **Improved Space Utilization:** SE-CBF minimizes redundant safety distances, allowing robots to navigate efficiently through narrow passages and densely populated areas.
- **Increased Efficiency:** SE-CBF reduced task completion times and smoother trajectories in both simulated traffic intersections and exploration experiments.
- **Scalability:** The SE-CBF method effectively manages multi-agent interactions, ensuring collision-free operation and optimized navigation in systems with multi-agent density.

The development of SE-CBF has huge implications for autonomous systems, particularly in robotics, logistics, and autonomous vehicles. The SE-CBF enhances the feasibility of deploying mobile robots in complex real-world scenarios such as crowded warehouses, urban environments, and disaster response operations. Moreover, the SE-CBF technique allows the combination of sophisticated obstacle avoidance techniques and multi-agent coordination approaches.

Although the SE-CBF method shows significant advantages over traditional CBF approaches, it is not without limitations. One limitation of SE-CBF is the computational overhead due to the real-time adjustments of safety zones. When the tasks associated with multi-agent environments become complex, it would be challenging for systems to remain quickly responsive. The new technique relies on the accuracy of sensor data for obstacle detection and localization. Any inaccuracies or delays in sensor inputs could lead to dangerous performance or safety risks. Furthermore, the scope of real-world experiments conducted in this study was limited to a controlled environment with a number of robots and available space. While these tests provided a

valuable understanding of SE-CBF's capabilities, they may not fully reflect the problem when working in large-scale or highly dynamic scenarios. These limitations provide opportunities for the refinement and testing of the method.

The limitations of SE-CBF outlined in this work can be the focus of future research as well as its broader application. One area of improvement is the optimization of the execution time of the algorithm so it can be implemented in real-time where agents are densely located. This may involve use of more sophisticated heuristics or even the deployment of machine learning. System's failure tolerance must also be increased which is essential for enhancing the performance quality. Large scale experiments with varying degrees of robots and obstacles would reveal the shortcomings of SE-CBF and mitigate them. These directions of research are expected to provide a more detailed and effective SE-CBF approach to robotics for refinement and robotics for adaptation.

This study demonstrates the feasibility and effectiveness of SE-CBF as a robust and flexible obstacle avoidance strategy for autonomous robots. Limitations with conventional CBF techniques have been addressed; SE-CBF significantly enhances navigation efficiency, adaptability, and safety in complex environments. These advancements mark a critical step toward the broader application of mobile robotics in real-world scenarios, setting the stage for further innovations in autonomous navigation and control systems.

# References

- [1] Krause, M., & Rossiter, J. (2024). Highly stretchable additively manufactured capacitive proximity and tactile sensors for soft robotic systems. *IEEE Sensors Journal*, 24(1), 123-130. <https://doi.org/10.1109/JSEN.2024.10057215>
- [2] Smith, J., & Brown, L. (2021). Use of interaction design methodologies for human–robot collaboration in industrial applications. *IEEE Transactions on Human-Machine Systems*, 51(4), 345-356. <https://doi.org/10.1109/THMS.2021.9537740>
- [3] Sousa, N., Oliveira, N., & Praça, I. (2021). A multi-agent system for autonomous mobile robot coordination. arXiv preprint. <https://doi.org/10.48550/arXiv.2109.12386>
- [4] Ames, A. D., Coogan, S., Egerstedt, M., Notomista, G., Sreenath, K., & Tabuada, P. (2019). Control Barrier Functions: Theory and Applications. arXiv. <https://arxiv.org/abs/1903.11199>
- [5] Lee, J., Kim, S., & Park, D. (2021). Development of a 3D point cloud-based robot localization and navigation system for urban environments. *Journal of Robotics and Autonomous Systems*, 135, 103709. <https://doi.org/10.20965/jrm.2015.p0356>
- [6] Meng, H., Zhang, L., & Liu, Q. (2021). Dynamic path planning of mobile robots using adaptive dynamic programming *IEEE Transactions on Industrial Electronics*, 68(5), 4398-4407. <https://doi.org/10.1016/j.eswa.2023.121112>
- [7] Chunxi Cheng, Qixin Sha, Bo He, Guangliang Li, Path planning and obstacle avoidance for AUV: A review, *Ocean Engineering*, Volume 235, 2021, 109355, ISSN 0029-8018, <https://doi.org/10.1016/j.oceaneng.2021.109355>
- [8] Anish Pandey, Saroj Kumar, Krishna Kant Pandey, Dayal R. Parhi, Mobile robot navigation in unknown static environments using ANFIS controller, *Perspectives in Science*, Volume 8, 2016, Pages 421-423, ISSN 2213-0209, <https://doi.org/10.1016/j.pisc.2016.04.094>.
- [9] Saboukhi, A. (2023). Designing and implementing a small-size Automated Fiber Placement (AFP) head capable of depositing thermoset layers on V-shape structures (Doctoral dissertation, Concordia University).
- [10] Feng, Z., Wang, J., & Chen, R. (2021). Improved artificial potential field algorithm for collision avoidance in autonomous vehicles. *IEEE Transactions on Intelligent Vehicles*, 6(2), 298-306.
- [11] Cen, L., & Cheng, Z. (2007). Coordinating potential field and genetic algorithm for dynamic mobile robot path planning. *Robotics and Autonomous Systems*, 55(12), 970-982.
- [12] Yang, X., Liu, H., & Zhang, P. (2017). Trajectory planning of six-DOF manipulators based on GA-APF algorithm. *Journal of Mechanical Engineering*, 53(5), 425-433.
- [13] Buch, J., Liao, S.-C., & Seiler, P. (2021). Robust control barrier functions with sector bounded uncertainties. *IEEE Control Systems Letters*, 5(4), 1325-1330. <https://doi.org/10.1109/LCSYS.2021.3073086>

