PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA
MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH

Mohamed Boudiaf University of M'sila
Faculty of Mathematics and Computer Science
Department of Mathematics

Ordre Number: .....

THESIS

Presented for attainment the degree of

Doctorate in Sciences

Specialty: Mathematics.
Option : Applied Mathematics.

by

GUETTAF Wahid

Title

Coordination of Multi-Robot Systems

Defended on: 22/04/2026 before the jury composed of:

SAADI Khalil	Prof	University of M'sila	President
BOUDERAH Brahim	Prof	University of Mostaganem	Supervisor
BENSALEM Naceureddine	Prof	University of Sétif 1	Examiner
SAADAOUI Kheir	MCA	University of M'sila	Examiner
BENAISSA Lakhdar	MCA	Algiers University 1	Examiner

Academic Year: 2025-2026

Dedication

To my parents and grandparents,
for your boundless love and wisdom. Your sacrifices, guidance, and unwavering belief in me have shaped who I am today. I am eternally grateful for the values you instilled in me, values that continue to inspire my journey in life and in this work.

To my wife and daughters,
for your love and steadfast support, which form the foundation of my strength. To my wife, for your patience, encouragement, and unwavering faith in me; and to my two beautiful daughters, for bringing immeasurable joy and meaning to my life. This thesis is a tribute to your love and sacrifices, and I dedicate it to you with all my heart.

To my teachers,
for your invaluable mentorship, wisdom, and the knowledge you have generously shared with me over the years. I am deeply grateful for your dedication and for nurturing my passion for learning. This work stands as a testament to the profound impact you have made on my academic and personal growth.

Acknowledgements

I would like to express my deepest gratitude to my thesis supervisor, Professor Brahim Bouderah, for his unwavering support, insightful guidance, and remarkable patience throughout the course of this work. His encouragement and scholarly counsel have been invaluable in shaping this research and in helping me overcome numerous challenges.

I am also sincerely grateful to the members of the examination committee, Prof Khalil Saadi, Prof Naceureddine Bensalem, and Dr. Kheir Saadaoui, for agreeing to examine my thesis and for their constructive feedback and valuable suggestions, which significantly enriched the final version of this thesis.

I would like to extend my heartfelt thanks to Professor Nadhir Messaï from the University of Reims for his constant encouragement and for generously welcoming me to his office at CRESTIC. His support and thoughtful insights during our discussions were instrumental in advancing this research.

I am deeply appreciative of Dr. Kahina Louadj from the University of Bouira for her kind support and invaluable guidance. I would also like to express my sincere thanks to Professor Philippe Marthon, Emeritus Professor at ENSEIHT, Toulouse, for his insightful comments and for generously sharing his expertise during our discussions.

Contents

Genral Introduction **4**

1 Introduction to Multi-Robot Systems and Coordination Approaches **7**

1.1 Multi-Robot Systems 7

1.1.1 Robots 7

1.1.2 Communication Systems 8

1.2 Classical Coordination Approaches 9

1.2.1 Centralized Coordination 10

1.2.2 Distributed Coordination 14

1.3 Coordination Problems 19

1.3.1 Real-Time Synchronization 19

1.3.2 Partial and Incomplete Communication 19

1.3.3 Direct Robot Interactions 20

1.3.4 Redundancy in Robot Roles 21

1.3.5 Decision-Making Conflicts 21

1.3.6 Robot Heterogeneity and Coordination 22

1.3.7 Scalability and Coordination in Large-Scale Systems 23

1.4 Practical Application Examples 24

1.4.1 Industrial Applications 24

1.4.2 Space Exploration 25

1.4.3 Search and Rescue Operations 25

1.4.4 Agricultural Applications 26

2 Mathematical Modeling for Multi-Robot Formation Problem **28**

2.1 Introduction to the Multi-Robot Formation Problem 29

2.1.1 Mathematical Definition of the Formation Problem 29

2.1.2 Role Assignment and Formation Topology 29

2.2 Kinematic Modeling for Multi-Robot System 33

2.2.1 The Single-Robot Kinematic Model: A Foundational Building Block . . . 34
2.3 Graph Theory . . . 36
2.3.1 Background and Representation of Multi-Robot Systems . . . 37
2.3.2 Spectral Graph Theory . . . 40
2.4 Formation Control Problems . . . 44
2.4.1 Mathematical Formulation of Formation Control . . . 45
2.4.2 Formation Control Problem . . . 46
2.4.3 Graph-Based Formation Control . . . 47

3 Safe Multi-Robot Systems: A Control Barrier Function Approach 50
3.1 Introduction and Motivation . . . 50
3.2 Control Barrier Function (CBF) Theory . . . 51
3.2.1 Control Lyapunov Functions . . . 54
3.2.2 Control Barrier Functions . . . 56
3.3 High-Order Control Barrier Functions . . . 59
3.4 Integration with Multi-Robot Systems . . . 63
3.4.1 System Model and Safety Objectives . . . 63
3.4.2 Barrier Function Construction for Multi-Robot Safety . . . 63
3.4.3 High-Order Control Barrier Functions for Multi-Robot Systems 64
3.4.4 Decentralized Safety-Critical Control Synthesis . . . 65
3.5 Case Study: Application of High-Order Control Barrier Functions . . . 65
3.5.1 System Dynamics and Environment . . . 65
3.5.2 Safety Constraints and HOCBF Design . . . 66
3.5.3 Control Synthesis and Simulation Setup . . . 68
3.5.4 Results and Discussion . . . 68
3.6 Conclusion . . . 71

4 HOCBF-Based Safe Formation Control for Multi-Robot Systems 74
4.1 Introduction . . . 74
4.1.1 Formation Deformation . . . 74
4.1.2 Control Barrier Functions (CBF): . . . 75
4.1.3 Related Work . . . 75
4.2 Problem Definition . . . 76
4.2.1 Multi-Robot System with Double Integrator Dynamics . . . 76
4.2.2 Desired Formation . . . 77
4.2.3 Controlled Formation Deformation and Recovery . . . 78
4.2.4 Control Objectives . . . 78
4.3 Control Barrier Functions . . . 78
4.3.1 Higher-Order Control Barrier Functions . . . 79

4.4 Case Studies in Multi-Robot Formation Control 80
4.4.1 Formation Preservation in Navigation 80
4.4.2 Formation Reconfiguration via Scaling and Rotation 82
4.4.3 Formation Reconfiguration via Shape Change 85
4.5 Conclusion 87

General Conclusion **89**

Bibliography **91**

Genral Introduction

Over the past decades, the field of multi-robot systems (MRS) has emerged as a central research area within robotics, driven by the increasing demand for autonomous and cooperative robotic platforms in industrial automation, environmental monitoring, transportation, and critical missions such as search and rescue. Unlike single-robot systems, multi-robot teams offer enhanced robustness, scalability, and efficiency by distributing tasks among several agents. However, achieving coordinated behavior in such systems remains a fundamental challenge, requiring advanced models of communication, decision-making, and control. Classical coordination paradigms including centralized, decentralized, and distributed architectures, have been extensively studied to address issues of real-time synchronization, incomplete information sharing, and heterogeneous robot capabilities [1, 2]. These approaches laid the foundation for modern cooperative robotics by establishing rigorous frameworks for interaction and consensus.

Within coordinated multi-robot systems, formation control has become one of the most prominent subfields, enabling robots to maintain specific geometric arrangements while navigating through dynamic environments. Formation control methods have traditionally relied on graph-theoretical representations[3, 4], where inter-robot relationships are modeled through sensing or communication edges [5]. Early contributions employed potential fields, leader–follower schemes[6, 7], and consensus-based control[8, 9, 10], providing simple yet effective solutions for small teams. However, as modern applications demand higher autonomy, larger-scale deployments, and resilience to disturbances, these classical models face limitations, particularly in handling dynamic obstacles, communication delays, and local decision conflicts.
As multi-robot systems increasingly operate in complex, uncertain, or safety-critical environments, establishing guaranteed safety has become imperative. Control Barrier Functions (CBFs) have recently emerged as a powerful formal framework for enforcing safety constraints in nonlinear and multi-agent systems [11, 12, 13, 14]. By ensuring forward invariance of safe sets, CBFs allow robots to maintain prescribed minimum distances, avoid collisions, and satisfy dynamic constraints in real time. For

systems with higher relative degrees, such as double integrator dynamics. Higher-Order CBFs (HOCBFs) extend these guarantees by explicitly incorporating higher-order state derivatives into the safety formulation [15, 16, 17]. These tools have opened new opportunities for safety-critical coordination across heterogeneous robotic teams.

Despite these advances, significant challenges remain in enabling multi-robot systems to adapt their formation shapes or behaviors while preserving safety guarantees. Real-world environments often require teams to deform, scale, rotate, or completely reconfigure their formations to avoid obstacles or achieve mission objectives. Recent research has thus explored deformable and flexible formations, distributed quadratic programming (QP) schemes, and local consensus-based optimization techniques that allow robots to compute safe control inputs using only neighbor information [14]. Nevertheless, the integration of safety-critical control, formation adaptation, and distributed optimization remains an open research problem, particularly in large-scale or communication-constrained systems.

The present thesis contributes to this research effort by developing a unified framework for safe and adaptive formation control using Higher-Order Control Barrier Functions in combination with distributed coordination strategies. The approach enables multi-robot teams to preserve their formation, perform controlled deformations, and recover their nominal geometric structure after avoiding obstacles. Through detailed mathematical modeling, theoretical guarantees, and case studies, including formation preservation, scalable deformation, and shape morphing, this work advances the state of the art in safety-critical coordination of multi-robot systems. The results demonstrate the potential of combining formal safety methods with distributed control to enable robust, scalable, and autonomous multi-robot coordination in real-world environments.

The remainder of this thesis is organized as follows. Chapter 1 provides the foundational background for multi-robot coordination, introducing the mathematical modeling of multi-robot systems based on double-integrator dynamics, as well as the graph-theoretical tools used to represent interactions and formation structures. The chapter also reviews classical coordination strategies and highlights their limitations in dynamic or constrained environments.

Chapter 2 formulates the multi-robot formation control problem, defining the desired formation, the mechanisms governing deformation and reformation, and the set of control objectives that must be satisfied, ranging from formation maintenance to collision avoidance and efficient energy use. This chapter establishes the problem framework upon which the safety-critical control strategy is built.

Chapter 3 develops the proposed safety-critical coordination methodology. It intro-

duces Control Barrier Functions (CBFs) and Higher-Order CBFs (HOCBFs) for systems with higher relative degree, and incorporates these constraints into a distributed Quadratic Programming (QP) formulation. The chapter details how each robot computes its control input using only local information from its neighbors, ensuring decentralized execution, guaranteed safety, and the ability to accommodate formation deformation.

Finally, Chapter 4 presents a comprehensive set of case studies that illustrate the effectiveness and versatility of the proposed framework. These scenarios include navigation under rigid formation constraints, deformation through controlled scaling and rotation, and seamless shape morphing between distinct geometric configurations. The results demonstrate the framework's ability to guarantee safety, preserve formation coherence, and maintain precise coordination even in complex and dynamic environments. The chapter concludes by synthesizing the main findings, discussing inherent limitations, and outlining promising directions for future research.

Chapter 1

Introduction to Multi-Robot Systems and Coordination Approaches

Multi-robot systems (MRS) have emerged as a powerful solution for solving complex problems that cannot be efficiently addressed by single robot. These systems consist of multiple autonomous robots working collaboratively to accomplish a common task. This chapter explores the fundamentals of multi-robot systems, their coordination approaches, key coordination challenges, and practical application examples.

1.1 Multi-Robot Systems

A multi-robot system (MRS) consists of multiple autonomous robots that share the common goal of accomplishing a given task. Each robot in the system can perceive its environment and make decisions based on its own state and interactions with other robots in the group. The Components of a MRS are: robots, communication systems and coordination Algorithms. The classification of multi-robot systems is illustrated in figure 1.1.

1.1.1 Robots

Each robot in an MRS can be a drone, a ground robot, or an underwater robot, among others. These robots are equipped with sensors to perceive their environment and actuators to move.

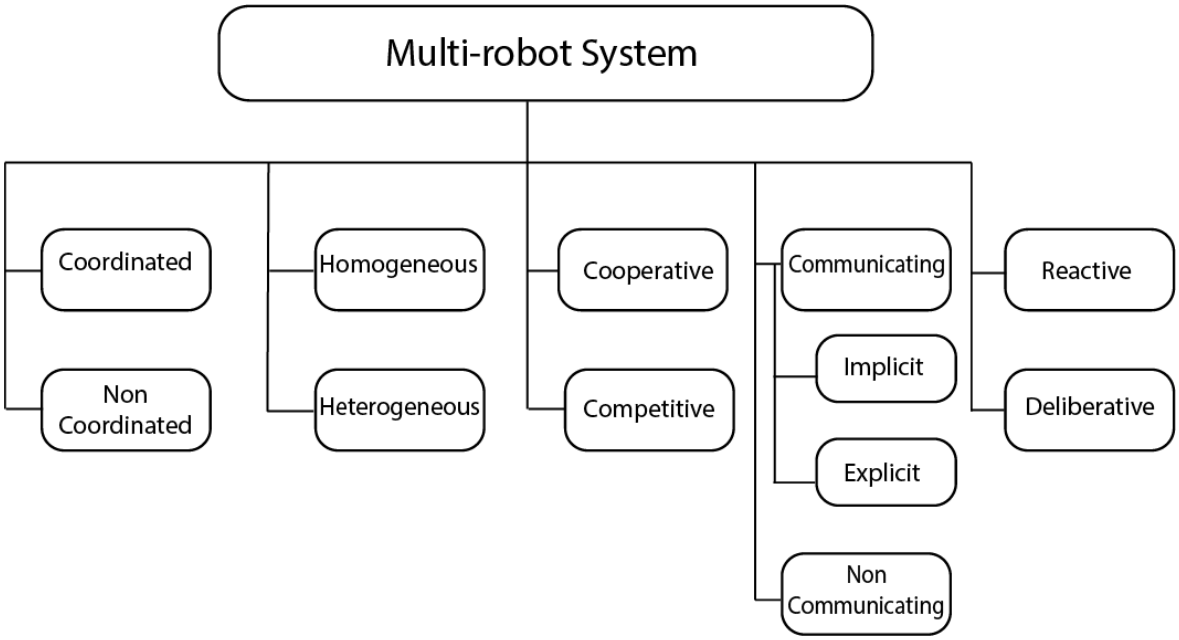

Figure 1.1: Classification of Multi-Robot Systems[18]

Figure 1.2: Drone

Figure 1.3: Ground Robot

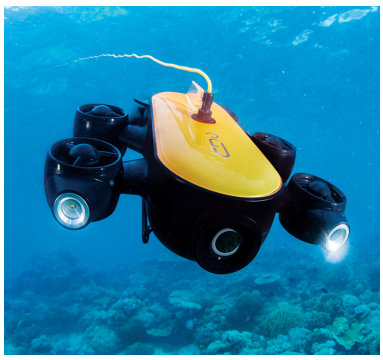

Figure 1.4: Underwater Robot

1.1.2 Communication Systems

Communication is a crucial component of multi-robot systems (MRS), facilitating the coordination and synchronization of robotic tasks. Robots must communicate their status, environment, and tasks to collaborate effectively without causing interruption or redundancy. This communication allows each robot to enhance its knowledge base with information from others, facilitating the coordination of operations such as path planning, resource allocation, and obstacle avoidance. The efficacy of this interchange directly influences the system's overall efficiency, as seamless communication enables robots to adapt to changing scenarios and execute real-time modifications to their operations.

Diverse communication protocols are used to enable information exchange between robots. These may include local communication systems, such as infrared or Bluetooth, for robots in close proximity, or global communication networks, such as Wi-Fi

or 5G, enabling robots to communicate over extensive distances or via a central server. The selection of communication mechanism is determined by factors such as the operating environment, data transfer needs, and the number of robots involved. Effective communication protocols are crucial for sustaining real-time coordination, ensuring that robots remain synchronized and can reliably and efficiently achieve the collective objective.

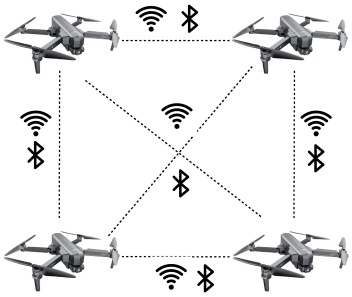

Figure 1.5: Swarm of Drones

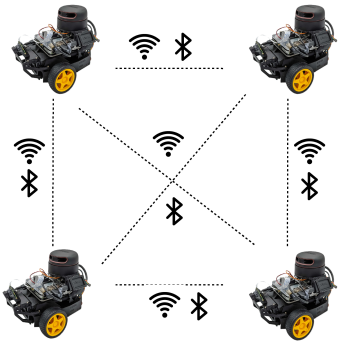

Figure 1.6: Swarm of Ground Robots

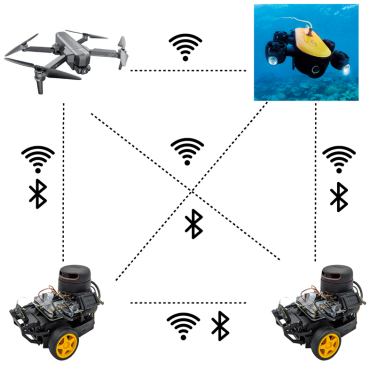

Figure 1.7: Heterogeneous Swarm of Robots

1.2 Classical Coordination Approaches

Multi-robot systems rely on several coordination approaches to ensure effective collaboration among robots. Coordination algorithms are essential for the effective functioning of multi-robot systems (MRS), as they define the interactions and collaborations between robots to achieve a common goal. These algorithms ensure that each robot operates in synchrony with others, effectively distributing tasks while preventing redundancy or interference in their operations. In the absence of effective coordination, robots may execute tasks autonomously, leading in inefficiencies, collisions, or incomplete task execution.

The principal objective of coordination algorithms is to facilitate autonomous robot operations while ensuring a cohesive strategy throughout the system. This includes various tasks, such as path planning, resource allocation, task distribution, and decision-making. These algorithms must adapt to dynamic environments, handle uncertainties, and adjust to real-time changes in the system or environment, depending on the application.

Coordination methods in multi-robot systems (MRS) are primarily classified into two categories: *centralized and distributed (decentralized) coordination.* These methodologies characterize the processes by which robots make decisions and interact with one

another to accomplish a common objective. Centralized coordination relies on a central authority or leader for decision-making, whereas decentralized coordination enables robots to function autonomously and make localized decisions informed by shared information. Each technique possesses distinct advantages and limitations that affect the design and performance of the relevant coordination algorithms.

1.2.1 Centralized Coordination

Centralized coordination requires the presence of a central controller, or leader robot, that supervises the actions of all robots within the system. This method involves the central unit receiving information from all robots, analyze it, and sending commands or instructions to them based on a detailed understanding of the task at hand. The central unit possesses comprehensive information regarding the environment and the states of the robots, enabling it to make optimal decisions for the overall system. This method tends to be more straightforward to execute, and may result in improved overall optimization due to centralized control.

Centralized coordination presents several significant advantages, primarily derived from the centralized control provided by a single unit or controller. A key characteristic is global decision-making, where the central controller has access to all pertinent data from each robot in the system. This enables intelligent decision-making that enhances the system's overall performance, assuring effective resource utilization and synchronized robotic movements. In addition, task allocation in centralized systems is highly efficient, as the central controller can designate specific tasks to each robot based on the system's objectives and a comprehensive understanding of the environment. This is especially advantageous for intricate tasks requiring precise coordination, such as path planning and resource allocation. The centralization of decision-making enhances efficiency, as the central unit can precisely organize and allocate tasks, reducing redundancy and minimizing operational errors. Nonetheless, a notable disadvantage of centralized coordination is the bottleneck risk: when the number of robots in the system increases the central controller may find it challenging to oversee the increasing complexity, which could result in delays or inefficiencies within the system. Moreover, if the central unit malfunctions, the entire system faces potential collapse, as all robots depend on it for directives, rendering centralized coordination susceptible to singular points of failure.

The following examples explain the applications and benefits of centralized coordination algorithms in multi-robot systems.

1. **Centralized Task Allocation:**In centralized task allocation, a central controller or system receives a list of tasks and allocates them to individual robots to optimize a global objective, such as minimizing completion time, managing battery charge, or increasing efficiency [19]. This approach is widely used in scenarios where precise coordination is required among robots to complete a task effectively. In large-scale warehouse environments, centralized task allocation is essential for managing robotic tasks. For example, Amazon Robotics uses a central controller that efficiently distributes assignments such as picking, sorting, and transporting items to robots in real time. This coordination ensures that robots avoid collisions and that tasks are allocated efficiently according to the current status of the warehouse. The central system continuously updates task assignments, enhancing throughput and reducing idle times, thereby improving the overall efficiency of the warehouse. The following diagram illustrates the main components and workflow of the centralized task allocation process.

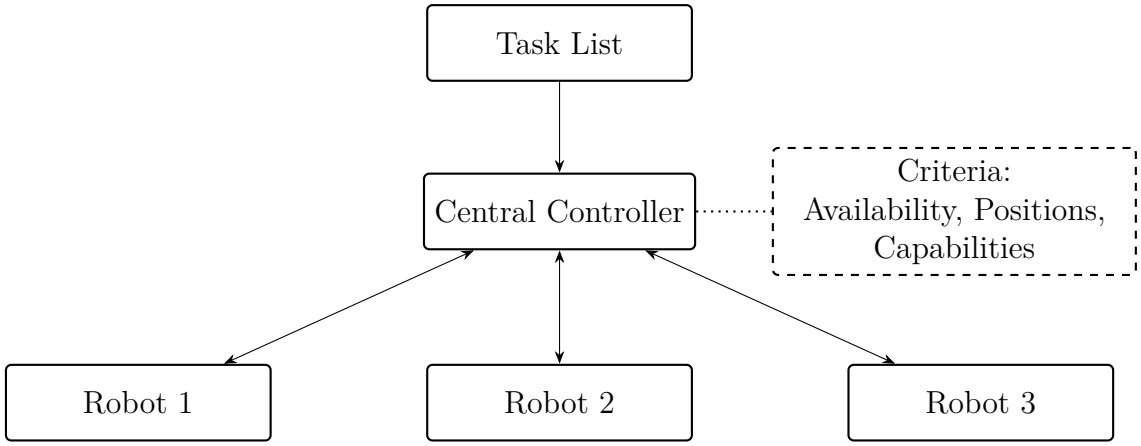

Figure 1.8: Centralized task allocation

Figure 1.9: Amazon robotics

2. **Global Path Planning :**Global path planning in multi-robot systems (MRS) aims to determine optimal trajectories for all robots to achieve objectives while minimizing conflicts, energy consumption, and time [20]. The A* algorithm [21] is one of the most well-known and widely used algorithms for global path planning,

particularly in grid-based environments. It operates by searching for the shortest path between a robot's current position and its goal, taking into account both the cost to reach the goal and the cost from the current position. The algorithm uses a heuristic function, typically the Euclidean distance, to estimate the cost of reaching the goal from a given point. A* combines Dijkstra's algorithm, which guarantees the shortest path, with heuristic functions that guide the search in a more efficient manner.

In multi-robot systems, A* algorithm can be extended to handle the coordination between robots by considering not only the obstacles in the environment but also the positions of other robots. The central controller can adapt the A* algorithm to avoid potential collisions and ensures that robots do not block each other's paths. For example, the algorithm can dynamically update a robot's path as other robots move or if the environment changes, allowing for real-time path adjustment.

Artificial potential field methods [22] offer an alternative approach to global path planning, relying on artificial 'forces' to guide the robots along their paths. In this method, the robot is considered to be under the influence of attractive forces toward its goal and repulsive forces from obstacles or other robots. The goal attracts the robot, while obstacles and robots create repulsive fields, guiding the robot to avoid collisions. The potential field approach is advantageous in terms of its simplicity and ability to quickly calculate a feasible path. However, it can suffer from local minima, where the robot becomes stuck in a position where the attractive and repulsive forces balance each other out, preventing further progress. To overcome this issue, various enhancements have been proposed, such as the use of virtual forces or dynamic adjustments to the potential fields based on the environment and robot behaviors. This approach can be applied to plan paths for all robots simultaneously, with each robot's repulsive field influenced by the positions of other robots. However, the interaction between robots' fields can lead to conflicts, especially in high-density environments. Centralized coordination is often used to manage these interactions and resolve conflicts by adjusting the robots' velocities or planning new paths if necessary.

While both A* and potential fields are widely used, several challenges remain in the application of these algorithms to multi-robot systems. One of the major challenges is collision avoidance, as the robots must not only avoid obstacles but also other robots. This issue is particularly pronounced when robots are operating in close proximity, where the repulsive forces from potential fields may not be sufficient to avoid collisions, or when the computational cost of real-time A* path recalculation becomes prohibitive. Another challenge is the scalability of the path planning algorithms as the number of robots increases. As the number of robots

grows, the interaction and coordination between their paths become more complex, leading to higher computational requirements. Advanced algorithms, such as decentralized approaches or multi-objective optimization methods, can address some of these challenges by reducing the dependency on centralized control and enabling more efficient path planning in large-scale systems. The illustration below shows the different path planning algorithms employed in multi-robot systems.

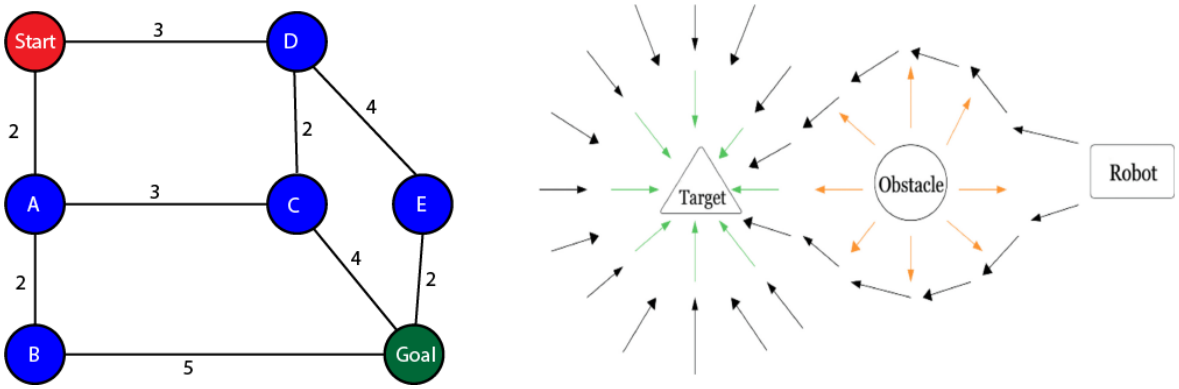

Figure 1.10: Graph Example and Artificial Potential Fields[20]

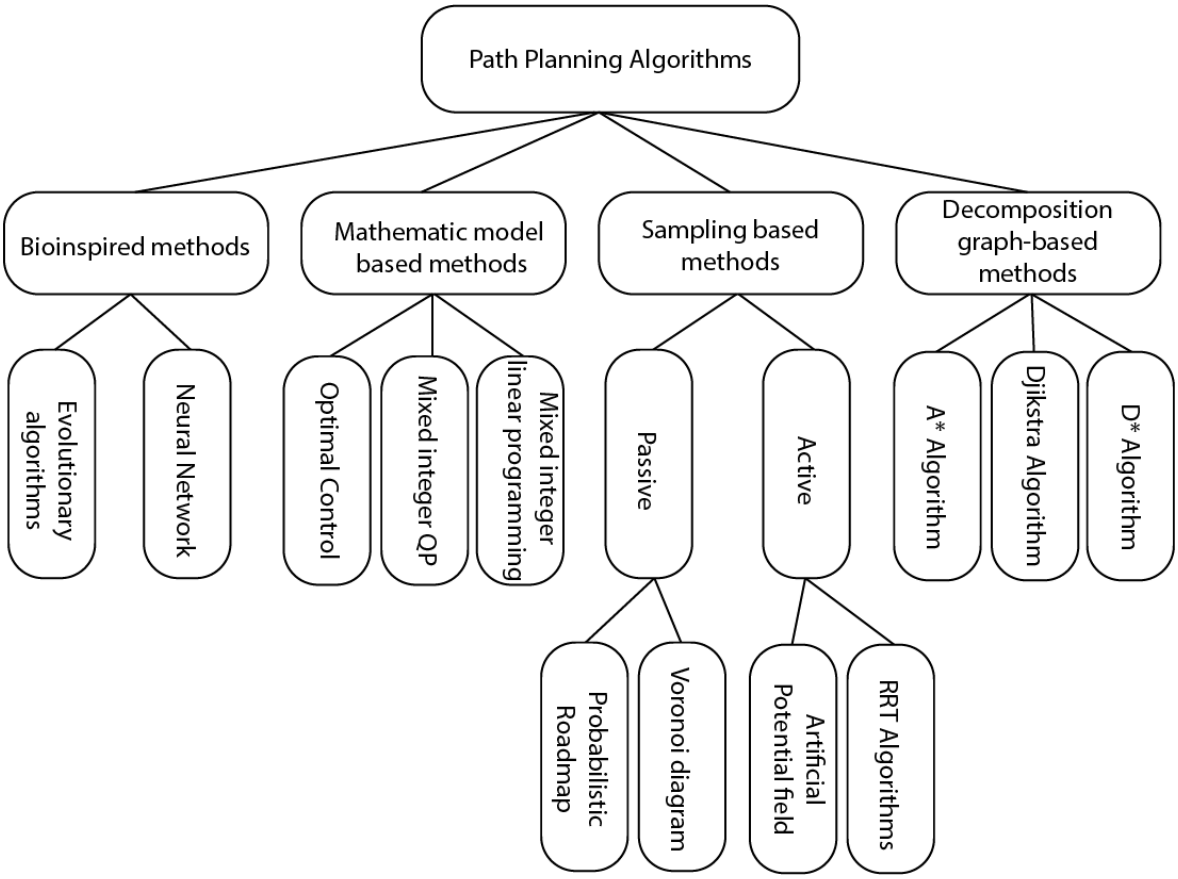

Figure 1.11: Path planning Algorithms[20]

The table below lists the path planning approaches that have been discussed and

presented, along with their advantages and limitations.

3. **Centralized Resource Management:** Centralized resource management [23] plays a pivotal role in coordinating multi-robot systems, especially when robots rely on limited resources such as battery power, computational capacity, or sensor data. In this approach, the central controller allocates resources dynamically based on real-time data, optimizing the overall efficiency of the system. For example, in an industrial environment, the central controller might assign charging stations to robots depending on their current battery levels or prioritize specific robots for critical tasks. By managing resources effectively, the system ensures that each robot operates at its peak performance, preventing resource overload or underutilization. This enhances the overall system's efficiency and contributes to the robots' longevity and reliability, ensuring they can continue performing their tasks without unnecessary delays or failures.

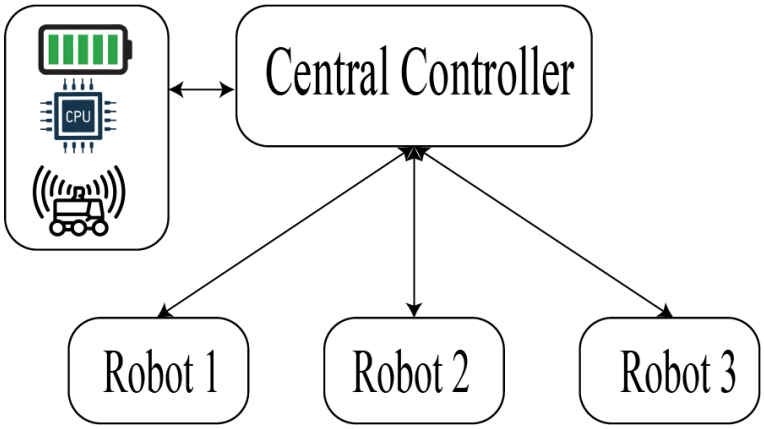

Figure 1.12: Centralized Resource Management

Centralized coordination can be highly effective in small-scale systems or for specific tasks, but it generally fails to scale to larger, more complex scenarios. Relying on a single central unit may hinder the system's ability to manage a large number of robots or to operate in environments with limited communication.

1.2.2 Distributed Coordination

Distributed coordination is crucial in multi-robot systems (MRS), where robots make decisions based on local information and interactions with their neighbors. Unlike centralized systems, which depend on a central controller, distributed systems allow robots to coordinate through direct communication and local decision-making. This approach offers enhanced scalability, fault tolerance, and flexibility. Four key coordination methods are Leader-Follower, Consensus-Based, Formation-Based, and Task Allocation.

Approach	Advantages	Limitations
Evolutionary Algorithms	Highly adaptable; capable of handling complex, dynamic environments; no need for gradient information.	Computationally expensive; may converge to local optima; performance depends on parameters.
Neural Networks	Can handle high-dimensional spaces and learn complex patterns; useful in environments with large variables.	Require significant training data and resources; prone to overfitting and poor generalization.
Optimal Control	Provides a rigorous framework for optimal path finding; guarantees the best path under ideal conditions.	Computationally expensive; requires detailed system modeling.
Mixed Integer QP	Effective for finding optimal solutions for path planning with discrete and continuous variables.	Solution time grows exponentially with problem size; less scalable for large systems.
Mixed Integer Linear Programming	Efficient and optimal for problems with linear constraints and objectives.	Cannot handle non-linear constraints effectively; limited to simpler problems.
Probabilistic Roadmap	Effective for high-dimensional spaces; handles obstacles well.	Requires many samples for completeness; less efficient in dynamic environments.
Voronoi Diagram	Efficient for path planning in environments with complex obstacles; prevents robot collisions.	Requires extensive preprocessing; less adaptable to dynamic environments
Artificial Potential Fields	Easy to implement and understand Applicable for real-time computation	Local minima; Can cause oscillations in narrow passages; Not suitable for large scale environment.
RRT Algorithms	Flexible and can handle complex, dynamic environments.	May not produce optimal paths; computationally intensive in high-dimensional spaces.
A* Algorithm	Efficient and widely used; guarantees the shortest path in grid-based environments.	Computationally expensive in large-scale environments; limited to grid-based problems.
Dijkstra Algorithm	Guarantees the shortest path; suitable for static environments.	Less efficient than A for large, complex environments.
D* Algorithm	Designed for dynamic environments; allows real-time replanning.	Computationally intensive; requires replanning as the environment changes.

Table 1.1: Advantages and Limitations of Path Planning Algorithms

1. **Leader-Follower Approach:** The leader-follower approach ([6], [24], [25], [26], [27]) is a widely used method in distributed multi-robot systems, where one robot (the leader) assumes the responsibility of guiding the rest of the robots (followers). This approach simplifies coordination by designating a single robot to make key decisions, such as choosing a path or determining the next task. The followers replicate the leader's behavior, ensuring that the robots move in a coordinated manner without interference. While this approach can be considered a form of centralized coordination, it can be adapted to a distributed system by allowing dynamic leader selection based on task requirements or environmental conditions. In distributed systems, the leader's role is often assigned based on robot capabilities, environmental context, or task-specific needs. For example, in a search-and-rescue mission, the robot with the most advanced sensors might serve as the leader. One of the key challenges of the leader-follower model is maintaining robustness when the leader robot fails or is obstructed. In such cases, leadership can be reassigned to another robot, ensuring the continuity of the mission. This approach is particularly useful in tasks requiring a well-defined sequence of operations, such as exploration or surveillance.

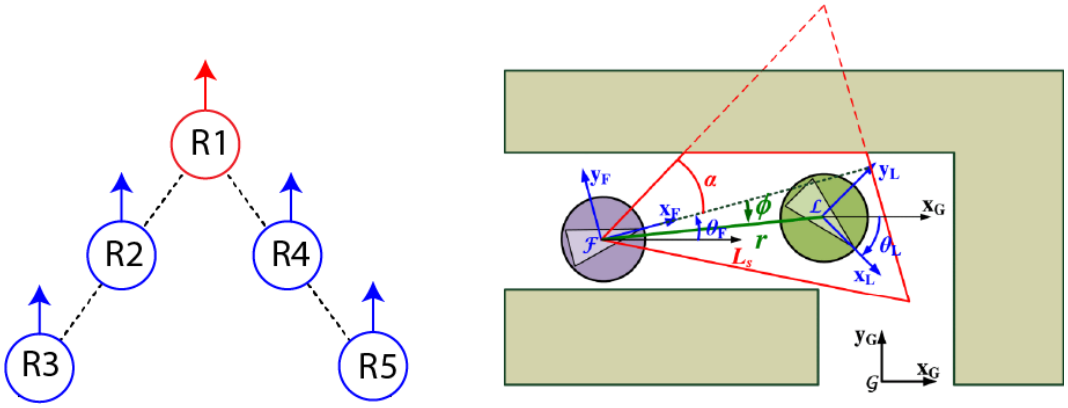

Figure 1.13: Leader-follower examples, Leader-follower in space with obstacles[6].

2. **Consensus-based Approach:** ([8], [9], [10], [28]) The consensus-based approach is crucial to distributed coordination, particularly when robots must achieve a common goal (such as direction, velocity, and other objectives) using only local information. In this technique, each robot communicates with its neighbors and adjusts its behavior based on the information it receives, with the goal of achieving global consensus. This strategy uses of gossip algorithms or average consensus, which allow robots to gradually agree on values like position, velocity, and task assignment. Consensus algorithms perform especially well in dynamic environments where centralized control is unfeasible or undesirable. The fundamental advantage of consensus-based systems is their resilience to failures, enabling the system to continue operating even if certain robots fail or

are disconnected. The goal is to ensure that the robots can reach an agreement with only limited local information. This method is highly scalable, making it useful in many multi-robot systems.

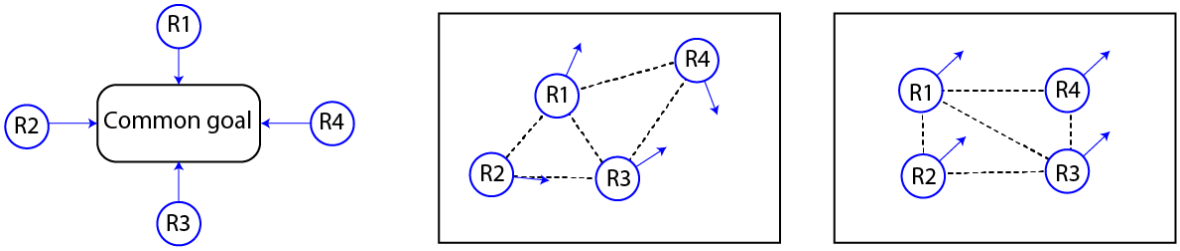

Figure 1.14: Consensus-based Approach Example.

3. **Formation-based Approach:**([29],[30],[31],[4], [7],[32]) The formation-based approach is important for maintaining a geometric structure among robots while completing a task. This approach is often used in applications such as surveillance, mapping, and environmental monitoring, where robots must traverse in specified patterns while maintaining specific relative positions. Formation-based coordination is achieved through local interactions between robots that alter their positions relative to one another while adhering to particular geometric constraints.

   A key challenge in formation-based methods is ensuring that robots maintain their formation when navigating obstacles or operating in restricted environments. Advanced algorithms are employed to determine optimal robot placements within the formation, ensuring stability and collision avoidance. Additionally, robots must be capable of adjusting their positions in real-time to preserve the formation while navigating through complex conditions. These challenges are typically addressed using virtual forces or control laws to maintain proper inter-robot distances.

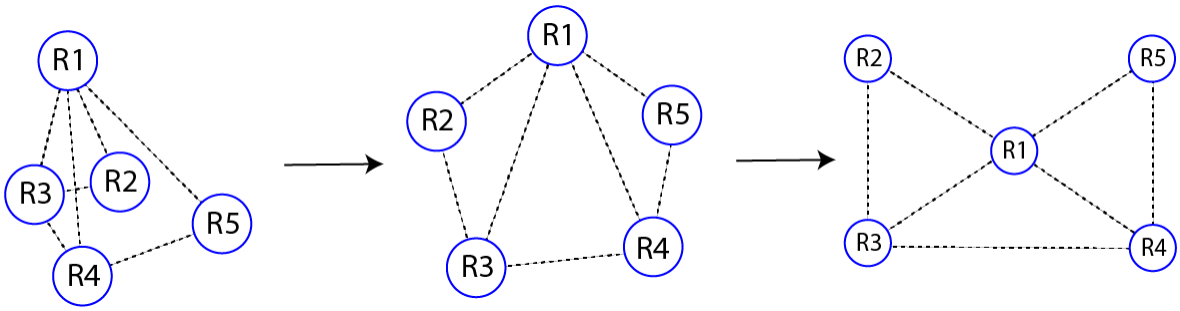

Figure 1.15: Example of formation with 5 robots: random positions, circular shape, rectangular shape.

4. **Task Allocation Approach:**([19], [33], [34], [35])The task Allocation Approach in decentralized systems seeks to effectively assign tasks to robots according to

their specific capabilities, availability, and the overarching requirements of the system. In contrast to centralized systems, where a central controller allocates tasks, decentralized task allocation algorithms enable robots to autonomously assign tasks through local interactions. This methodology promotes enhanced autonomy, scalability, and adaptability, making it especially effective in dynamic settings where task priorities and robot availability can change in real time.

Decentralized assignment of tasks is essential in situations requiring coordination between different robots, including search-and-rescue missions, environmental monitoring, or complex industrial procedures. In these environments, robots must cooperate effectively, exploiting local information to ensure that appropriate tasks are allocated to the most competent robots at the optimal time. Many techniques for decentralized task allocation have been developed, including auction-based systems, market-based models, and optimization-based algorithms. These models allow robots to negotiate and allocate tasks autonomously, leveraging real-time data to improve resource efficiency and reduce task completion times. Using these algorithms enables robots to function more efficiently, ensuring that necessary tasks are completed without overloading any robot. A significant challenge in decentralized task allocation is ensuring equitable and efficient task distribution. Given each robot's unique talents and resources, it is critical to ensure a fair allocation of responsibilities while improving the overall performance of the system. Achieving this equilibrium is difficult, especially in large systems with multiple robots, where the likelihood of some robots being underutilized or overtaxed increases. Despite these challenges, the decentralized architecture holds great potential for practical applications. By encouraging autonomy, flexibility, and adaptation, it enables multi-robot systems to perform complex tasks more successfully, particularly in dynamic and unpredictable environments.

Approach	**Advantages**	**Limitations**
Leader-Follower	Simplicity, centralized control	Dependence on the leader, vulnerability
Consensus-based	Autonomy, resilience	Complexity of consensus algorithms
Formation-based	Maintains configuration	Less flexibility in the presence of disturbances
Task Allocation	Efficiency, optimal resource management	Issues with dynamic task distribution

Table 1.2: Comparison of Coordination Approaches in Multi-Robot Systems

1.3 Coordination Problems

Despite considerable advances in multi-robot systems, significant coordination challenges persist, reducing their full potential. These challenges result from the complex interactions among robots, the dynamic nature of their environments, and the need for effective communication and decision-making, each of which is crucial for the successful execution of collective tasks.