- [14] Lopez, B. T., Slotine, J.-J. E., & How, J. P. (2020). Robust adaptive control barrier functions: An adaptive and data-driven approach to safety. *IEEE Control Systems Letters*, 5(3), 1031–1036. <https://doi.org/10.1109/LCSYS.2020.3047417>
- [15] Pu, F., Liang, J., Wang, S., Yu, X., Ji, X., Chen, Y., Zhang, K., Shi, R., & Wu, W. (2024). Hierarchical Consensus-Based Multi-Agent Reinforcement Learning for Multi-Robot Cooperation Tasks. arXiv. <https://arxiv.org/abs/2407.08164>
- [16] Lan, X., Qiao, Y., & Lee, B. (2021, February). Towards pick and place multi-robot coordination using multi-agent deep reinforcement learning. In 2021 7th International Conference on Automation, Robotics and Applications (ICARA) (pp. 85-89) doi: 10.1109/ICARA51699.2021.9376433.
- [17] Shalev-Shwartz, S., Shammah, S., & Shashua, A. (2016). Safe, multi-agent, reinforcement learning for autonomous driving. arXiv preprint arXiv:1610.03295.
- [18] Tan, X., & Dimarogonas, D. V. (2021). Distributed implementation of control barrier functions for multi-agent systems. *IEEE Control Systems Letters*, 6, 1879-1884.
- [19] Lyu, Y., Luo, W., & Dolan, J. M. (2022, October). Responsibility-associated multi-agent collision avoidance with social preferences. In 2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC) (pp. 3645-3651). IEEE.
- [20] Patle, B. K., Babu, L. D., Pandey, A., Parhi, D. R., & Jagadeesh, A. (2017). Path planning in uncertain environment by using firefly algorithm. *Defence Technology*, 13(1), 47-52. <https://doi.org/10.1016/j.dt.2018.06.004>
- [21] Xiao, W., Belta, C., & Cassandras, C. G. (2021). Adaptive control barrier functions. *IEEE Transactions on Automatic Control*, 67(5), 2267-2281.
- [22] Cheng, R., Khojasteh, M. J., Ames, A. D., & Burdick, J. W. (2020, December). Safe multi-agent interaction through robust control barrier functions with learned uncertainties. In 2020 59th IEEE Conference on Decision and Control (CDC) (pp. 777-783). IEEE.
- [23] Srinivasan, M., Abate, M., Nilsson, G., & Coogan, S. (2021). *Extent-compatible control barrier functions*. *Systems & Control Letters*, 150, 104895. <https://doi.org/10.1016/j.sysconle.2021.104895>
- [24] Zhang, Z., & Zhao, Z. (2014). A multiple mobile robots path planning algorithm based on A-star and Dijkstra algorithm. *International Journal of Smart Home*, 8(3), 75-86. [https://gvpress.com/journals/IJSH/vol8\\_no3/7.pdf](https://gvpress.com/journals/IJSH/vol8_no3/7.pdf)
- [25] X. Song, "Research and design of robot obstacle avoidance strategy based on multi sensor and fuzzy control," in 2022 IEEE 2nd International Conference on Data Science and Computer Application (ICDSCA), pp. 930–933, 2022
- [26] Kurt Dresner and Peter Stone. A multiagent approach to autonomous intersection management. *Journal of Artificial Intelligence Research*, 31:591–656, 2008
- [27] H. Su and M. Liu, "Force sensor based motion control of wheeled mobile robots," 2005 IEEE International Conference on Information Acquisition, Hong Kong, China, 2005, pp. 5 pp.-, doi: 10.1109/ICIA.2005.1635146.

- [28] S. B. Marapane, M. Holder and M. M. Trivedi, "Coordinating motion of cooperative mobile robots through visual observation," Proceedings of IEEE International Conference on Systems, Man and Cybernetics, San Antonio, TX, USA, 1994, pp. 2260-2265 vol.3, doi: 10.1109/ICSMC.1994.400201.
- [29] B. Lisien, D. Morales, D. Silver, G. Kantor, I. Rekleitis and H. Choset, "Hierarchical simultaneous localization and mapping," Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453), Las Vegas, NV, USA, 2003, pp. 448-453 vol.1, doi: 10.1109/IROS.2003.1250670.
- [30] Rondoni, C., Scotto di Luzio, F., Tamantini, C., Tagliamonte, N. L., Chiurazzi, M., Ciuti, G., & Zollo, L. (2024). Navigation benchmarking for autonomous mobile robots in hospital environment. *Scientific Reports*, 14, Article 18334. <https://doi.org/10.1038/s41598-024-69040-z>
- [31] Kang, M., Chen, Q., Fan, Z., Yu, C., Wang, Y., & Yu, X. (2024). A RRT-based path planning scheme for multi-DOF robots in unstructured environments. *Computers and Electronics in Agriculture*, 218, 108707. <https://doi.org/10.1016/j.compag.2024.108707>