1.3.1 Real-Time Synchronization

A key coordination challenge in multi-robot systems (MRS) is the requirement for real-time synchronization. When several autonomous robots collaborate towards a common goal, it is essential to maintain synchronization in their activities for effective performance. Real-time synchronization allows robots to synchronize their motions and decision-making processes instantaneously, which is especially important in dynamic environments requiring immediate responses. However, achieving synchronization is not without its challenges. Latency and network delays can significantly impact the system's ability to synchronize actions across all robots. For instance, if one robot receives an outdated update or fails to communicate promptly, the entire system may operate asynchronously, leading to inefficiencies or even system failures. This issue is particularly pronounced in large-scale systems where coordinating a larger number of robots increases the complexity of maintaining synchronization.

Another challenge arises from the heterogeneity of robots in a multi-robot system. Robots may have varying levels of computational power, communication range, and sensor accuracy, which can cause discrepancies in how they perceive the environment or execute tasks. These differences complicate the synchronization of actions, especially when robots need to work together in close coordination. Additionally, in environments with variable communication quality or unreliable networks, robots may not always be able to communicate in real-time, further exacerbating synchronization problems. These situations are particularly challenging in dynamic settings where environmental conditions can rapidly change, demanding immediate adjustments in robot actions.

1.3.2 Partial and Incomplete Communication

Another significant coordination problem in multi-robot systems (MRS) arises from partial and incomplete communication among robots. In decentralized systems, where robots operate autonomously, communication is often limited to local interactions between neighboring robots and may not always provide a complete picture of the entire system's state. This limitation poses a critical challenge, as each robot's ability to make informed decisions depends on the quality and completeness of the information

it receives. In many real-world scenarios, robots may face communication delays or signal interference, resulting in the loss or distortion of data. For instance, in large-scale environments or remote locations with poor connectivity, robots may struggle to communicate in real-time, leading to delayed or missing updates. This issue can disrupt the coordination process, as robots may not have access to timely information about the tasks or environment, potentially leading to inefficient execution or even conflicts between robots.

The issue of partial information is particularly problematic when robots are required to collaborate on complex tasks, such as path planning or resource allocation. In these cases, missing or outdated data can lead to misunderstandings between robots, resulting in redundant actions, inefficient resource usage, or collisions. For example, if a robot is unaware of a nearby robot's position due to incomplete communication, it may make decisions that interfere with the other robot's trajectory, leading to unnecessary delays or errors. Furthermore, as the number of robots in a system increases, the complexity of communication grows exponentially. Ensuring that each robot receives the necessary information from all relevant sources without overwhelming the communication network becomes increasingly difficult. The more robots involved, the higher the likelihood that some information will be missed or misinterpreted, especially in dynamic environments where data is constantly changing.

1.3.3 Direct Robot Interactions

A significant coordination problem in multi-robot systems (MRS) occurs during direct robot interactions, especially when robots need to physically collaborate on tasks such as object manipulation, lifting, or navigating through constrained spaces. These tasks often require precise and synchronized actions, where even small deviations or misalignments can result in failures or inefficiencies. In such scenarios, collisions or interference between robots become a significant concern. As robots interact physically, the potential for accidents increases, especially when they need to operate in tight or shared spaces. For instance, if two robots are tasked with lifting a heavy object together, even slight miscalculations or lack of coordination can lead to the object being dropped or damaged, wasting both time and resources.

Additionally, maintaining a stable formation or configuration becomes a challenge when robots need to stay in relative positions to each other. In environments where obstacles are present or when robots must adapt to dynamic conditions, such as navigating through cluttered areas or adjusting to changing environmental factors, maintaining the correct formation while avoiding collisions becomes increasingly complex.

Furthermore, the physical capabilities of the robots themselves (such as manipulation precision, speed, and force control) may vary, leading to discrepancies in how tasks are

performed. These differences can create coordination breakdowns, where some robots may move faster or slower than others or fail to follow instructions accurately, affecting the overall performance of the task.

1.3.4 Redundancy in Robot Roles

Another significant coordination problem in multi-robot systems (MRS) arises from redundancy in robot roles, where multiple robots may be capable of performing the same task, leading to potential conflicts or inefficiencies. This issue is particularly prevalent in large-scale systems or when robots are tasked with solving complex problems that require multiple resources. Redundancy often occurs when robots have overlapping capabilities, such as two or more robots being able to perform identical actions like path following or object manipulation. While having multiple robots capable of the same task can be beneficial for ensuring system robustness and fault tolerance, it can also lead to unnecessary duplication of efforts. For example, if two robots are performing the same task simultaneously, such as picking up the same object, it can result in wasted time, energy, and computational resources, ultimately reducing the system's overall efficiency.

Moreover, when redundancy is not managed effectively, it can create coordination conflicts. Robots may unknowingly compete for the same resources or space, leading to collisions, task duplication, or unnecessary interference. For instance, in environments where robots need to navigate through narrow corridors or limited spaces, redundant movement paths can result in congestion and delay in completing the task. Additionally, managing redundancy often involves determining how to best distribute tasks among robots to avoid overloading certain units while underutilizing others. Efficient task allocation becomes crucial to balance workload distribution, ensuring that all robots contribute effectively to the task at hand without overlap.

1.3.5 Decision-Making Conflicts

In multi-robot systems (MRS), decision-making conflicts arise when robots, operating autonomously, make independent decisions that are not coordinated with the actions of other robots. This issue is especially prominent in decentralized systems, where robots are tasked with executing actions based on local information rather than relying on a central controller. While decentralization offers advantages in scalability and fault tolerance, it also introduces the possibility of conflicts between robots when their actions overlap or contradict each other. These conflicts typically emerge when multiple robots make decisions that affect the same task or shared resource. For instance, if two robots independently decide to complete the same task, such as moving an object or choosing a path, this can lead to resource contention or task duplication, resulting in wasted

effort, time, and energy. In some cases, robots might also inadvertently obstruct one another or create a bottleneck, slowing down the overall task completion.
The lack of global knowledge in decentralized decision-making systems exacerbates these conflicts. Since robots only have access to local information about their environment and their own state, they may be unaware of the actions or intentions of other robots nearby. This lack of shared awareness can lead to misaligned priorities and inefficient task execution. For example, if a robot moves toward an area that another robot is already planning to navigate, both robots could end up in the same location at the same time, causing delays or errors in their actions. Furthermore, when robots are faced with complex, time-sensitive tasks that require prioritization or negotiation for resources, conflicts in decision-making can arise from differing task prioritization schemes or conflicting objectives. Some robots might prioritize speed, while others may focus on accuracy, leading to suboptimal coordination and performance.

1.3.6 Robot Heterogeneity and Coordination

A significant coordination problem in multi-robot systems (MRS) arises from the heterogeneity of the robots involved. In many systems, robots possess different capabilities, such as variations in sensor types, processing power, mobility, or specialized tools. While heterogeneity can be advantageous in certain scenarios, it introduces several challenges in terms of coordination and task allocation. One of the primary issues is the mismatch in capabilities between robots. Robots with varying computational powers, sensor accuracies, or actuation strengths may struggle to coordinate effectively when performing tasks that require uniform performance across all units. For example, a robot with advanced sensors may perform tasks that require high precision, while another robot with limited sensing capabilities may struggle to contribute effectively to the same task. Such differences can lead to uneven workload distribution, where some robots are overburdened while others remain underutilized.
Moreover, the interaction between heterogeneous robots can be complex. Robots with different movement abilities, such as ground robots and aerial drones, may need to collaborate in environments that require precise spatial coordination. Differences in speed, navigation constraints, or environmental adaptation can hinder their ability to perform joint tasks efficiently. For instance, aerial drones might be much faster in certain environments but could face difficulties working in confined spaces alongside slower, ground-based robots. Additionally, managing interoperability among heterogeneous robots introduces challenges in communication and task delegation. Robots with different communication protocols or data formats might struggle to exchange information effectively, leading to inefficiencies or even system breakdowns. The need for adaptable coordination algorithms that can accommodate the diverse capabilities

of different robots becomes critical in these cases.

1.3.7 Scalability and Coordination in Large-Scale Systems

One of the most pressing coordination problems in multi-robot systems (MRS) is scalability, particularly when the system consists of a large number of robots. As the number of robots in a system increases, so does the complexity of coordinating their actions, ensuring efficient communication, and managing resources. The challenge of scalability is compounded by the fact that, as more robots are added, the interactions and dependencies between robots grow exponentially, making the system more difficult to manage and less efficient. In large-scale systems, maintaining real-time coordination becomes a significant issue. As the size of the system increases, communication overheads rise, leading to delays in data exchange and slower decision-making. This is particularly problematic in dynamic environments where quick adjustments are necessary to handle changes such as obstacles, mission updates, or changes in the environment. The more robots involved, the more challenging it becomes to ensure that all robots are synchronized and working toward a common goal without causing conflicts or interference between them.
Furthermore, centralized coordination approaches, while effective for small systems, often face scalability issues in large-scale systems due to the increasing amount of data that needs to be processed and the risk of bottlenecks at the central controller. In decentralized systems, where robots make independent decisions based on local information, the challenge lies in ensuring that the robots are still able to cooperate effectively without overwhelming the communication network or causing conflicting decisions. Another issue is resource management in large systems, where the allocation of tasks and resources such as energy, sensors, or processing power must be carefully balanced to avoid overloading certain robots while others remain underutilized. As the number of robots increases, task allocation algorithms need to become more sophisticated to handle the growing complexity and ensure that tasks are distributed efficiently across the system.

In summary, as the number of robots in a multi-robot system increases, the challenges of coordination grow significantly. Efficient communication, synchronization, task allocation, and resource management become progressively harder to maintain, particularly in dynamic or large-scale environments. Addressing these issues requires the development of advanced algorithms and strategies that can scale effectively while ensuring that all robots work cohesively toward a common goal.

1.4 Practical Application Examples

Recent improvements in robotics, artificial intelligence, and coordination algorithms have greatly increased the number of feasible uses for multi-robot systems (MRS).These systems operate most effectively when robots are collaborate strongly , be flexible, and possess scalability. This section discusses key applications of MRS, including industry, space exploration, search and rescue missions, agriculture, and environmental monitoring. Managing resources like energy or sensors is crucial for robots to be able to work on long missions without needing assistance.

1.4.1 Industrial Applications

Industrial automation is a prominent area where multi-robot systems are demonstrating their capabilities. The capacity to deploy teams of robots operating independently or semi-autonomously to execute tasks in warehouses, factories, or distribution centers has transformed logistics and supply chain management. The essential elements of industrial applications for MRS include:

Warehouse Automation: In scenarios such as Amazon's fulfillment centers, autonomous mobile robots (AMRs) cooperate to deliver articles, control inventory, and maintain effective stock management. These robots must collaborate to prevent collisions, enhance their paths for energy efficiency, and guarantee timely completion of tasks.

Collaborative Manufacturing: MRS can be implemented in manufacturing facilities for product assembly or inspection tasks. Robots can collaborate to lift large objects, assemble components, and conduct quality assurance inspections. They can also adjust to changes in manufacturing schedules or product designs autonomously.

Fleet Coordination for Material Handling: In complex manufacturing operations, MRS can facilitate the transportation of raw materials, components, or completed products throughout several production steps. Coordination algorithms enable a fleet of robots to collaborate by distributing tasks and preventing congestion in paths.

The principal challenges facing industrial multi-robot systems (MRS) center on real-time coordination, scalability, and safety. Effective scheduling along with task distribution is essential for preventing delays and maximizing throughput in dynamic industrial settings. As industrial environments expand in scale and complexity, MRS must adapt smoothly to handle extensive teams of robots, each exhibiting differing levels of autonomy, while preserving operational efficiency. The integration of robots with human workers raises safety concerns, requiring the establishment of stringent safety measures to reduce the risk of accidents and maintain a secure working environment for all parties involved.

1.4.2 Space Exploration

Multi-robot systems have crucial applications in space research, where human involvement is sometimes restricted due to vast distances and dangerous conditions. These systems facilitate the exploration of planetary surfaces, the execution of scientific investigations, and the support of prospective manned missions.

Planetary Exploration: Planetary exploration involves the collaborative operation of rovers and drones to investigate areas such as Mars. These robots collect data, analyze soil and atmospheric samples, and chart the topography. In many instances, multiple robots are dispatched to the same site, operating autonomously to address different parts of a scientific mission.

Swarm Robotics for Exploration: A team of robots can be deployed to explore unknown or dangerous areas. These robots cooperate, sharing information and adaptively adjusting their actions to ensure that the exploration is thorough and safe. Their ability to swiftly adapt to new information makes them highly effective for tasks such as locating ice deposits or identifying geological structures.

Robotic Assistants for Manned Missions: Multi-robot systems may also contribute to future manned missions. Robots might help astronauts in maintaining the operations of space stations or lunar bases by performing routine maintenance, establishing infrastructure, and contributing to scientific research.

In the domain of space exploration, multi-robot systems (MRS) address numerous significant challenges, such as communication delay, severe climates, and resource distribution. Given the considerable distances separating Earth from space missions, communication between robots and Earth may encounter significant delays, rendering autonomous decision-making and local collaboration among robots crucial for mission success. Moreover, robots must be constructed to withstand extreme temperatures, radiation, and variable terrain conditions, all of which provide considerable hurdles to their operation and resilience in space. Additionally , as these robots are expected to function autonomously for extended durations, efficient resource management, especially regarding energy, is crucial to ensure their continued autonomy and effectiveness in space exploration missions.

1.4.3 Search and Rescue Operations

Search and rescue is one of the most important applications for multi-robot systems, especially in dangerous or inaccessible environments. MRS can collaborate to swiftly find and assist people who remain trapped in areas affected by disasters.

Autonomous Search: Following an emergency such as an earthquake or flood, multi-robot systems can independently navigate through rubble to locate survivors. These

robots can locate individuals via sensors such as sonar or thermal imaging.

Cooperative Navigation: Robots engaged in search and rescue operations must collaborate in their movements, often operating in teams to efficiently cover large areas within a time limit. They must communicate details about the terrain, risks, and previously traversed areas.

Uncertain environments: MRS can be employed hazardous places that are dangerous for human presence, including after chemical spills, fires, or during mining activities. These robots can assess dangers, remove debris, and assist individuals with transportation.

The primary issues with multi robot systems in search and rescue operations include localization and mapping, resilience, and real-time decision-making. In unexpected regions, robots must accurately generate maps and ascertain their precise location to collaborate efficiently. Moreover, these robots must exhibit exceptional reliability, as they frequently function in adverse environments characterized by debris, smoke, and various obstacles that can hinder their effectiveness. The urgency of search and rescue missions requires that robots make prompt decisions based on sensor data, which is frequently insufficient or noisy, complicating the task further. These issues necessitate a synthesis of sophisticated localization technology, resilient robotic designs, and effective decision-making algorithms to ensure the successful completion of these critical missions.

1.4.4 Agricultural Applications

Another area that has developed efficiently is the use of multi-robot systems in agriculture[36], especially for precision farming. These systems can operate together to monitors crops, perform tasks such as planting and harvesting, and optimize resource usage of resources.

Precision Agriculture: MRS can use cameras and sensors to monitor the health of crops and identify areas that need watering, fertilizing, or pest treatment. Robots work together to ensure that each portion of the farm gets the resources it needs in the most effective way possible.

Autonomous Harvesting: Robots can autonomously pick fruits and vegetables on large farms. To cover vast areas quickly without damaging the crops, these robots need to collaborate.

Environmental Monitoring: Robots can collect data on the environment, such as soil moisture and weather conditions, that farmers can use to improve their practices and reduce waste.

Agricultural multi-robot systems encounter challenges distinct from those found in robots utilized in manufacturing or exploration. Adaptability presents a major

hurdle, as agricultural robots must continuously contend with unpredictable weather, varying soil quality, and diverse geography. Agricultural robots must exercise caution while interacting with living organisms, in contrast to industrial robots that often engage with inanimate objects. This ensures that farms and animals remain undamaged during operations. Their design and functionality must be exceptionally accurate and sensitive. Swarm Farm Robotics is a prominent agricultural initiative that employs a consortium of robots collaborating to enhance the efficiency of farming tasks such as crop spraying and seed planting. These robots may operate collaboratively to optimize resource and labor utilization, demonstrating the transformative potential of multi-robot systems in advancing current agricultural practices.

Chapter 2

Mathematical Modeling for Multi-Robot Formation Problem

Formation control is widely recognized as one of the foundational challenges in the control of multi-robot systems, particularly when dealing with mobile robots. In this chapter, we provide a comprehensive exploration of this topic, beginning with an overview of the mechanisms through which formations can be defined and specified. We then proceed to introduce a series of graph-based strategies designed to facilitate formation control. Regardless of the specific characteristics of a given formation task, these problems share a common objective: the coordination of mobile robots to achieve a predefined configuration that adheres to a particular geometric shape or relative state. Moreover, formation control problems typically involve an additional layer of complexity, which is the assignment of distinct roles or responsibilities to individual robots within the formation. To formalize this concept, formation control can be characterized by two primary components: the *shape* or *relative state*, and the *assignment of roles*. The first component specifies the desired arrangement of the robots, essentially defining what the formation should "look like" in terms of its geometry and relative positioning. The second component focuses on assigning specific roles to each robot within the formation, determining which robot assumes responsibility for which part of the configuration. Together, these two components form the core structure of any formation control problem, ensuring that both the physical arrangement of the robots and the allocation of tasks are properly managed effectively to achieve the desired formation.

2.1 Introduction to the Multi-Robot Formation Problem

Multi-robot systems have become a significant area of research due to their applications in a wide range of domains, including autonomous vehicles, drone swarms, and collaborative robots in industrial settings. One of the primary challenges in these systems is the formation control problem, which involves coordinating multiple robots to achieve and maintain a specific geometric configuration. The formation align with a desired shape or relative state, and the robots must perform their tasks while maintaining this formation in dynamic environments.

2.1.1 Mathematical Definition of the Formation Problem

Consider the configuration of a multi-robot system with N robots ($N \geq 2$). The position of each robot i, ($i \in \{1, \cdots, N\}$) in the formation can be represented by the vector $p_i \in \mathbb{R}^n$:

$$p_i = \begin{bmatrix} p_{i,1} \\ \vdots \\ p_{i,n} \end{bmatrix}$$

The goal of formation control is to coordinate these robots in such a way that the relative distances and angles between them remain fixed (or within specified bounds), forming a desired geometric shape. The relative distance between any two robots i and j ($i \neq j$) is given by the Euclidean distance:

$$d_{ij} = \|p_i - p_j\|. \tag{2.1}$$

A formation control problem can be formally described by minimizing the deviation from a set of desired inter-robot distances d_{ij}^{des}. The objective is to ensure that the distance between each pair of robots stays as close as possible to the predefined desired distances, which can be expressed as:

$$\min_{p_i} \sum_{i<j} \left(d_{ij} - d_{ij}^{\text{des}}\right)^2. \tag{2.2}$$

This objective function ensures that the robots remain at the correct relative distances, a fundamental aspect of maintaining a stable formation.

2.1.2 Role Assignment and Formation Topology

Beyond maintaining a specific shape, effective multi-robot formation control requires dynamically assigning roles to each robot. This allocation is critical, as it determines

how each member contributes to the group's collective structure and coordinated movement. In practice, this involves designating which robot acts as the leader, which ones are followers, and defining the interactions needed to preserve the formation's stability. Complementing this is the formation's *topology*, which describes the underlying network of connections between robots, whether through communication links or spatial relationships. This architecture governs how information is shared across the group and profoundly influences the control strategies needed to maintain cohesion. Researchers have developed several common topological models, each with distinct strengths, limitations, and ideal use cases.

Leader-Follower Topology

The leader-follower topology designates a single robot as the formation's guide. The other robots, or followers, then coordinate their movements to maintain predefined positions relative to this leader. This approach is particularly well-suited for applications like convoy driving, where one robot leads the way and the others follow at a fixed distance.

the relative position of robot i (a follower) with respect to the leader L is represented as:

$$p_{iL} = p_i - p_L,$$

where p_i, p_L are the positions of robot i and the leader L respectively.

The goal in this scenario is to control the followers to maintain a fixed formation relative to the leader by adjusting their velocities or accelerations:

$$\dot{p}_i = v_L + v_i,$$

Where v_L is the velocity of the leader, and v_i is the relative velocity of follower i. The followers maintain their relative positions by correcting their movements based on the leader's position.

While this approach simplifies the control scheme by centralizing it around the leader's position, it consequently introduces a critical vulnerability: the entire formation's integrity depends on the leader's reliability. This creates a single point of failure, as the entire formation becomes vulnerable to the leader malfunctioning or deviating from its path.

Distributed Topology

In a distributed topology, no single robot acts as a central leader; instead, all robots are considered equal peers. Each agent maintains the formation by relying exclusively on local information exchanged with its immediate neighbors through pairwise communication. This decentralized architecture enhances both the scalability of the system and its robustness to individual robot failures.

In a distributed system, the position of robot i can be adjusted based on the relative

positions of its neighboring robots $j \in \mathcal{N}_i$, where $\mathcal{N}_i$ represents the set of neighbors of robot i, The control law for each robot can be expressed as:

$$\dot{p}_i = \sum_{j \in \mathcal{N}_i} k_{ij} \left(\|p_i - p_j\| - d_{ij}^{\text{des}} \right) \frac{p_j - p_i}{\|p_i - p_j\|}$$

where k_{ij} is a positive control gain.
$\frac{p_j - p_i}{\|p_i - p_j\|}$ is the unit vector pointing from robot i to robot j, ensuring that the adjustment is directed along the line connecting the two robots.
This control law drives each robot to maintain the desired formation with its neighbors. The summation over all neighbors ensures that robot i adjusts its position simultaneously to satisfy the relative distance constraints with each neighbor.

Switched Topology

Switched topology represents a hybrid control strategy that enables dynamic reconfiguration of the multi-robot system's communication architecture in response to situational demands or environmental changes. This approach proves particularly valuable in scenarios requiring adaptive formation control, such as obstacle avoidance or operation in dynamically evolving environments. In such systems, the underlying communication graph evolves temporally, permitting transitions between distinct topological configurations, for instance, from a leader-follower structure to a distributed or hierarchical architecture based on operational requirements.
Under specific operational conditions, a leader-follower configuration may offer the most effective paradigm for formation management. However, to ensure robustness, the system must be capable of topological adaptation in the event of leader impairment, such as obstacle encounters or operational failures. This fault tolerance is achieved by dynamically switching to a distributed control strategy, wherein the robots employ consensus-based coordination to maintain the formation without dependence on a central controller.
Mathematically, this approach can be modeled by incorporating switching rules into the control laws that determine when the formation should transition between different topologies.
The switching mechanism is formally characterized by a *switching signal* $\sigma(t) \in \Sigma$, where Σ denotes the set of admissible topologies, which determines the active communication graph at each time instant t. Consequently, the control law becomes explicitly dependent on$\sigma(t)$, enabling seamless transitions between distinct control paradigms such as leader-follower and distributed architectures in response to real-time operational conditions:

$$\dot{p}_i = f_{\sigma(t)} \left(p_i, p_{j_{j \in \mathcal{N}_i^{\sigma(t)}}} \right),$$

where $f_{\sigma(t)}(\cdot)$ denotes the topology-specific control policy, and $\mathcal{N}_i^{\sigma(t)}$ represents the neighborhood set of robot i under topology $\sigma(t)$. This formalism provides a mathematical foundation for adaptive formation control, enhancing system resilience and performance in dynamic environments at the expense of increased analytical complexity in stability analysis and controller synthesis.

Hierarchical Topology

This approach to multi-robot formation control organizes the system into a hierarchical structure, effectively dividing it into distinct tiers or subgroups. Within this framework, robots occupying higher levels function as leaders, providing guidance to subordinate subgroups. This hierarchical arrangement facilitates efficient decision-making and robust coordination across extensive systems, thereby achieving an effective synthesis of centralized command and decentralized execution. Such a topology proves especially advantageous in large-scale applications, where a purely decentralized approach may struggle with the prohibitive complexities of communication and coordination.

A hierarchical formation typically employs a tree-like or layered structure for organizing robots. High-level robots function as supervisors to manage the overall formation strategy and issue commands. In contrast, lower-level subordinates are responsible for executing these commands by maintaining their positions relative to their immediate superiors, ensuring local coordination that aggregates into global order.

Mathematically, the control law in a hierarchical system can be expressed similarly to that in distributed control, but with an added layer of abstraction. Robots at higher levels in the hierarchy adjust their positions based on global objectives or formation goals, while robots at lower levels adjust based on local interactions with their immediate supervisors.

The position of robot i at level l relative to its immediate leader j at level $l-1$ can be written as:

$$p_{ij} = p_i - p_j,$$

where p_i and p_j are the positions of robots i and j, respectively. The goal of each robot is to maintain a specific relative position to its leader or supervisor, similar to a leader-follower system. However, each level in the hierarchy can be responsible for a set of robots, allowing for more localized control.

The control law for the hierarchical system can be expressed as:

$$\dot{p}_i = \sum_{j \in \mathcal{N}_i} k_{ij} \left(\|p_i - p_j\| - d_{ij}^{\text{des}} \right) \frac{p_j - p_i}{\|p_i - p_j\|} + \delta \tag{2.3}$$

where δ represents global adjustments made by higher-level robots or supervisors that affect the lower-level robots, ensuring the entire formation moves cohesively. The parameters k_{ij} and d_{ij}^{des} are the same as in the distributed topology, but with the added

consideration of hierarchical influence.

A key advantage of the hierarchical approach is its marked reduction in communication and computational overhead compared to a fully centralized architecture. By constraining interactions to immediate superiors within the tree-like structure, lower-level robots operate with a high degree of locality. This natural restriction in communication pathways directly enhances the system's scalability to large numbers of agents. Nevertheless, this topology introduces specific challenges, particularly concerning the management of dynamic hierarchy transitions and the preservation of global stability should a high-level leader fail.

This architecture is ideally suited for heterogeneous multi-robot systems where agents possess varying capabilities. It enables a rational distribution of tasks: high-level leaders, often equipped with sophisticated sensor suites and greater processing power, can focus on strategic, formation-wide objectives. Simultaneously, subordinate robots, which may be simpler or more specialized, can dedicate their resources to tactical, localized functions such as obstacle avoidance and precise relative positioning, thereby optimizing the system's overall efficiency and robustness.

2.2 Kinematic Modeling for Multi-Robot System

The foundation of any rigorous analysis, control, and coordination strategy for Multi-Robot Systems (MRS) lies in a precise and comprehensive kinematic model. Kinematics, in this context, is the study of motion without considering the forces that cause it. It describes the geometric relationships between the positions, velocities, and orientations of the individual robots within a team and their collective state. For a single robot, kinematic modeling is a well-established discipline [37]. However, in MRS, the complexity increases significantly due to the need to model not only each agent's individual motion but also the interactions and relative motions between them, which are often dictated by task-specific constraints, such as maintaining a formation, manipulating a shared object, or navigating through tight spaces.

This section provides a detailed exposition of the kinematic modeling frameworks for MRS. We begin by delineating the fundamental components of a single-robot kinematic model, which serves as the foundational building block. Subsequently, we explore the two primary paradigms for MRS modeling: the centralized approach, which treats the entire collective as a single composite mechanism, and the decentralized approach, which focuses on the states and interactions of individual agents. A critical part of this discussion involves the modeling of kinematic constraints, both intrinsic (within a single robot) and extrinsic (arising from inter-robot interactions). Finally, we formalize the concept of the system's state space and its evolution, setting the stage for the control and coordination strategies.

2.2.1 The Single-Robot Kinematic Model: A Foundational Building Block

Before aggregating multiple robots into a system, it is imperative to define the kinematic model of a single robot. Consider a mobile robot operating on a flat, two dimensional surface, typically assumed to be frictionless for simplicity.
Its position (configuration) at time t is typically described by a vector $p_i = [x_i, y_i, \theta_i]^\top$, where (x_i, y_i) represents the Cartesian coordinates of a reference point on the robot and θ_i denotes its orientation relative to a global inertial frame.
The kinematic model establishes the relationship between the time derivative of the robot's position (i.e., its velocity) and the applied control inputs. In the case of a differential-drive configuration (figure 2.1), these control inputs correspond to the angular velocities of the left and right wheels, $(w_{L,i}, w_{R,i})$.
The kinematic model is given by:

$$\dot{p}_i = \begin{bmatrix} \dot{x}_i \\ \dot{y}_i \\ \dot{\theta}_i \end{bmatrix} = \begin{bmatrix} \frac{r}{2}\cos\theta_i & \frac{r}{2}\cos\theta_i \\ \frac{r}{2}\sin\theta_i & \frac{r}{2}\sin\theta_i \\ \frac{r}{l} & -\frac{r}{l} \end{bmatrix} \begin{bmatrix} w_{L,i} \\ w_{R,i} \end{bmatrix} \tag{2.4}$$

where r is the wheel radius and l is the distance between the wheels. This model is a classic example of a nonholonomic system due to the constraint that the robot cannot move instantaneously along its axle, a constraint that will be generalized for multi-robot systems (MRS).
To express this model in the standard control form

$$\dot{x} = f(x) + g(x)u \tag{2.5}$$

we define the state vector $x = [x_i, y_i, \theta_i]^\top$ and the control input $u = [w_{L,i}, w_{R,i}]^\top$.
$f(x) = [0, 0, 0]^\top$ is the drift term, which represents the robot's motion without control inputs (in this case, it is zero since there is no autonomous motion without wheel velocities), and $g(x)$ is the control matrix defined as:

$$g(x) = \begin{bmatrix} \frac{r}{2}\cos\theta_i & \frac{r}{2}\cos\theta_i \\ \frac{r}{2}\sin\theta_i & \frac{r}{2}\sin\theta_i \\ \frac{r}{l} & -\frac{r}{l} \end{bmatrix}$$

that maps the control inputs to the robot's velocity.
This formulation allows for the analysis of the system's dynamics and the design of

control laws based on the robot's state and control inputs.

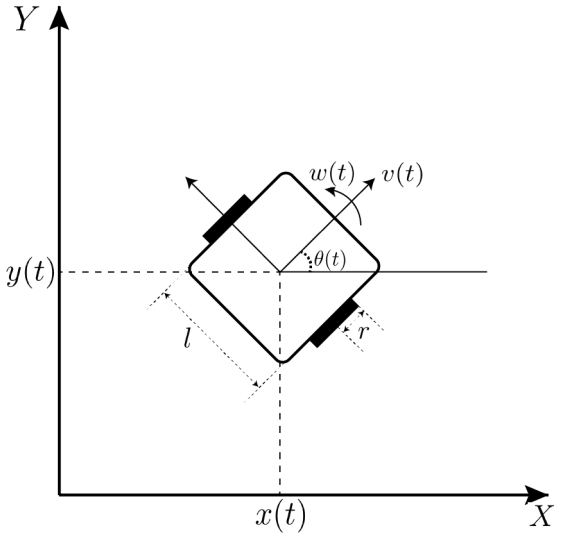

Figure 2.1: Schematic of a differential-drive robot

Unlike differential-drive robots, omnidirectional robots (figure 2.2)can move freely in any direction within the plane without being constrained by their orientation. This ability is enabled by the presence of multiple wheels (typically three or more), each mounted at different angles, that allow independent control of motion in all directions. For simplicity, we will consider a common configuration with three wheels arranged at 120° angles.

Let the state vector of the omnidirectional robot be defined by $x = [x_i, y_i, \theta_i]^t op$, where (x_i, y_i) represents the Cartesian coordinates of the robot, and θ_i is its orientation relative to a global inertial frame. The control inputs $u = [v_1, v_2, v_3]^\top$ represent the velocities of the three wheels, typically expressed as linear velocities. These wheel velocities influence the robot's linear and angular velocities, and the kinematic model of robot i is given by:

$$\dot{p}_i = \begin{bmatrix} \dot{x}_i \\ \dot{y}_i \\ \dot{\theta}_i \end{bmatrix} = \begin{bmatrix} \cos\theta_{1,i} & \cos\theta_{2,i} & \cos\theta_{3,i} \\ \sin\theta_{1,i} & \sin\theta_{2,i} & \sin\theta_{3,i} \\ \frac{1}{l} & \frac{1}{l} & \frac{1}{l} \end{bmatrix} \begin{bmatrix} v_{1,i} \\ v_{2,i} \\ v_{3,i} \end{bmatrix} \tag{2.6}$$

where $\theta_{1,i}, \theta_{2,i}, \theta_{3,i}$ represent the angles of the three wheels relative to the global frame of robot, and l is the distance between each wheel and the center of the robot.

The angles $\theta_{1,i}, \theta_{2,i}, \theta_{3,i}$ describe the orientation of the individual wheels relative to the robot's frame of reference and directly influence the robot's motion in x and y directions. This model allows the omnidirectional robot to move in any direction, as the individual wheel velocities can be controlled independently. The system is described as fully holonomic, meaning the robot can move in any direction within the plane without

any constraints on its motion, unlike differential-drive robots.
The kinematic model for omnidirectional robots arises from the fact that each wheel contributes independently to the robot's motion in the x and y directions. The robot's total linear velocity in the x and y directions is a linear combination of the individual wheel velocities, weighted by their respective contributions to the robot's movement. Its angular velocity is determined by a constant factor that depends on the wheel configuration and the geometry of the robot. Each wheel's angular velocity is mapped into a motion vector, which collectively defines the robot's resultant velocity in the Cartesian plane.
The linear velocities are expressed as:

$$\begin{aligned}\dot{x}_i =& v_{1,i}\cos\theta_{1,i} + v_{2,i}\cos\theta_{2,i} + v_{3,i}\cos\theta_{3,i}\\ \dot{y}_i =& v_{1,i}\sin\theta_{1,i} + v_{2,i}\sin\theta_{2,i} + v_{3,i}\sin\theta_{3,i}\end{aligned}$$

The angular velocity $\dot{\theta}_i$ is influenced by all three wheels equally, with the control matrix ensuring that the robot can rotate independently of its linear motion:

$$\dot{\theta}_i = \frac{1}{l}\left(v_{1,i} + v_{2,i} + v_{3,i}\right)$$

We can now express the kinematic model in the standard control form 2.5 where:

$$f(x) = \begin{bmatrix}0\\0\\0\end{bmatrix}, \quad g(x) = \begin{bmatrix}\cos\theta_{1,i} & \cos\theta_{2,i} & \cos\theta_{3,i}\\ \sin\theta_{1,i} & \sin\theta_{2,i} & \sin\theta_{3,i}\\ \frac{1}{l} & \frac{1}{l} & \frac{1}{l}\end{bmatrix} \tag{2.7}$$

Thus, the kinematic model for the i^{th} omnidirectional robot is written as:

$$\dot{x}_i = f(x_i) + g(x_i)u_i$$

where $u_i = [v_{1,i}, v_{2,i}, v_{3,i}]^\top$ represents the wheel velocities. The control matrix $g(x)$ incorporates the angles $\theta_{1,i}, \theta_{2,i}, \theta_{3,i}$ of the wheels, which directly influence the robot's movement in the plane. This formulation allows us to independently control the robot's movement in any direction, making it an ideal model for omnidirectional robotic systems.

2.3 Graph Theory

Graph theory provides a fundamental framework for the coordination and control of multi-robot systems, particularly in the domain of formation control. This section

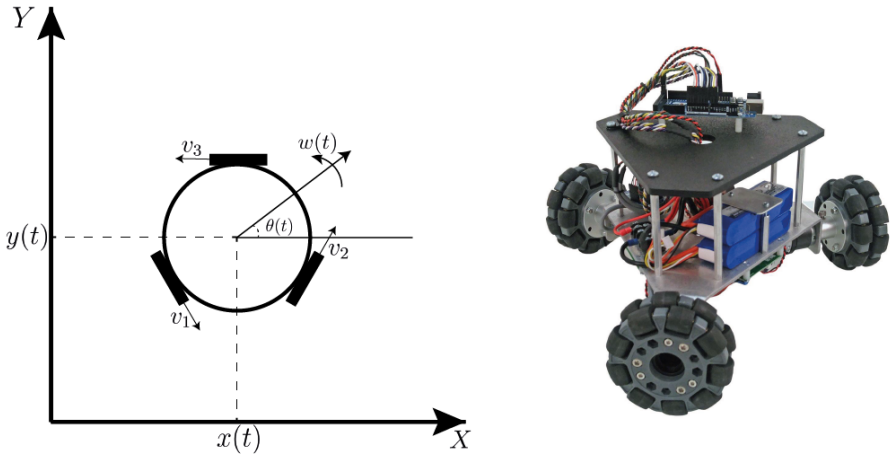

Figure 2.2: omnidirectional robot

examines how graph-theoretic concepts are applied to preserve formation structures, guarantee network connectivity, and support the design of distributed algorithms for efficient task execution. Precise formation control is often indispensable in multi-robot applications such as surveillance, mapping, and exploration, where maintaining well-defined relative positions among robots is critical to overall system performance.

2.3.1 Background and Representation of Multi-Robot Systems

A multi-robot system can be represented as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{v_1, v_2, \ldots, v_n\}$ is the set of vertices representing the robots, and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges representing the interactions among them.

This graph-theoretic model abstracts the robots' physical embodiments by representing them as nodes. Consequently, the algorithmic design primarily focuses on the network topology and inter-agent interactions. The set of edges $\mathcal{E}$ captures the critical relationships between the robots including communication, sensing, and physical collaboration. all of which are crucial for the effective functioning of the system. Such a representation allows for scalable, decentralized algorithms that can be applied to multi-robot systems with heterogeneous physical capabilities.

The edges in this graph can be classified into three primary categories, each representing a distinct type of interaction between the robots:

Communication Edges:

An *undirected* edge $e_{ij} \in \mathcal{E}$ is established between robots v_i and v_j if there is a bidirectional communication link between them. This typically occurs when the robots are

within a predefined Euclidean distance, thus defining a reliable communication range. These edges serve as the backbone for coordination, enabling the exchange of state information, task allocation commands, and planning data. Efficient communication is paramount in distributed multi-robot systems, where real-time updates are necessary for maintaining synchronization.

Sensing Edges:

Directed edges in the graph can represent the robots' sensing capabilities. A directed edge e_{ij} from robot v_i to robot v_j indicates that robot v_i is able to perceive v_j through its onboard sensors, such as cameras, LiDAR, or ultrasonic rangefinders. These edges provide crucial information on relative pose, velocity, and, in some cases, identity. Sensing edges are critical for tasks such as formation maintenance, reactive control, and situational awareness in scenarios where communication might be limited or intermittent. They allow robots to coordinate based on direct physical observations, even in the absence of explicit communication links.

Physical Interaction Edges:

In tasks requiring direct physical collaboration such as cooperative object manipulation, transport, or formation control, edges can represent a physical coupling between robots in the context of task coordination, abstracting the physical interactions into a graph-based model. These edges model kinematic and dynamic constraints imposed by shared tasks. The nature of the connection can be either rigid, forming a closed kinematic chain, or flexible, allowing for compliance between robots. The weight of these edges may be associated with the magnitude of the force exerted through the physical interaction. These types of edges are particularly important in scenarios like collaborative manipulation, where they enable algorithms to model and coordinate the physical actions of multiple robots.

This model with three types of edges (communication, sensing, and physical interaction) offers a powerful abstraction that facilitates the design of scalable, generic algorithms. These algorithms can be applied to tasks such as task planning, formation control, and resource allocation, particularly in a decentralized manner, without being tied to specific low-level physical embodiments of the robots. By reasoning over the graph $\mathcal{G}$, system-level controllers can be developed that are largely agnostic to the individual characteristics of the robots, focusing instead on the emergent properties of the network as a whole.

The figure 2.3 illustrates three representative graph structures commonly utilized in multi-robot systems. The left figure presents an undirected graph, where edges between nodes are bidirectional, indicating symmetric communication among agents. The mid-

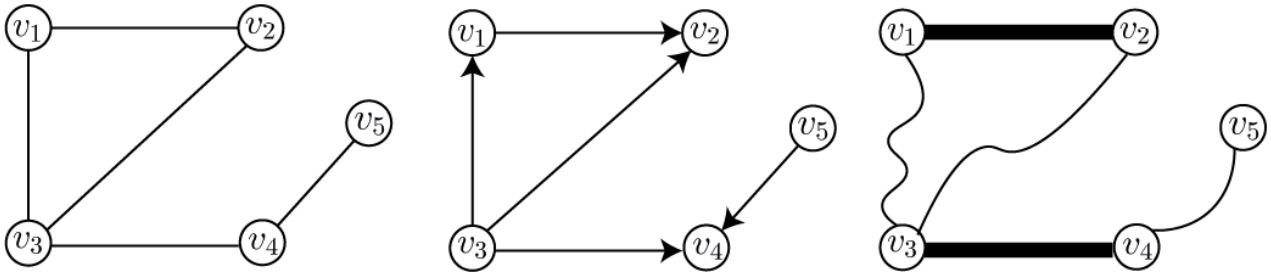

Figure 2.3: Undirected Graph, Directed Graph, Rigid and Flexible Graph

dle figure shows a directed graph, highlighting asymmetric communication links that define the flow of information. The right figure depicts a hybrid graph model that differentiates between two types of communication links: rigid links, represented by bold lines, and flexible links, illustrated by curved lines. This comparative representation emphasizes the structural variability of inter-robot communication topologies and their influence on coordination and control mechanisms.

Another important and widely used type of graph in practical multi-robot systems is the *weighted* graph. Unlike the unweighted representations where all connections are treated equally, a weighted graph assigns a numerical value to each edge, thereby quantifying the strength, cost, or reliability of the interaction between robots. Formally, a weighted graph is denoted as $\mathcal{G}(\mathcal{V}, \mathcal{E}, w)$, where $w : \mathcal{E} \leftrightarrow \mathbb{R}_+$ is a function that assigns a non-negative real value to each edge. The weight function w provides a measure of the length or cost of the connection between two vertices. In the path planing problem, The notion of shortest paths, or geodesics [3], between vertices can be defined by considering the path length as the sum of the weights along the edges. Specifically, if $\pi(v_i, v_j)$ denotes the set of all paths connecting vertices v_i and v_j, then a geodesic between v_i and v_j is any path that minimizes:

$$\min_{p \in \pi(v_i, v_j)} \text{length}(p)$$

In a connected weighted graph, the *diameter* is characterized as the maximum length among all geodesics within the graph.

In the figure 2.4, the weighted communication graph of the five-robots has a diameter of 10, representing the maximum geodesic distance between any pair of robots. Notably, the minimum path cost between v_1 and v_5 is 9, achieved along the route $v_1 \rightarrow v_3 \rightarrow v_5$, indicating a moderate level of connectivity despite the relatively sparse network topology.

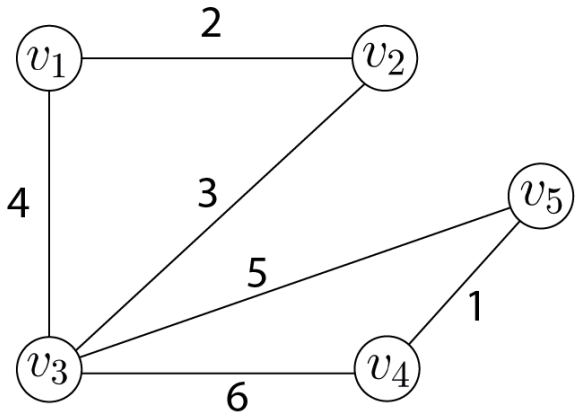

Figure 2.4: Weighted Graph

2.3.2 Spectral Graph Theory

Spectral graph theory provides a rigorous mathematical framework for characterizing the structural and dynamical properties of complex networks through the spectral analysis of matrices associated with graphs. Within the context of multi-robot systems, sensor networks, and distributed control architectures, this theory enables the quantitative assessment of key network attributes such as connectivity, robustness, consensus convergence, and information flow efficiency without the need to explicitly enumerate all possible paths. This section introduces the fundamental principles of spectral graph theory that are most relevant to the modeling, analysis, and design of cooperative robotic networks.

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ denote an undirected and unweighted graph representing a multi-robot system, where $\mathcal{V} = \{v_1, v_2, \ldots, v_n\}$ is the set of robots (nodes) and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ represents bidirectional communication links between robots. The structure of $\mathcal{G}$ is captured by its *adjacency* matrix $A = [a_{ij}] \in \mathbb{R}^{n \times n}$, defined as:

$$a_{ij} = \begin{cases} 1, & \text{if } (v_i, v_j) \in \mathcal{E}, \\ 0, & \text{otherwise.} \end{cases}$$

Since $\mathcal{G}$ is undirected, the adjacency matrix is symmetric, i.e., $A = A^T$.

The *degree* matrix D is a diagonal matrix whose elements correspond to the degree of each node, such that $D_{ii} = \sum_{j=1}^{n} a_{ij}$. The *Laplacian* matrix of the graph, which plays a central role in spectral analysis, is defined as:

$$L = D - A.$$

The Laplacian encodes both the local connectivity (through node degrees) and the global structure (through adjacency). It is symmetric and positive semi-definite, implying that all its eigenvalues are real and non-negative.

Let $\lambda_1, \lambda_2, \ldots, \lambda_n$ denote the eigenvalues of L, ordered such that:

$$0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n.$$

The smallest eigenvalue $\lambda_1 = 0$ always corresponds to the eigenvector $\mathbf{1}_n = [1, 1, \ldots, 1]^T$, which represents a uniform state across all nodes. The second smallest eigenvalue, λ_2, known as the *algebraic connectivity* or *Fiedler value*, is particularly significant as it quantifies the level of connectivity within the network. A larger λ_2 indicates a more strongly connected graph and, consequently, a more cohesive multi-robot team. If $\lambda_2 > 0$, the graph $\mathcal{G}$ is connected; conversely, if $\lambda_2 = 0$, the graph is disconnected.

The largest eigenvalue λ_n relates to the maximum degree of the network and reflects the overall communication load and synchronizability of the system. Together, the Laplacian spectrum $\{\lambda_i\}$ provides a compact summary of the network's topological features.

In multi-robot coordination tasks such as formation control, consensus, and synchronization, the Laplacian matrix directly influences the system's collective dynamics. For example, consider a continuous-time consensus protocol of the form:

$$\dot{x}(t) = -Lx(t),$$

where $x(t) = [x_1(t), x_2(t), \ldots, x_n(t)]^T$ represents the state of all robots. The convergence rate of the consensus process is governed by λ_2: a higher algebraic connectivity results in faster convergence and enhanced robustness to communication failures.

The eigenstructure of L allows engineers to assess the resilience of the network to link losses, optimize topology for minimal energy consumption, or ensure rapid agreement across the team. Moreover, the eigenvectors corresponding to smaller eigenvalues reveal clusters or groups of robots that are more tightly connected, aiding in hierarchical coordination or modular formation control.

Several spectral metrics derived from the Laplacian spectrum are relevant in multi-robot networks:

- **Spectral radius:** $\rho(L) = \lambda_n$, which determines the upper bound of dynamic response speed and can indicate potential communication bottlenecks.
- **Algebraic connectivity:** λ_2, measuring how well the robots are interconnected and how efficiently information propagates.

- **Spectral gap:** $\lambda_2 - \lambda_1 = \lambda_2$, which quantifies the robustness of the network to disconnections.

Physically, a high algebraic connectivity implies that even if a few communication links fail, the team remains cohesive and capable of maintaining collective objectives. Conversely, a low λ_2 reveals vulnerabilities to isolation or fragmentation within the swarm.

To illustrate the concepts discussed above, consider a simple undirected and unweighted communication network composed of five robots, denoted as $\mathcal{V} = \{v_1, v_2, v_3, v_4, v_5\}$. The communication edges are defined as:

$$\mathcal{E} = \{(v_1, v_2), (v_1, v_3), (v_2, v_3), (v_3, v_4), (v_4, v_5)\}.$$

This topology represents a small multi-robot team where robot v_3 acts as a central relay

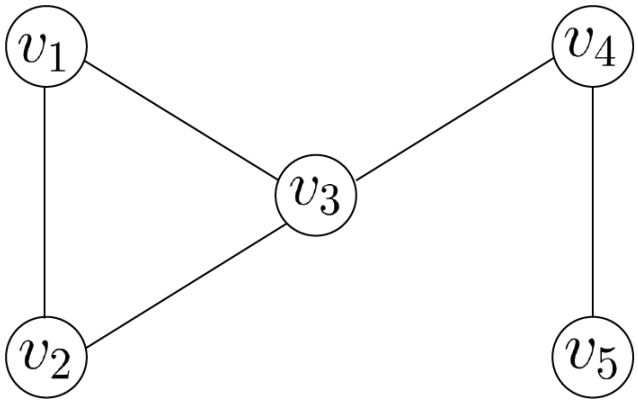

Figure 2.5: Undirected communication graph for 5 robots system illustrating Laplacian-based connectivity

between the two subgroups $\{v_1, v_2\}$ and $\{v_4, v_5\}$. The adjacency matrix A describing the above communication links is:

$$A = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}.$$

The corresponding degree matrix D is diagonal, with entries equal to the number of links for each robot:

$$D = \mathrm{diag}(2, 2, 3, 2, 1).$$

The Laplacian matrix $L = D - A$ is therefore:

$$L = \begin{bmatrix} 2 & -1 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 & 0 \\ -1 & -1 & 3 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & -1 & 1 \end{bmatrix}.$$

The eigenvalues of the Laplacian matrix L are approximately:

$$\lambda(L) = \{0, 0.5188, 2.3111, 3, 4.1701\}. \tag{2.8}$$

The smallest eigenvalue $\lambda_1 = 0$ corresponds to the uniform eigenvector $\mathbf{1}_5 = [1, 1, 1, 1, 1]^\top$, confirming that the network is connected.
The second smallest eigenvalue $\lambda_2 = 0.5188$ represents the *algebraic connectivity* of the network, indicating a relatively weak connection between the two ends of the chain (particularly between robots v_3 and v_4). The largest eigenvalue $\lambda_5 = 4.1701$ reflects the maximum diffusion rate of information through the network.
From a multi-robot perspective, this spectral profile implies that:

- The network is connected, enabling all robots to eventually reach consensus or coordinate.
- The low value of λ_2 suggests that the network could become disconnected if the link between v_3 and v_4 fails.
- The central node v_3 has the highest degree, acting as a critical communication hub that balances local and global information exchange.

Since the objective is the coordination of a multi-robot system, the following results are essential for characterizing the connectivity of such a system.

Theorem 2.1 *[3] The graph $\mathcal{G}$ is connected if and only if the second smallest eigenvalue λ_2 is strictly positive.*

Proof: The Laplacian matrix $L = D - A$ is symmetric and positive semi-definite. Moreover, it satisfies $L\mathbf{1} = \mathbf{0}$, where $\mathbf{1} \in \mathbb{R}^n$ is the all-ones vector; hence, 0 is always an eigenvalue of L, and we denote the eigenvalues in non-decreasing order as $0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$.

We first show that if $\mathcal{G}$ is connected, then $\lambda_2 > 0$. Let $\mathbf{x} \in \mathbb{R}^n$ satisfy $L\mathbf{x} = \mathbf{0}$. Then,

$$\mathbf{x}^\top L\mathbf{x} = 0.$$

Using the well-known quadratic form identity for the Laplacian,

$$\mathbf{x}^\top L\mathbf{x} = \sum_{(i,j)\in E} (x_i - x_j)^2,$$

it follows that $(x_i - x_j)^2 = 0$ for every edge $(i, j) \in E$, i.e., $x_i = x_j$ whenever vertices i and j are adjacent. Since $\mathcal{G}$ is connected, any two vertices are linked by a path, and by transitivity, all entries of $\mathbf{x}$ must be equal. Thus, $\mathbf{x} \in \text{span}\{\mathbf{1}\}$, and the nullspace of L is one-dimensional. Consequently, the algebraic multiplicity of the eigenvalue 0 is one, which implies $\lambda_2 > 0$.

Conversely, assume $\lambda_2 > 0$. Then the eigenvalue 0 has multiplicity one. Suppose, for contradiction, that $\mathcal{G}$ is disconnected. Then $\mathcal{G}$ contains at least two connected components, say C_1 and C_2. Define a vector $\mathbf{y} \in \mathbb{R}^n$ by

$$y_i = \begin{cases} 1 & \text{if } i \in C_1, \\ 0 & \text{otherwise.} \end{cases}$$

Since no edge connects C_1 to its complement, we have $y_i = y_j$ for all $(i, j) \in E$, and thus $L\mathbf{y} = \mathbf{0}$. However, $\mathbf{y}$ is not a scalar multiple of $\mathbf{1}$, contradicting the assumption that the nullspace of L is one-dimensional. Therefore, $\mathcal{G}$ must be connected. ■.

2.4 Formation Control Problems

Formation control is a fundamental problem in multi-robot systems (MRS), wherein a group of robots must coordinate their motion to either maintain a prescribed geometric formation or navigate collectively toward a common goal while satisfying inter-robot spatial constraints. This problem has attracted significant research interest due to its broad applicability in domains such as autonomous ground vehicles, unmanned aerial vehicles (UAVs), mobile sensor networks, and collaborative robotics. Formation control inherently involves both *geometric constraints*, which define the desired relative positions among robots, and *dynamic constraints*, which govern the stability, convergence, and optimality of the collective motion.

The ability to achieve and maintain a desired formation is essential for the successful execution of complex cooperative tasks. In swarm robotics, for instance, formation maintenance enables emergent collective behaviors such as area coverage, environmental exploration, and target tracking. Moreover, coordinated motion enhances system robustness by facilitating collision avoidance, improving sensing coverage, and enabling energy-efficient navigation strategies.

Mathematically, formation control can be cast as a distributed coordination problem,

where each robot computes its control input based on local measurements relative to its neighbors. In the simplest architecture, a *leader-follower* framework is adopted, wherein one robot acts as a leader and the others regulate their motion to preserve fixed relative positions with respect to the leader. However, more scalable and robust approaches employ *decentralized, leaderless* control schemes, where all agents cooperate symmetrically without reliance on a central authority, thereby improving fault tolerance and flexibility.

A primary challenge in formation control lies in designing control laws that guarantee convergence to and maintenance of the desired formation in the presence of model uncertainties, external disturbances, communication delays, and limited sensing capabilities. These control laws typically rely on feedback mechanisms that use local relative position and velocity measurements to compute corrective actions for each robot.

Formation control is inherently interdisciplinary, drawing upon control theory, graph theory, optimization, and differential geometry. From a control-theoretic perspective, MRS are modeled as networked multi-agent systems with interaction topologies encoded by graphs, and stability is analyzed using Lyapunov-based or passivity-based methods. Mathematically, the problem often involves nonlinear dynamics and constrained optimization, particularly when the desired formation is specified in terms of inter-robot distances, relative angles, or global shape constraints.

2.4.1 Mathematical Formulation of Formation Control

Formation control in multi-robot systems entails coordinating the motion of a group of robots such that they preserve specific geometric relationships over time. This subsection presents a rigorous mathematical formulation of the problem, encompassing state representation, kinematic models, formation specifications, and error dynamics. Each robot i in the system is described by a state vector $\mathbf{p}_i$ that captures its position, orientation, and possibly higher-order dynamics. For a robot operating in the plane, a common minimal state representation is

$$\mathbf{p}_i = [x_i, y_i, \theta_i]^\top,$$

where $(x_i, y_i) \in \mathbb{R}^2$ denotes the robot's Cartesian position and $\theta_i \in [0, 2\pi)$ its orientation.

Remark 2.1 *For omnidirectional platforms, the state may additionally include linear velocities $(\dot{x}_i, \dot{y}_i)$ or angular velocity $\dot{\theta}_i$, depending on the modeling requirements.*

The control input $\mathbf{u}_i$ depends on the robot's actuation model. For a differential-drive robot, $\mathbf{u}_i$ typically consists of the angular velocities of the left and right wheels:

$$u_i = [w_{L,i}, w_{R,i}]^\top$$

where $w_{L,i}$ and $w_{R,i}$ are the angular velocities of the left and right wheels, respectively. For omnidirectional robots with three or more independently actuated wheels, $\mathbf{u}_i$ may represent the individual wheel velocities $[v_{1,i}, v_{2,i}, v_{3,i}]^\top$.

2.4.2 Formation Control Problem

In formation control, the objective is to ensure that the relative positions among robots adhere to a predefined geometric configuration, which may either static or time-varying. This desired formation is typically represented using a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where each node $\nu_i \in \mathcal{V}$ corresponds to a robot and each edge $(i, j) \in \mathcal{E}$ indicates a geometric constraint between robots i and j.
For each edge $(i, j) \in \mathcal{E}$, the desired relative position is specified by a vector $\mathbf{d}_{ij} \in \mathbb{R}^2$, given by:

$$\mathbf{d}_{ij} = [d_{x,ij}, d_{y,ij}]^\top, \tag{2.9}$$

where $d_{x,ij}$ and $d_{y,ij}$ denote the desired offsets in the global x- and y-directions, respectively.

Remark 2.2 *These offsets are constant for rigid formations; however, they may be time-varying in dynamic or reconfigurable formations.*

Let $\mathbf{p}_i = [x_i, y_i]^\top$ and $\mathbf{p}_j = [x_j, y_j]^\top$ denote the planar positions of robots i and j in a common inertial frame. The actual relative position vector is defined as

$$\mathbf{e}_{ij} = \mathbf{p}_i - \mathbf{p}_j = \begin{bmatrix} x_i - x_j \\ y_i - y_j \end{bmatrix}.$$

The formation error (the deviation from the desired configuration) is then given by:

$$\mathbf{e}_{ij}^d = \mathbf{e}_{ij} - \mathbf{d}_{ij} = \begin{bmatrix} (x_i - x_j) - d_{x,ij} \\ (y_i - y_j) - d_{y,ij} \end{bmatrix}.$$

This error vector quantifies the mismatch in both distance and direction between the current and desired relative positions. Its Euclidean norm is given by:

$$\|\mathbf{e}_{ij}^d\| = \sqrt{(x_i - x_j - d_{x,ij})^2 + (y_i - y_j - d_{y,ij})^2},$$

represents the scalar formation error magnitude.
In applications involving non-holonomic robots or orientation-sensitive tasks (e.g., linear, circular, or grid formations), the relative orientation error must also be considered:

$$\theta_{ij}^{d} = \theta_i - \theta_j - \theta_{ij}^{\text{des}}, \tag{2.10}$$

where θ_{ij}^{des} is the desired relative heading between robots i and j.
The time evolution of the formation error is governed by the relative kinematics of the robots. Assuming single-integrator dynamics $\dot{\mathbf{p}}_i = \mathbf{u}_i$)(for simplicity), the error dynamics are

$$\dot{\mathbf{e}}_{ij}^{d} = \dot{\mathbf{e}}_{ij} = \dot{\mathbf{p}}_i - \dot{\mathbf{p}}_j = \mathbf{u}_i - \mathbf{u}_j.$$

For general kinematic models, $\dot{\mathbf{p}}_i = g_i(\mathbf{p}_i)\mathbf{u}_i$, where $g_i(\cdot)$ is the robot-specific kinematic mapping. This leads to the following error dynamics:

$$\dot{\mathbf{e}}_{ij}^{d} = g_i(\mathbf{p}_i)\mathbf{u}_i - g_j(\mathbf{p}_j)\mathbf{u}_j.$$

A standard decentralized control law that drives the formation error to zero employs proportional feedback:

$$\mathbf{u}_i = -K \sum_{j \in \mathcal{N}_i} \mathbf{e}_{ij}^{d},$$

where $K \in \mathbb{R}^{2\times 2}$ is a positive definite gain matrix and $\mathcal{N}_i$ denotes the set of neighbors of robot i in the interaction graph $\mathcal{G}$. Under appropriate connectivity assumptions (e.g., $\mathcal{G}$ is connected), this control law ensures asymptotic convergence of $\mathbf{e}_{ij}^{d} \to \mathbf{0}$ for all $(i, j) \in \mathcal{E}$.

2.4.3 Graph-Based Formation Control

Building upon the graph-theoretic foundations established in the previous section, graph-based formation control utilizes the interaction topology among robots to encode coordination constraints and synthesize distributed control laws. In this framework, the formation task is not merely a geometric specification but rather a networked control problem, where the structure of the underlying graph $\mathcal{G}$ directly influences feasibility, stability, and robustness.

A central principle is that the Laplacian matrix $L = D - A$ (which has already been shown to characterize connectivity through its second smallest eigenvalue λ_2) also governs the dynamics of formation error propagation. This connection between network topology and closed-loop stability was rigorously formalized in [38], which demonstrated that the stability of vehicle formations under decentralized control is entirely determined by the spectral properties of the underlying information flow graph.

When each robot updates its motion based on relative measurements from its neighbors, the collective closed-loop system often admits a compact representation of the form

$$\dot{\mathbf{p}} = -(L \otimes I_d)\,\mathbf{f}(\mathbf{p}), \tag{2.11}$$

where $\mathbf{p} = [\mathbf{p}_1^\top, \ldots, \mathbf{p}_n^\top]^\top \in \mathbb{R}^{nd}$ stacks the positions of all n robots in $\mathbb{R}^d$, and $\mathbf{f}(\cdot)$ encodes the formation error function. The Kronecker product with the identity I_d reflects the decoupling of dynamics across spatial dimensions, while the Laplacian ensures that control actions are consistent with the interaction graph.
Two dominant paradigms emerge in this context: *displacement-based* and *distance-based* formation control. In the displacement-based approach, the desired formation is specified by a set of constant relative vectors $\{\mathbf{d}_{ij}\}_{(i,j)\in\mathcal{E}}$, and each robot computes its control input to minimize the deviation $\mathbf{e}_{ij}^d = (\mathbf{p}_i - \mathbf{p}_j) - \mathbf{d}_{ij}$. Under single-integrator dynamics ($\dot{\mathbf{p}}_i = \mathbf{u}_i$), the proportional control law

$$\mathbf{u}_i = -k \sum_{j\in\mathcal{N}_i} \mathbf{e}_{ij}^d, \quad k > 0,$$

yields global asymptotic stability of the desired formation, provided that $\mathcal{G}$ is connected (i.e., $\lambda_2 > 0$) [39]. This result follows directly from the fact that the error dynamics evolve in the range space of L, which is orthogonal to the consensus subspace span$\{\mathbf{1}\}$. However, displacement-based control implicitly assumes a shared global or local coordinate frame among neighboring robots which may not be available or may be heterogeneous in certain environments. To address this limitation, distance-based formation control relies exclusively on inter-agent distance measurements $\|\mathbf{p}_i - \mathbf{p}_j\|$. The formation is then defined by a set of target distances $\{d_{ij}^*\}_{(i,j)\in\mathcal{E}}$, and stability is typically achieved through gradient descent on a potential function such as

$$V(\mathbf{p}) = \frac{1}{4} \sum_{(i,j)\in\mathcal{E}} \left(\|\mathbf{p}_i - \mathbf{p}_j\|^2 - (d_{ij}^*)^2\right)^2.$$

While this approach is frame-free and can be implementable with range sensors (e.g., lidar), it introduces a fundamental challenge: multiple configurations (some of which may be reflections or distortions of the desired shape) may satisfy the same distance constraints. Consequently, convergence to the correct formation is not guaranteed unless the graph $\mathcal{G}$ possesses sufficient structural rigidity.

This observation underscores the critical role of *rigidity theory* in formation control. A formation is said to be *infinitesimally rigid* in $\mathbb{R}^2$ if the only continuous motions that preserve all edge lengths are rigid-body transformations, such as translation and rotation. From a graph-theoretic perspective, this property is equivalent to the rigidity

matrix having full rank ($2n-3$ for $n \geq 3$ agents in the plane) [40]. Triangulated graphs such as complete graphs on three or more nodes, or Henneberg constructions, are canonical examples of rigid graphs and are widely used in practice to ensure formation uniqueness [41].

Recent extensions of graph-based formation control address dynamic environments through time-varying topologies, directed communication, and hybrid control architectures. For instance, under switching graphs that are jointly connected over bounded intervals, consensus-based formation laws remain stable [42]. Similarly, integrating Laplacian-based feedback with model predictive control enables online reconfiguration while respecting collision-avoidance and actuation constraints [43].

Chapter 3

Safe Multi-Robot Systems: A Control Barrier Function Approach

3.1 Introduction and Motivation

In recent years, the concept of barrier functions has developed into a fundamental framework in control theory, offering rigorous tools to guarantee safety and enforce system constraints in complex dynamical systems. Initially developed in the context of constrained optimization during the 1960s and 1970s, barrier functions were conceived to prevent optimization trajectories from exiting feasible regions by imposing a penalty for approaching constraint boundaries [44]. Over time,The elegance of this idea found a natural extension within control theory, a field in which the assurance of closed-loop system safety has become a fundamental and enduring priority.

The introduction of Control Barrier Functions (CBFs) represented a major advancement in the formalization of safety guarantees within feedback control design. Rooted in the principles of Lyapunov stability theory, CBFs offer rigorous tools to ensure the forward invariance of safe sets, thereby guaranteeing that once the system state enters a prescribed safe region, it remains confined within it for all future time. This framework established a systematic and verifiable methodology for enforcing safety in continuous-time nonlinear systems, effectively addressing challenges that classical control techniques often struggled to overcome[12],[11].

A principal motivation for the adoption of Control Barrier Functions (CBFs) stems from their capacity to harmonize safety requirements with performance-oriented control objectives. By embedding CBF-based constraints within optimization-based control architectures (such as quadratic programs (QPs)) engineers can synthesize controllers that rigorously enforce safety while minimally perturbing a nominal, task-driven control law. When integrated with Control Lyapunov Functions (CLFs), CBFs enable

a unified framework in which stability and safety are jointly guaranteed. This synergy allows real-time controllers to reconcile potentially competing objectives, such as trajectory tracking and collision avoidance, without compromising formal safety guarantees[11][13].

In recent years, the theory of CBFs has expanded into several essential extensions. Zeroing Control Barrier Functions (ZCBFs)[12] refine the concept by allowing the barrier function to approach zero smoothly at the boundary of the safe set, resulting in smoother control actions and improved robustness. High-Order Control Barrier Functions (HOCBFs) generalize the concept even further, making it applicable to systems with higher relative degrees (i.e., where the safety constraint depends on higher-order derivatives of the state) [11]. These advances have significantly broadened the applicability of barrier-based safety methods to complex, nonlinear, and multi-agent systems.

3.2 Control Barrier Function (CBF) Theory

The investigation of safety in dynamical systems has its origins in the 1940s, with the seminal work of Nagumo[45],[46], who formulated the necessary and sufficient conditions for ensuring set invariance.

Consider a continuous-time dynamical system

$$\dot{x} = f(x), \tag{3.1}$$

Where $x \in \mathbb{R}^n$.
Given the safe set $\mathcal{C}$ defined as

$$\mathcal{C} = \{x \in \mathbb{R}^n | h(x) \geq 0\} \tag{3.2}$$

where $h : \mathbb{R}^n \to \mathbb{R}$ is a smooth function.
The boundary of the set $\mathcal{C}$ is defined as

$$\partial\mathcal{C} = \{x \in \mathbb{R}^n | h(x) = 0\},$$

and the interior is

$$\text{Int } (\mathcal{C}) = \{x \in \mathbb{R}^n | h(x) > 0\}.$$

The regularity assumption

$$\frac{\partial h}{\partial x}(x) \neq 0, \quad \forall x \in \partial C,$$

means the boundary is a smooth hyper-surface and there are no flat or singular boundary points.

Definition 3.1 (Invariant Set) *The set $\mathcal{C} \subset \mathbb{R}^n$ is said to be* invariant *for the system*

(3.1) if for every initial condition $x(0) \in \mathcal{C}$, the corresponding solution $x(t)$ satisfies $x(t) \in \mathcal{C}$ for all $t > 0$.

Nagumo's Theorem[45],[46] gives necessary and sufficient conditions for set invariance based upon the derivative of h on the boundary of $\mathcal{C}$:

$$\mathcal{C} \quad \text{is invariant if and only if} \quad \dot{h}(x) \geq 0 \quad \forall x \in \partial C.$$

The terminology *barrier* finds its roots in optimization, where barrier functions are systematically embedded within cost functions to discourage or exclude solutions that approach undesirable or infeasible regions.

In the case of barrier certificates, one introduces an unsafe set $\mathcal{C}_u$ together with an initial set $\mathcal{C}_0$, and defines a function $B : \mathbb{R}^n \to \mathbb{R}$ such that $B(x) \leq 0$ for all $x \in \mathcal{C}_0$ and $B(x) > 0$ for all $x \in \mathcal{C}_u$.
The function B qualifies as a barrier certificate if the condition

$$\dot{B}(x) \leq 0$$

ensures the invariance of the safe set $\mathcal{C}$.
By selecting the safe set as the complement of the unsafe region $\mathcal{C} = \mathcal{C}_u^c$, and defining $B(x) = -h(x)$, the barrier certificate condition reduces to

$$\dot{h}(x) \geq 0,$$

which guarantees the invariance of $\mathcal{C}$. This formulation is equivalent to Nagumo's theorem applied on the boundary of the set.

Definition 3.2 (Reciprocal Barrier Function(RBF) *[12] For the dynamical system* (3.1)*, a continuously differentiable function $B : \text{Int}(\mathcal{C}) \to \mathbb{R}$ is said to be a* reciprocal barrier function *(RBF) for the set $\mathcal{C}$, defined by the continuously differentiable function $h : \mathbb{R}^n \to \mathbb{R}$ via* (3.2)*, if there exist class-$\mathcal{K}$ functions $\alpha_1, \alpha_2, \alpha_3$ such that, for all $x \in \text{Int}(\mathcal{C})$,*

$$\begin{aligned} \frac{1}{\alpha_1(h(x))} \leq B(x) \leq \frac{1}{\alpha_2(h(x))}, \\ L_f B(x) \leq \alpha_3(h(x)). \end{aligned} \tag{3.3}$$

Reciprocal barrier functions provide an alternative framework for safety-critical control, particularly useful when the safe set boundary is approached, as $B(x)$ tends to infinity and thus actively repels the system state from $\partial\mathcal{C}$. This property makes RBFs particularly effective for applications that demand robust avoidance capabilities, such as navigating obstacles in cluttered environments.

Definition 3.3 (Extended Class-$\mathcal{K}$ Function) *A continuous function $\alpha : (-b, a) \to (-\infty, \infty)$, for some $a, b > 0$, is said to belong to the class of* extended class-$\mathcal{K}$ functions *if it is strictly increasing and satisfies $\alpha(0) = 0$.*

Definition 3.4 (Zeroing Barrier Function (ZBF)) *[12] For the dynamical system (3.1), a continuously differentiable function $h : \mathbb{R}^n \to \mathbb{R}$ is called a* zeroing barrier function *(ZBF) for the set $\mathcal{C}$, defined by (3.2), if there exists an extended class-$\mathcal{K}$ function α and a domain $\mathcal{X} \subseteq \mathbb{R}^n$ with $\mathcal{C} \subseteq \mathcal{X}$ such that, for all $x \in \mathcal{X}$,*

$$L_f h(x) \geq -\alpha(h(x)). \tag{3.4}$$

The zeroing barrier function (ZBF) framework was introduced to overcome a key limitation of reciprocal barrier functions (RBFs): while RBFs grow unbounded as the system approaches the boundary of the safe set (leading to numerical instability and impractical control demands) ZBFs remain finite over the entire domain $\mathcal{X}$. By enforcing the condition

$$L_f h(x) \geq -\alpha(h(x)),$$

ZBFs ensure that the safe set $\mathcal{C}$ is forward invariant without requiring the barrier function itself to diverge. This property makes ZBFs particularly well-suited for real-time implementation in safety-critical robotic systems, where bounded control signals and computational robustness are essential. While the preceding results focused on

Property	**BF**	**RBF**	**ZBF**
Definition	$h : \mathcal{X} \to \mathbb{R}$, $\mathcal{C} = \{x \mid h(x) \geq 0\}$	$B : \mathrm{Int}(\mathcal{C}) \to \mathbb{R}$	$h : \mathcal{X} \to \mathbb{R}$, $\mathcal{C} = \{x \mid h(x) \geq 0\}$
Regularity	$h \in \mathcal{C}^1(\mathcal{X})$	$B \in \mathcal{C}^1(\mathrm{Int}(\mathcal{C}))$	$h \in \mathcal{C}^1(\mathcal{X})$
Behavior as $x \to \partial\mathcal{C}$	$h(x) \to 0$	$B(x) \to +\infty$	$h(x) \to 0$
Boundedness on $\mathcal{C}$	Bounded	Unbounded	Bounded
Safety condition	$\dot{h}(x) \geq 0$ on $\partial\mathcal{C}$	$\dot{B}(x) \leq \alpha_3(h(x))$	$\dot{h}(x) \geq -\alpha(h(x))$ on $\mathcal{X}$
Key advantage	Simple, direct link to Nagumo's theorem	Strong repulsion from boundary	Finite values, global condition on $\mathcal{X}$
Main limitation	Only enforces invariance on the boundary	Numerical instability near $\partial\mathcal{C}$	Requires extended α for global invariance

Table 3.1: Comparison of barrier function frameworks for autonomous systems (3.1)

closed dynamical systems (systems evolving autonomously without external inputs), the theory of viability [47, 48] extends these concepts to open systems.
Consider a nonlinear control-affine system

$$\dot{x} = f(x) + g(x)u, \tag{3.5}$$

where $x \in \mathcal{X} \subset \mathbb{R}^n$ denotes the system state,
$u \in \mathcal{U} \subset \mathbb{R}^m$ is the control input, $\mathcal{U}$ is the admissible inputs set,
$f : \mathbb{R}^n \to \mathbb{R}^n$ represents the drift (or unforced) dynamics,
$g : \mathbb{R}^n \to \mathbb{R}^{n \times m}$ is the input matrix that maps control inputs to state-space directions,
f and g are assumed locally Lipschitz.
This class of systems encompasses a wide range of robotic platforms.
This extension necessitates a shift from passively invariant sets to *controlled invariant sets* that is, sets that can be rendered invariant through the appropriate design of a feedback controller.

3.2.1 Control Lyapunov Functions

To motivate the use of safety mechanisms such as control barrier functions for systems of this form, we begin by recalling the classical objective of system stabilization. Specifically, consider a nonlinear control system (3.5) with the goal of asymptotically stabilizing the state x to an equilibrium point $x^* = 0$, i.e., ensuring that $x(t) \to 0$.
In a nonlinear setting, this is typically achieved by constructing a feedback control law $u = k(x)$ that drives a positive definite Lyapunov function $V : \mathcal{X} \subset \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ to zero.
More precisely, if there exists a control law $u = k(x)$ such that

$$\dot{V}(x, k(x)) \leq -\gamma(V(x)), \tag{3.6}$$

where

$$\dot{V}(x, k(x)) = L_f V(x) + L_g V(x)\, k(x),$$

then the system is asymptotically stable at $x^* = 0$.
Here, $\gamma : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is a class-$\mathcal{K}$ function (defined for simplicity over the non-negative real line), satisfying $\gamma(0) = 0$ and being strictly increasing.

Remark 3.1 *The terms $L_f V(x)$ and $L_g V(x)$ denote the Lie derivatives of the function V along the vector fields f and g, respectively.*
In particular, $L_f V(x) = \nabla V(x)^\top f(x)$ and $L_g V(x) = \nabla V(x)^\top g(x)$.
This notation allows the time derivative of V along the trajectories of the controlled

system to be expressed compactly.

Consequently, the problem of stabilizing a nonlinear system can be interpreted as designing a control input that induces a one-dimensional stable dynamics on the Lyapunov function V, governed by the differential inequality

$$\dot{V} \leq -\gamma(V).$$

By the comparison lemme (Lemma 3.4 in [49]), this ensures that the full-order system (3.5) is asymptotically stable under the feedback law $u = k(x)$.
The preceding observations motivate the concept of a *control Lyapunov function* (CLF), which characterizes system stability without requiring explicit construction of the feedback controller $u = k(x)$. This idea was first rigorously formalized by Sontag in 1983 [50], demonstrating that it suffices to ensure the existence of a control input satisfying a specific inequality on the time derivative of V.

Definition 3.5 *A positive definite function $V : \mathcal{X} \subset \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ is a CLF if, for all $x \in \mathcal{X}$, there exists a control input $u \in U$ such that*

$$\inf_{u \in \mathcal{U}} \left[L_f V(x) + L_g V(x)\, u \right] \leq -\gamma(V(x)), \tag{3.7}$$

where $\gamma : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is again a class-$\mathcal{K}$ function.

This definition is significant because it enables us to characterize the entire set of stabilizing controllers at each state x, defined as:

$$K_{\text{clf}}(x) := \left\{ u \in \mathcal{U} \mid L_f V(x) + L_g V(x)\, u \leq -\gamma(V(x)) \right\}.$$

Note that this condition is affine in u, making it directly compatible with optimization-based control synthesis. Furthermore, it provides a practical criterion for verifying whether a given V qualifies as a CLF. For instance, if $\mathcal{U} = \mathbb{R}^m$, then it is straightforward to verify that:

$$L_g V(x) = 0 \quad \Rightarrow \quad L_f V(x) \leq -\gamma(V(x)) \quad \Rightarrow \quad K_{\text{clf}}(x) \neq \emptyset.$$

Thus, the existence of a non-empty set of feasible controls at each point implies the existence of stabilizing controllers.
More generally, the following theorem provides the foundational link between the existence of a control Lyapunov function and asymptotic stabilization.

Theorem 3.1 *[51] Consider the nonlinear control system* (3.5)*. If there exists a control Lyapunov function $V : \mathcal{X} \subset \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ satisfying condition* (3.7)*, then any Lipschitz*

continuous feedback control law $u : \mathcal{X} \to U$ *such that* $u(x) \in K_{\text{clf}}(x)$ *for all* $x \in \mathcal{X}$ *renders the equilibrium* $x^* = 0$ *asymptotically stable.*

3.2.2 Control Barrier Functions

Unlike stability, which concerns driving the system state toward a point or set, safety is concerned with enforcing the invariance of a prescribed set i.e., ensuring that the system never leaves a designated safe region. Specifically, we consider a set $\mathcal{C} \subset \mathcal{X} \subset \mathbb{R}^n$ defined as the superlevel set of a continuously differentiable function $h : \mathcal{X} \to \mathbb{R}$:

$$\begin{aligned} \mathcal{C} &= \{x \in D \,|\, h(x) \geq 0\}, \\ \partial\mathcal{C} &= \{x \in D \,|\, h(x) = 0\}, \\ \text{Int}(\mathcal{C}) &= \{x \in D \,|\, h(x) > 0\}. \end{aligned} \tag{3.8}$$

We refer to $\mathcal{C}$ as the *safe set*.
Let $u = k(x)$ be a feedback controller law such that the resulting closed-loop system

$$\dot{x} = f_{\text{cl}}(x) := f(x) + g(x)k(x) \tag{3.9}$$

is locally Lipschitz.
Under this assumption, for any initial condition $x_0 \in \mathcal{X}$, there exists a maximal interval of existence $I(x_0) = [0, \tau_{\max})$ over which the solution $x(t)$ of (3.9) is unique and well-defined. If the system is forward complete (see, e.g., [49]), then $\tau_{\max} = \infty$. This property enables a formal definition of safety.

Definition 3.6 *[13] The set* $\mathcal{C}$ *is said to be* forward invariant *if, for every initial condition* $x_0 \in \mathcal{C}$*, the corresponding solution* $x(t)$ *of system* (3.9) *satisfies* $x(t) \in \mathcal{C}$ *for all* $t \in I(x_0)$*. The system* (3.9) *is said to be* safe *with respect to the set* $\mathcal{C}$ *if* $\mathcal{C}$ *is forward invariant.*

Definition 3.7 *A function* $\alpha : \mathbb{R} \to \mathbb{R}$ *is called an* extended class-$\mathcal{K}_\infty$ function *if it satisfies* $\alpha(0) = 0$ *and is strictly increasing on* $\mathbb{R}$*.*

Definition 3.8 *[13] Let* $\mathcal{C} \subset \mathcal{X} \subset \mathbb{R}^n$ *be the superlevel set of a continuously differentiable function* $h : \mathcal{X} \to \mathbb{R}$*, i.e.,* $\mathcal{C} = \{x \in \mathcal{X} \mid h(x) \geq 0\}$*. Then,* h *is said to be a* control barrier function *(CBF) for the control system* (3.5) *if there exists an extended class-*$\mathcal{K}_\infty$ *function* $\alpha : \mathbb{R} \to \mathbb{R}$ *such that*

$$\sup_{u \in \mathcal{U}} \left[L_f h(x) + L_g h(x)\, u \right] \geq -\alpha(h(x)) \tag{3.10}$$

for all $x \in \mathcal{X}$*.*

The extension of invariance conditions originally defined only on the boundary $\partial\mathcal{C}$ via $\dot{h} \geq 0$ to the entire set $\mathcal{C}$ was first formalized in [47]. Specifically, the condition

$$\dot{h} \geq -h \quad \forall x \in \mathcal{C},$$

was proposed as a means to enforce safety over the full domain. This can be interpreted as a special case of a control barrier function (CBF) where the class-$\mathcal{K}_\infty$ function is chosen as $\alpha(r) = r$, as in Definition (3.10).

Guaranteed safety via Control Barrier Functions (CBFs) is achieved by characterizing the set of all admissible control inputs that preserve the safety of the system. Specifically, for a given state $x \in \mathcal{X}$, we define the safe control set as

$$K_{\text{cbf}}(x) := \{u \in \mathcal{U} \mid L_f h(x) + L_g h(x)\, u + \alpha(h(x)) \geq 0\}\,, \tag{3.11}$$

which quantifies the collection of all control inputs at x that ensure the system remains within the safe set $\mathcal{C}$. As in the case of Control Lyapunov Functions (CLFs), this formulation allows us to explicitly characterize the feasible region of control actions that maintain safety. The main result established in [12], and central to the theory of CBFs, is that the existence of such a function h implies that the closed-loop system is guaranteed to be safe provided that the applied control input $u(x)$ is selected from $K_{\text{cbf}}(x)$ at each point in time.

Theorem 3.2 *[13] Let $\mathcal{C} \subset \mathbb{R}^n$ be a set defined as the superlevel set of a continuously differentiable function $h : \mathcal{X} \subset \mathbb{R}^n \to \mathbb{R}$, i.e., $\mathcal{C} = \{x \in \mathcal{X} \mid h(x) \geq 0\}$. If h is a control barrier function on $\mathcal{X}$ and $\frac{\partial h}{\partial x}(x) \neq 0$ for all $x \in \partial\mathcal{C}$, then any Lipschitz continuous feedback controller $u : \mathcal{X} \to U$ such that $u(x) \in K_{\text{cbf}}(x)$ for all $x \in \mathcal{X}$ renders the set $\mathcal{C}$ forward invariant. Additionally, if $\mathcal{C}$ is bounded and contains an equilibrium point, it is asymptotically stable under such a controller.*

Remark 3.2 *It is crucial to highlight that this result not only guarantees the invariance of the safe set $\mathcal{C}$ but also ensures its asymptotic stability. This property plays a pivotal role in practical implementations, as it provides a safety net against small deviations from the desired state. Although external disturbances, modeling errors, or noise may cause the system to temporarily leave the safe set, the main CBF theorem ensures that controllers designed within the set $\mathcal{K}_{cbf}(x)$ will drive the system back to $\mathcal{C}$, thereby maintaining safety over time and preventing long-term instability.*

While Theorem 3.2 establishes that the existence of a control barrier function is sufficient to guarantee safety, it is natural to ask whether this condition is also necessary.

the following theorem establishes a necessary condition for a set to be invariant under a controller. Specifically, if it is possible to maintain the system within a safe set $\mathcal{C}$ using a Lipschitz continuous controller, then a control barrier function (CBF) must necessarily exist on this set.

Theorem 3.3 *[13] Let $\mathcal{C} \subset \mathcal{X} \subseteq \mathbb{R}^n$ be a compact set, where $\mathcal{C}$ is defined as the superlevel set of a continuously differentiable function $h : \mathcal{X} \to \mathbb{R}$. Suppose that $\nabla h(x) \neq 0$ for all $x \in \partial\mathcal{C}$. If there exists a Lipschitz continuous feedback controller $u = k(x)$ such that the closed-loop system (3.9) renders $\mathcal{C}$ forward invariant, then the restriction of h to $\mathcal{C}$, i.e., $h|_{\mathcal{C}} : \mathcal{C} \to \mathbb{R}_{\geq 0}$, is a control barrier function on $\mathcal{C}$.*

Definition 3.9 (Reciprocal Control Barrier Function(RCBF)) *[12] Consider the control system (3.5) and the set $\mathcal{C} \subset \mathcal{X} \subseteq \mathbb{R}^n$ defined as the superlevel set of a continuously differentiable function $h : \mathcal{X} \to \mathbb{R}$ via (3.2). A continuously differentiable function $B : \mathrm{Int}(\mathcal{C}) \to \mathbb{R}$ is called a* reciprocal control barrier function *(RCBF) if there exist class-$\mathcal{K}$ functions $\alpha_1, \alpha_2, \alpha_3$ such that, for all $x \in \mathrm{Int}(\mathcal{C})$,*

$$\frac{1}{\alpha_1(h(x))} \leq B(x) \leq \frac{1}{\alpha_2(h(x))},$$
$$\inf_{u \in U} \left[L_f B(x) + L_g B(x)\, u - \alpha_3(h(x)) \right] \leq 0. \tag{3.12}$$

The RCBF framework extends the reciprocal barrier function concept to control-affine systems, enabling safety guarantees through a control law that counteracts the unbounded growth of $B(x)$ near the boundary $\partial\mathcal{C}$ while ensuring forward invariance of the safe set.
The RCBF B is said to be locally Lipschitz continuous if the functions α_3 and $\frac{\partial B}{\partial x}$ are both locally Lipschitz continuous. For a given RCBF B, define the set of admissible control inputs at each point $x \in \mathrm{Int}(\mathcal{C})$ as

$$K_{\mathrm{rcbf}}(x) := \left\{ u \in U \mid L_f B(x) + L_g B(x)\, u - \alpha_3(h(x)) \leq 0 \right\}. \tag{3.13}$$

Selecting control values from this set ensures forward invariance of $\mathcal{C}$, as formalized in the following corollary:

corollaire 3.1 *[12] Let $\mathcal{C} \subset \mathbb{R}^n$ be a set defined by (3.2), and let B be an associated locally Lipschitz continuous RCBF for the system (3.1). Then, any feedback controller $u : \mathrm{Int}(\mathcal{C}) \to \mathcal{U}$ satisfying $u(x) \in K_{\mathrm{rcbf}}(x)$ for all $x \in \mathrm{Int}(\mathcal{C})$ renders the set $\mathrm{Int}(\mathcal{C})$ forward invariant.*

Definition 3.3 of extended class-$\mathcal{K}$ functions naturally leads to the second major class of control barrier functions: the zeroing control barrier function (ZCBF).

Definition 3.10 (Zeroing Control Barrier Function(ZCBF)) *[12] Let $\mathcal{C} \subset \mathcal{X} \subseteq \mathbb{R}^n$ be a set defined as the superlevel set of a continuously differentiable function $h : \mathbb{R}^n \to \mathbb{R}$ via* (3.2)*. The function h is called a* zeroing control barrier function *(ZCBF) on $\mathcal{X}$ if there exists an extended class-$\mathcal{K}$ function $\alpha : \mathbb{R} \to \mathbb{R}$ such that*

$$\sup_{u \in \mathcal{U}} \left[L_f h(x) + L_g h(x)\, u + \alpha(h(x)) \right] \geq 0, \quad \forall x \in \mathcal{X}. \tag{3.14}$$

The ZCBF h is said to be locally Lipschitz continuous if both α and ∇h are locally Lipschitz continuous on $\mathcal{X}$.

For a given ZCBF h, define the set of admissible control inputs at each point $x \in \mathcal{X}$ as

$$K_{\text{zcbf}}(x) := \left\{ u \in \mathcal{U} \mid L_f h(x) + L_g h(x)\, u + \alpha(h(x)) \geq 0 \right\}. \tag{3.15}$$

The following result guarantees forward invariance of the safe set under any controller selecting values from this set:

corollaire 3.2 *Let $\mathcal{C} \subset \mathbb{R}^n$ be a set defined by* (3.2)*, and let h be a locally Lipschitz continuous ZCBF on $\mathcal{X} \subset \mathcal{C}$. Then, any Lipschitz continuous feedback controller $u : \mathcal{X} \to \mathcal{U}$ satisfying $u(x) \in K_{\text{zcbf}}(x)$ for all $x \in \mathcal{X}$ renders the set $\mathcal{C}$ forward invariant.*

Remark 3.3 *Note that while the application of a ZCBF ensures forward invariance of $\mathcal{C}$, it does not guarantee forward completeness of the closed-loop system. Specifically, if $x_0 \in \text{Int}(\mathcal{C})$, then $x(t) \in \mathcal{C}$ for all $t \in [0, \tau_{\max})$, where $\tau_{\max} = \sup I(x_0)$ is the maximal interval of existence. Forward completeness ($\tau_{\max} = \infty$) requires additional assumptions, such as global Lipschitz continuity or boundedness of trajectories.*

3.3 High-Order Control Barrier Functions

The barrier function frameworks discussed so far assume that the time derivative of the barrier function explicitly depends on the control input. However, in many robotic systems such as those modeled by double integrators, unicycles, or other higher-order dynamics, the control input does not appear in the first derivative of $h(x)$. In such cases, classical control barrier functions (CBFs) are not directly applicable. To overcome this limitation, high-order barrier functions (HOBFs) and their control-affine counterparts, high-order control barrier functions (HOCBFs), extend the CBF framework by incorporating higher-order time derivatives of $h(x)$ until the control input appears. This extension allows for the systematic design of safe controllers for a wide range of nonlinear systems with relative degrees greater than one, while maintaining

the formal safety guarantees inherent in the CBF framework.

Given a function $h : \mathbb{R}^n \to \mathbb{R}$ that is r-times continuously differentiable, and a set of sufficiently smooth extended class-$\mathcal{K}$ functions $\alpha_1(\cdot), \alpha_2(\cdot), \ldots, \alpha_r(\cdot)$, we define a sequence of auxiliary functions:

$$\psi_0(x) = h(x), \quad \psi_k(x) = \frac{d}{dt}\psi_{k-1}(x) + \alpha_k(\psi_{k-1}(x)), \quad 1 \leq k \leq r, \tag{3.16}$$

along with the corresponding superlevel sets:

$$\mathcal{C}_{\psi_{k-1}} = \{x \in \mathbb{R}^n \mid \psi_{k-1}(x) \geq 0\}. \tag{3.17}$$

Definition 3.11 (High-Order Barrier Function(HOBF)) *[17] A r-times continuously differentiable function $h : \mathbb{R}^n \to \mathbb{R}$ is called a* high-order barrier function *of degree r for the system* (3.1) *if there exist differentiable extended class-$\mathcal{K}$ functions $\alpha_k : \mathbb{R} \to \mathbb{R}$, for $k = 1, 2, \ldots, r$, and an open set $\mathcal{X} \subset \mathbb{R}^n$ such that*

$$\mathcal{C} := \bigcap_{k=1}^{r} \mathcal{C}_{\psi_{k-1}} \subseteq \mathcal{X}, \tag{3.18}$$

and

$$\psi_r(x) \geq 0, \quad \forall x \in \mathcal{X}, \tag{3.19}$$

where the functions $\psi_k(x)$ are recursively defined as in (3.16).

The recursive construction of the functions $\psi_k(x)$ and their associated super-level sets provides a systematic means to propagate safety constraints through higher derivatives of $h(x)$. The following result establishes that if these conditions are satisfied on an open domain $\mathcal{X}$, then the resulting safe set $\mathcal{C}$ is forward invariant under the autonomous dynamics(3.1).

Proposition 3.1 *[17] Let $h : \mathbb{R}^n \to \mathbb{R}$ be a high-order barrier function of degree r for the autonomous system* (3.1)*, with associated functions $\psi_k(x)$ and safe set $\mathcal{C} = \bigcap_{k=0}^{r-1}\{x \mid \psi_k(x) \geq 0\}$. Suppose that f is locally Lipschitz continuous and that the extended class-$\mathcal{K}$ functions α_k are continuously differentiable. If $\psi_r(x) \geq 0$ for all x in an open set $\mathcal{X} \subseteq \mathcal{C}$, then the set $\mathcal{C}$ is forward invariant.*

The recursive structure of high-order barrier functions not only ensures forward invariance of the safe set but, under appropriate conditions, also induces asymptotic convergence toward its interior. This dual capability is formalized in the following result:

Proposition 3.2 *Let $h : \mathcal{X} \to \mathbb{R}$ be a high-order barrier function of degree r for the autonomous system* (3.1)*, with $\mathcal{X} \subset \mathbb{R}^n$ open and f locally Lipschitz continuous. Define the safe set as*

$$\mathcal{C} := \bigcap_{k=0}^{r-1} \{x \in \mathcal{X} \mid \psi_k(x) \geq 0\}, \tag{3.20}$$

where the functions ψ_k are constructed as in (3.16)*. If $\mathcal{C}$ is nonempty and contains an equilibrium point of the system, then $\mathcal{C}$ is asymptotically stable.*

Remark 3.4 *Unlike classical barrier functions which only guarantee that trajectories remain within a safe set. Proposition 3.2 establishes that high-order barrier functions inherently encode both safety and asymptotic stability. This property eliminates the need for separate Lyapunov-based stabilization mechanisms in autonomous systems, thereby unifying safety and performance objectives within a single theoretical framework.*

Definition 3.12 (Relative Degree) *[15] Let $h : \mathbb{R}^n \to \mathbb{R}$ be a sufficiently differentiable function. The* relative degree *of h with respect to the control-affine system* (3.5) *is defined as the smallest integer $r \geq 1$ such that the r-th Lie derivative of h along the system dynamics depends explicitly on the control input u, i.e.,*

$$L_g L_f^{r-1} h(x) \neq 0, \quad \text{and} \quad L_g L_f^k h(x) = 0 \quad \text{for all } k = 0, 1, \ldots, r-2. \tag{3.21}$$

Equivalently, it is the number of times one must differentiate $h(x)$ along the system trajectories until the control input u appears explicitly in the expression.

Definition 3.13 (Least Relative Degree) *[17] Given an arbitrary open set $\mathcal{X} \subset \mathbb{R}^n$, a r-times continuously differentiable function $h : \mathbb{R}^n \to \mathbb{R}$ is said to have* least relative degree r *with respect to the system* (3.5) *if*

$$L_g L_f^k h(x) = 0, \quad \forall x \in \mathcal{X}, \quad k = 0, 1, \ldots, r-2, \tag{3.22}$$

and

$$L_g L_f^{r-1} h(x) \neq 0, \quad \forall x \in \mathcal{X}. \tag{3.23}$$

The condition of least relative degree is significantly weaker than that of uniform relative degree, as defined in [15, 49]. While the latter requires the Lie derivative $L_g L_f^{r-1} h(x)$ to be nonzero everywhere (i.e., uniformly), the least relative degree condition only requires this for points within the domain $\mathcal{X}$, without imposing global non-vanishing constraints. This relaxation allows for broader applicability in systems where the control input does not appear in the first $r-1$ derivatives of h over the entire state space, but does so locally within a region of interest.

Formally, a high-order control barrier function is then defined as follows:

Definition 3.14 (High-Order Control Barrier Function) *[17] Consider the control-affine system* (3.5) *and a function* $h : \mathbb{R}^n \to \mathbb{R}$ *that is* r*-times continuously differentiable. The function* h *is called a* high-order control barrier function *(HOCBF) of order* r *if there exist differentiable extended class-*$\mathcal{K}$ *functions* $\alpha_k : \mathbb{R} \to \mathbb{R}$*, for* $k = 1, 2, \ldots, r$*, and an open set* $\mathcal{X} \subset \mathbb{R}^n$ *such that:*

1. h *has least relative degree* r *in* $\mathcal{X}$ *(as defined in Definition 3.13),*
2. *For all* $x \in \mathcal{X}$*,*

$$\sup_{u \in \mathcal{U}} \left[L_f \psi_{r-1}(x) + L_g \psi_{r-1}(x)\, u + \alpha_r(\psi_{r-1}(x)) \right] \geq 0, \tag{3.24}$$

where the auxiliary functions $\psi_k(x)$ *are recursively defined as in* (3.16)*, and the safe set is given by*

$$\mathcal{C} := \bigcap_{k=0}^{r-1} \left\{ x \in \mathcal{X} \mid \psi_k(x) \geq 0 \right\}. \tag{3.25}$$

Remark 3.5 *When the order is set to* $r = 1$*, the high-order control barrier function reduces to the zeroing control barrier function (ZCBF). Specifically, the auxiliary function becomes* $\psi_0(x) = h(x)$*, and the HOCBF condition* (3.24) *simplifies to* $\sup_{u \in \mathcal{U}} \left[L_f h(x) + L_g h(x)\, u + \alpha_1(h(x)) \right] \geq 0$*, which is precisely the ZCBF condition given in Definition 3.10.*

Similar to Proposition 3.1, the following result guarantees the forward invariance of the safe set $\mathcal{C}$. Given a high-order control barrier function h, we define, for each $x \in \mathcal{X}$, the set of admissible control inputs as

$$K_{\text{hocbf}}(x) := \left\{ u \in \mathcal{U} \mid L_f \psi_{r-1}(x) + L_g \psi_{r-1}(x)\, u + \alpha_r(\psi_{r-1}(x)) \geq 0 \right\}. \tag{3.26}$$

Theorem 3.4 *[17] Let* $h : \mathbb{R}^n \to \mathbb{R}$ *be a high-order control barrier function (HOCBF) of order* r*, with associated auxiliary functions* $\psi_k(x)$ *defined as in* (3.16)*. Let the safe set be defined as*

$$\mathcal{C} := \bigcap_{k=0}^{r-1} \left\{ x \in \mathbb{R}^n \mid \psi_k(x) \geq 0 \right\}. \tag{3.27}$$

Then, any locally Lipschitz continuous feedback controller $u : \mathbb{R}^n \to \mathbb{R}^m$ *such that* $u(x) \in K_{\text{hocbf}}(x)$ *for all* $x \in \mathbb{R}^n$ *renders the set* $\mathcal{C}$ *forward invariant under the dynamics of system* (3.5)*.*

3.4 Integration with Multi-Robot Systems

In multi-robot systems (MRS), ensuring safety, particularly collision avoidance among agents and with static obstacles, while executing cooperative tasks such as formation control, coverage, or target tracking is a fundamental challenge. Traditional reactive methods often lack formal guarantees, while planning-based approaches struggle with dynamic environments and scalability. Control Barrier Functions (CBFs) have emerged as a powerful framework to address this challenge by providing real-time, formally verifiable safety guarantees that are compatible with high-level task objectives. In this section, we extend the CBF formalism to multi-robot systems with high-order dynamics using High-Order Control Barrier Functions (HOCBFs), enabling safe coordination under realistic robotic models.

3.4.1 System Model and Safety Objectives

Consider a team of N robots operating in a shared workspace $\mathcal{W} \subseteq \mathbb{R}^d$, where each robot $i \in \{1, \dots, N\}$ is governed by a control-affine, high-order dynamical system:

$$\dot{x}_i = f_i(x_i) + g_i(x_i)u_i, \tag{3.28}$$

where $x_i \in \mathbb{R}^{n_i}$ is the state of robot i, $u_i \in \mathbb{R}^{m_i}$ is its control input, and f_i, g_i are sufficiently smooth vector fields. Common examples include:

1. *Double integrator*: $x_i = [p_i^\top, v_i^\top]^\top$, $\dot{p}_i = v_i$, $\dot{v}_i = u_i$,
2. *Unicycle model*: $x_i = [p_i^\top, \theta_i]^\top$, $\dot{p}_i = v_i[\cos\theta_i, \sin\theta_i]^\top$, $\dot{\theta}_i = \omega_i$, with $u_i = [v_i, \omega_i]^\top$.

In both cases, the control input does not appear in the first derivative of the position p_i, necessitating the use of HOCBFs.

The primary safety objectives are:

1. **Inter-robot collision avoidance**: $\|p_i - p_j\| \geq d_{\text{safe}}$ for all $i \neq j$,
2. **Obstacle avoidance**: $\|p_i - p_{\text{obs},k}\| \geq r_{\text{safe}}$ for all obstacles k,

where $d_{\text{safe}} > 0$ and $r_{\text{safe}} > 0$ are safety margins. These constraints define a global safe set $\mathcal{C} \subset \mathbb{R}^n$, with $n = \sum_{i=1}^N n_i$.

3.4.2 Barrier Function Construction for Multi-Robot Safety

For each pairwise safety constraint between robots i and j, we define a candidate barrier function:

$$h_{ij}(x) = \|p_i - p_j\|^2 - d_{\text{safe}}^2. \tag{3.29}$$

Similarly, for robot i and obstacle k:

$$h_i^{\text{obs},k}(x) = \|p_i - p_{\text{obs},k}\|^2 - r_{\text{safe}}^2. \tag{3.30}$$

Each h is smooth and defines a safe set $\mathcal{C}_h = \{x \mid h(x) \geq 0\}$. The global safe set is the intersection of all such sets:

$$\mathcal{C} = \bigcap_{i<j} \mathcal{C}_{h_{ij}} \cap \bigcap_{i,k} \mathcal{C}_{h_i^{\text{obs},k}}. \tag{3.31}$$

However, for robots with high-order dynamics (e.g., unicycles), the control input u_i does not appear in $\dot{h}_{ij}$ or $\ddot{h}_{ij}$. For instance, in the unicycle model, u_i first appears in the third derivative of h_{ij}. Thus, the relative degree of h_{ij} with respect to the system dynamics is $r = 3$. This necessitates the use of HOCBFs.

3.4.3 High-Order Control Barrier Functions for Multi-Robot Systems

For each barrier function h (e.g., h_{ij}), we construct the sequence of auxiliary functions ψ_k as in Definition 3.14:

$$\psi_0 = h, \quad \psi_k = \dot{\psi}_{k-1} + \alpha_k(\psi_{k-1}), \quad k = 1, \ldots, r,$$

where α_k are extended class-$\mathcal{K}$ functions. The HOCBF condition for robot i with respect to constraint h is then:

$$L_{f_i}\psi_{r-1}^{(i)}(x) + L_{g_i}\psi_{r-1}^{(i)}(x)\, u_i + \alpha_r(\psi_{r-1}^{(i)}(x)) \geq 0, \tag{3.32}$$

where $\psi_{r-1}^{(i)}$ denotes the $(r-1)$-th auxiliary function as seen from robot i's dynamics. Note that $\psi_{r-1}^{(i)}$ depends on the states of neighboring robots (e.g., x_j for h_{ij}), making the constraint *coupled*.

For each robot i, the set of admissible control inputs is:

$$K_{\text{hocbf}}^{(i)}(x) = \bigcap_{\substack{j \neq i \\ k}} \left\{ u_i \in \mathcal{U}_i \mid \text{HOCBF condition (3.32) holds for } h_{ij}, h_i^{\text{obs},k} \right\}. \tag{3.33}$$

This set is convex if the HOCBF conditions are affine in u_i which they are for control-affine systems.

3.4.4 Decentralized Safety-Critical Control Synthesis

To achieve safe coordination, each robot i solves a local quadratic program (QP) that minimally modifies a nominal control input $u_{i,\text{nom}}$ (e.g., from a formation controller) while satisfying all HOCBF constraints:

$$\begin{aligned} \min_{u_i} \quad & \|u_i - u_{i,\text{nom}}\|^2 \\ \text{s.t.} \quad & u_i \in K^{(i)}_{\text{hocbf}}(x). \end{aligned} \tag{3.34}$$

This QP is solved in real time using only local information (states of neighbors).

By Theorem 3.4, if the QP is feasible for all i and the resulting control law $u_i(x)$ is locally Lipschitz continuous, then the global safe set $\mathcal{C}$ is forward invariant. Feasibility is guaranteed if the interaction graph is connected and safety margins are respected initially [14].

3.5 Case Study: Application of High-Order Control Barrier Functions

To demonstrate the effectiveness of the proposed HOCBF-based control framework, we consider a team of five homogeneous double-integrator robots navigating within a bounded planar workspace containing static rectangular obstacles. Starting from arbitrary initial configurations, the robots are required to reach predefined goal positions while avoiding inter-robot collisions, obstacle collisions, and workspace boundary violations. Each agent independently solves a decentralized HOCBF-based quadratic program (QP) in real time, ensuring that all safety constraints are continuously satisfied, as verified by the non-negativity of the associated barrier functions.

3.5.1 System Dynamics and Environment

Each robot $i \in \{1, \dots, 5\}$ is modeled by the double-integrator dynamics:

$$\dot{p}_i = v_i, \quad \dot{v}_i = u_i, \tag{3.35}$$

where $p_i, v_i, u_i \in \mathbb{R}^2$ denote the position, velocity, and control input (acceleration) of robot i, respectively.
The overall system state is

$$x = [p_1^\top, v_1^\top, \dots, p_5^\top, v_5^\top]^\top \in \mathbb{R}^{20}.$$

Remark 3.6 *The multi-robot dynamics in* (3.35) *can be expressed in the standard control-affine form*

$$\dot{x} = f(x) + g(x)\,u, \tag{3.36}$$

where $x = [p_1^\top, v_1^\top, \ldots, p_5^\top, v_5^\top]^\top \in \mathbb{R}^{20}$ *and* $u = [u_1^\top, \ldots, u_5^\top]^\top \in \mathbb{R}^{10}$ *represent the global state and control input vectors, respectively. Each robot contributes four state variables (two position and two velocity components), yielding a total state dimension of* 20 *for the five-agent system.*

The drift and input vector fields are block-diagonal and given by

$$f(x) = \begin{bmatrix} v_1 \\ 0 \\ v_2 \\ 0 \\ \vdots \\ v_5 \\ 0 \end{bmatrix} \in \mathbb{R}^{20}, \qquad g(x) = \operatorname{diag}\left(\begin{bmatrix} 0 \\ I_2 \end{bmatrix}, \begin{bmatrix} 0 \\ I_2 \end{bmatrix}, \ldots, \begin{bmatrix} 0 \\ I_2 \end{bmatrix} \right) \in \mathbb{R}^{20\times 10}, \tag{3.37}$$

where I_2 *denotes the* 2×2 *identity matrix. This formulation explicitly reveals the linearity of the system with respect to the control input and its decoupled structure across agents, which enables decentralized control synthesis and distributed QP implementation.*

Robots evolve within a bounded square workspace

$$\mathcal{W} = [0, 10]^2 \subset \mathbb{R}^2,$$

containing two static obstacles and four virtual boundaries acting as walls.
Each robot i, $i \in \{1, \cdots, 5\}$ starts from rest ($v_i(0) = [0, 0]^\top$) and follows nontrivial paths from its initial position $p_i(0)$ to its goal p_i^{des}.

3.5.2 Safety Constraints and HOCBF Design

Safety is ensured through HOCBFs addressing inter-robot separation, obstacle avoidance, and workspace limits.

1. **Inter-robot collision avoidance:**

$$h_{ij}(x) = \|p_i - p_j\|^2 - d_{\text{safe}}^2, \quad i < j, \tag{3.38}$$

where d_{safe}. represents the minimum allowable distance between any two robots.

2. **Obstacle avoidance:**
   For each robot i and obstacle $\mathcal{O}_k$,

$$h_i^{\text{obs},k}(x) = \text{dist}(p_i, \mathcal{O}_k)^2 - r_{\text{safe}}^2, \tag{3.39}$$

   where $\text{dist}(p_i, \mathcal{O}_k)$ is the Euclidean distance between the robot and the closest point on the obstacle $\mathcal{O}_k$, and r_{safe}defines the safety margin.

3. **Workspace boundaries:**
   These are enforced implicitly through virtual wall obstacles surrounding the workspace.

Since the dynamics (3.35) are second-order, each barrier function has a *relative degree of two*. Following the standard HOCBF construction, auxiliary functions are defined as

$$\begin{aligned}
\psi_0 &= h, \\
\psi_1 &= \dot{h} + \alpha_1(h), \\
\psi_2 &= \dot{\psi}_1 + \alpha_2(\psi_1),
\end{aligned}$$

where $\alpha_1(s) = \alpha_2(s) = \gamma s$ with $\gamma > 0$.
In the decentralized framework, each robot enforces its own safety constraints through a local HOCBF condition that accounts for both obstacle and inter-robot interactions. For barriers associated with obstacles or workspace boundaries, which depend only on the state of robot i, the HOCBF constraint takes the standard single-agent form

$$L_f\psi_1^{(i)}(x) + L_{g_i}\psi_1^{(i)}(x)\, u_i + \alpha_2\Big(\psi_1^{(i)}(x)\Big) \geq 0. \tag{3.40}$$

For inter-robot safety constraints, the barrier function $h_{ij}(x)$ depends on the relative states of robots i and j, and its derivative involves both control inputs u_i and u_j. To enable decentralized computation, each robot substitutes the neighboring control input u_j with an estimated value $\hat{u}_j$ obtained from the previous time step or a local communication exchange. The resulting decentralized HOCBF constraint for robot i is expressed as

$$L_f\psi_1^{ij}(x) + L_{g_i}\psi_1^{ij}(x)\, u_i + \alpha_2\Big(\psi_1^{ij}(x)\Big) \geq -\, L_{g_j}\psi_1^{ij}(x)\, \hat{u}_j, \tag{3.41}$$

which defines an affine inequality in the control variable u_i. This formulation preserves forward invariance of the safe set while allowing each robot to solve its own quadratic program (QP) independently, thus achieving fully decentralized safety enforcement.

3.5.3 Control Synthesis and Simulation Setup

At each control step, each robot independently computes its control input by solving a local quadratic program (QP) of the form

$$\begin{aligned} \min_{u_i \in \mathbb{R}^2} \quad & \|u_i - u_{i,\text{nom}}\|^2 \\ \text{s.t.} \quad & \text{HOCBF constraints (3.40) and (3.41) hold for all active barriers.} \end{aligned} \tag{3.42}$$

The objective function minimizes the deviation of the actual control input u_i from the nominal command $u_{i,\text{nom}}$, thereby promoting smooth control actions while reducing unnecessary control effort. The nominal input is designed as a proportional–derivative (PD) feedback term that drives each robot toward its desired goal position:

$$u_{i,\text{nom}} = -k_p(p_i - p_i^{\text{des}}) - k_v v_i, \tag{3.43}$$

where $k_p > 0$ and $k_v > 0$ represent the proportional and derivative gains, respectively. This quadratic program (QP) formulation ensures that the safety constraints derived from the HOCBF framework are rigorously enforced while guaranteeing convergence to the target under continuous, energy-efficient control inputs.

Remark 3.7 *A proportional–derivative (PD) controller is adopted in this study, rather than a full proportional–integral–derivative (PID) controller, due to several considerations specific to the double-integrator dynamics and the requirements of multi-robot navigation. The integral term in a PID controller is generally introduced to eliminate steady-state errors resulting from constant disturbances or model bias, which are prevalent in first-order or position-only systems. In contrast, for the double-integrator model employed here, where each robot's state includes both position and velocity, it is well established that zero steady-state error in position can be achieved with proportional and derivative feedback alone, provided that the gains are appropriately selected.*

Furthermore, the inclusion of an integral action may introduce adverse effects such as increased overshoot, slower transient response, and possible instability, particularly in the presence of hard safety constraints enforced by HOCBFs and quadratic programming. Given that the main objectives are fast convergence to the goal, smooth velocity profiles, and strict safety guarantees, the PD controller provides an effective and sufficiently robust solution without requiring integral compensation.

3.5.4 Results and Discussion

The simulation is performed over a total horizon of T (s) with a discrete integration time step Δt. Each robot is equipped with a fixed sensing radius R (m), such that

another robot j is included in its neighbor set $\mathcal{N}_i$ if and only if $\|p_i - p_j\| \leq R$. The initial deployment is chosen to ensure that the sensing graph is connected, and this property is maintained throughout the simulation by virtue of the imposed HOCBF-based safety constraints. As an initial step, Fig. 3.1 illustrates the initial and final positions of all robots, as well as the placement of obstacles and workspace boundaries.

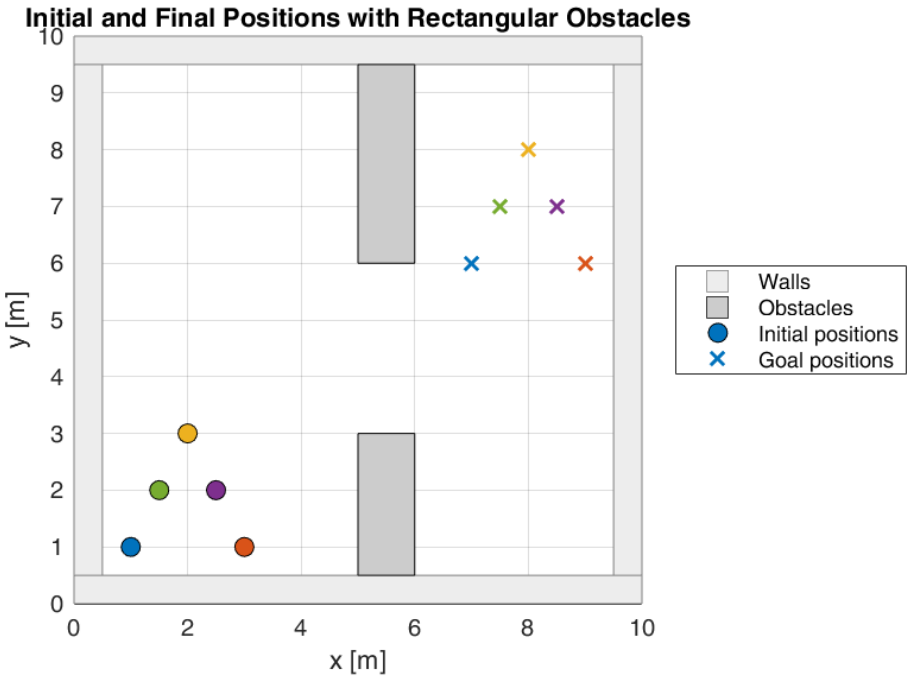

Figure 3.1: Initial and final positions of the five robots in the workspace, with obstacles and walls represented as shaded areas.

As seen in Fig. 3.1, the robots are deployed in the lower-left region and assigned nontrivial goal locations to induce intersecting paths that necessitate both collision avoidance and obstacle negotiation.
Figure 3.2 shows the complete trajectories followed by all robots as they navigate toward their respective goals while avoiding obstacles and maintaining safe inter-robot distances.

It is evident from Fig. 3.2 that the robots successfully adjust their paths to avoid both static obstacles and one another, resulting in smooth and collision-free trajectories throughout the workspace.
To assess safety, Fig. 3.3 presents the minimum value of the inter-robot barrier function $h_{ij}(t)$ over time.

As indicated in Fig. 3.3, the inter-robot barrier function remains strictly positive for the entire duration, confirming that collision avoidance constraints are satisfied at all times. Figure 3.4 reports the minimum value of the obstacle barrier function $h_{\text{obs}}(t)$ during the simulation.

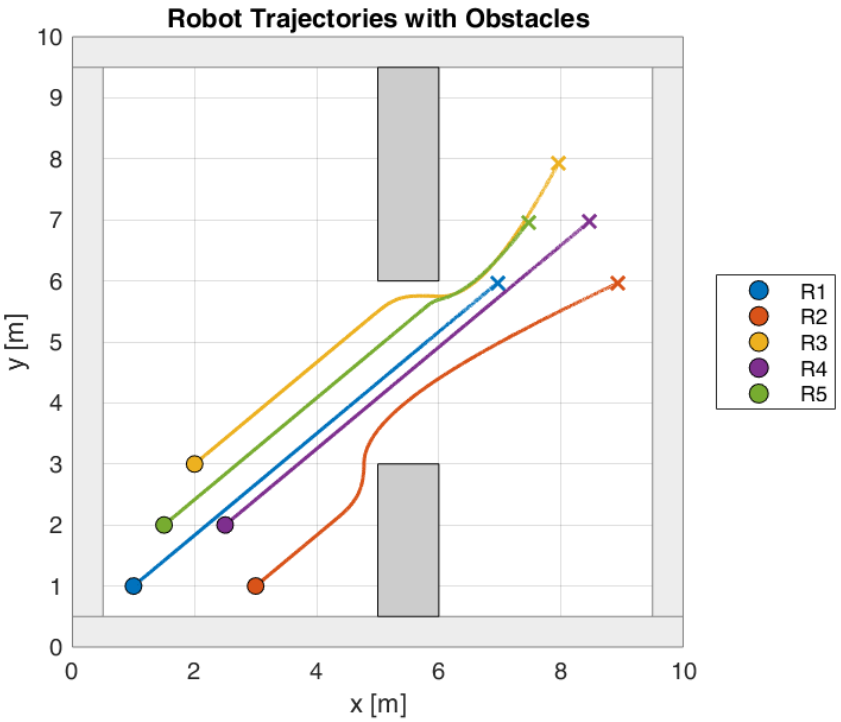

Figure 3.2: Trajectories of all robots from initial to goal positions, demonstrating safe navigation and obstacle avoidance.

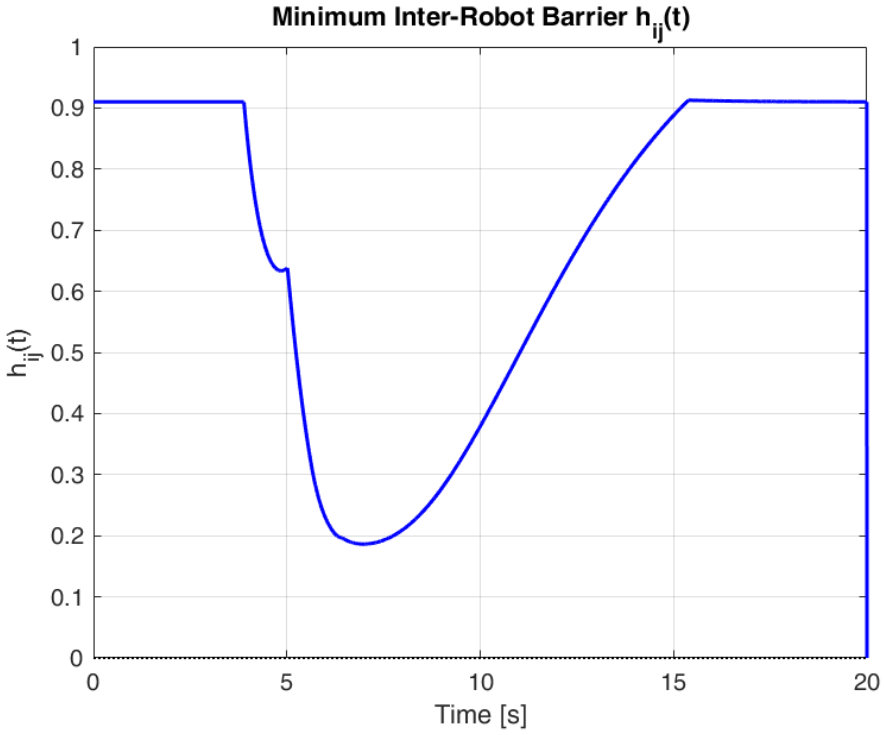

Figure 3.3: Minimum value of the inter-robot barrier function $h_{ij}(t)$ versus time. The safety boundary ($h_{ij} = 0$) is indicated as a dashed line.

The results in Fig. 3.4 demonstrate that the robots maintain a safe distance from all obstacles, as the minimum barrier function does not cross the safety threshold at

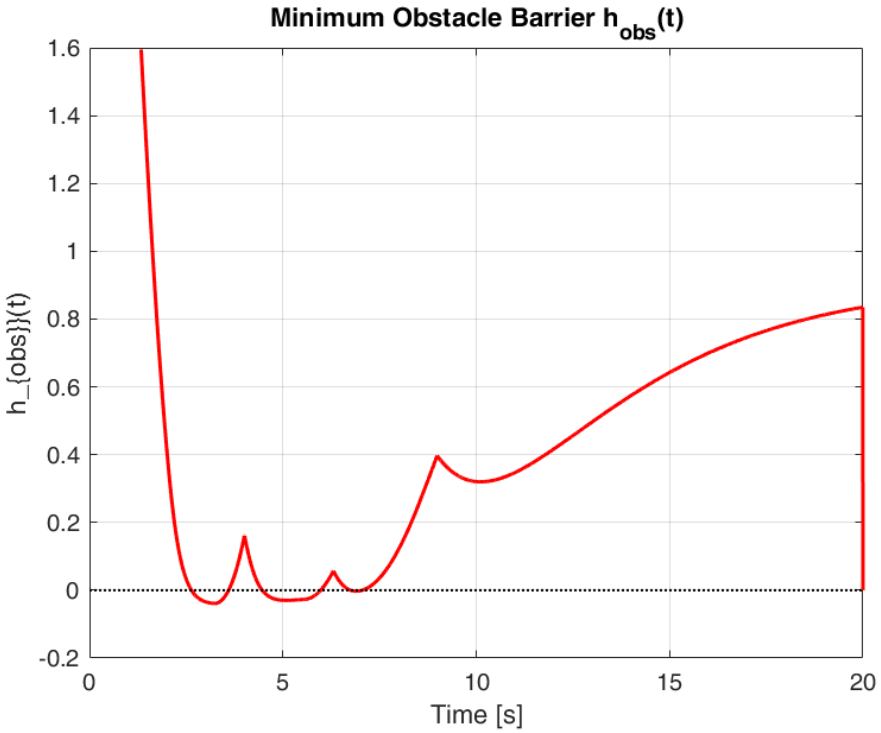

Figure 3.4: Minimum obstacle barrier function $h_{\text{obs}}(t)$ over time, with the safety threshold ($h_{\text{obs}} = 0$) indicated.

any time.
Finally, Fig. 3.5 plots the remaining distance to the goal for each robot as a function of time.

As shown in Fig. 3.5, all robots converge smoothly to their target positions, achieving zero steady-state error by the end of the simulation.
Additionally, Fig. 3.6 displays the control effort $\|u_i(t)\|$ exerted by each robot during the simulation.

Fig. 3.6 highlights that the control inputs remain within reasonable bounds, indicating energy-efficient and stable navigation.

3.6 Conclusion

This study has demonstrated the effectiveness of a decentralized HOCBF-based control strategy for safe multi-robot navigation under static obstacles and strict inter-robot safety constraints. By leveraging real-time solutions to local quadratic programs, each robot achieves collision avoidance, obstacle negotiation, and convergence to its goal with minimal coordination overhead. Simulation results confirm that the proposed approach maintains safety and generates smooth, efficient trajectories for all agents.

However, several challenges were identified during this work. The conservative nature of barrier function tuning can restrict maneuverability near obstacles, and se-

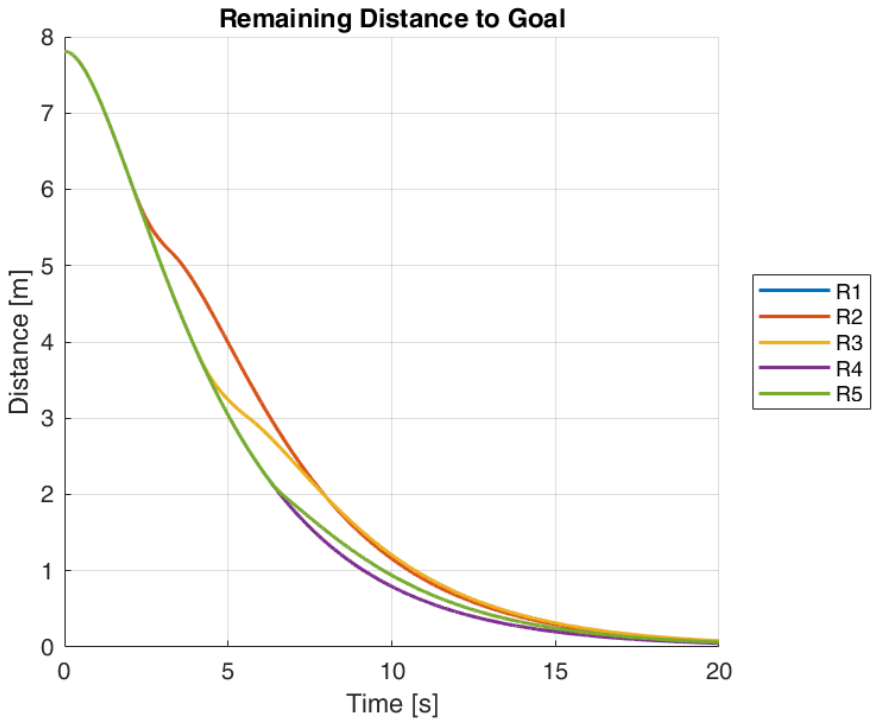

Figure 3.5: Time evolution of the remaining distance to the goal for each robot.

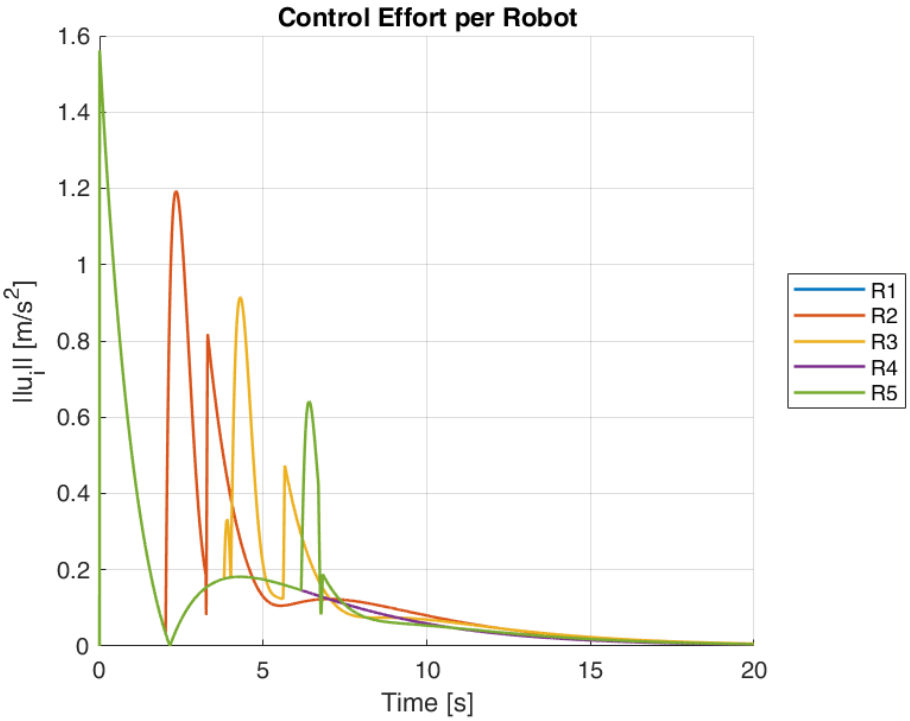

Figure 3.6: Control effort (norm of the input acceleration) for each robot as a function of time.

lecting appropriate HOCBF gains requires careful trade-offs between safety and efficiency. Furthermore, the current framework assumes perfect state information and

may be sensitive to uncertainties, sensor noise, or communication delays in practical deployments.

Future research directions include extending the method to enable navigation in unknown or partially known environments, where online mapping and simultaneous localization and mapping (SLAM) techniques become essential. The incorporation of neural networks and learning-based components, such as reinforcement learning or data-driven safety filters, offers the potential to enhance adaptability and robustness to model uncertainties and dynamic changes. Additionally, investigating the effects of communication constraints and partial observability, as well as conducting experimental validation on physical robotic platforms, will be critical steps toward real-world application and scalability.

Chapter 4

HOCBF-Based Safe Formation Control for Multi-Robot Systems

4.1 Introduction

In recent years, the issue of multi-robot formation control has gained substantial attention, especially in systems characterized by double integrator dynamics. This model is valuable because it accurately represents the inertial characteristics in robotic navigation, making it ideal for high-speed and agile maneuvers. A primary challenge in multi-robot formation navigation lies in ensuring that the robots maintain a predefined formation while maneuvering through dynamic and potentially cluttered environments. Traditionally, many methods have assumed a rigid formation, where the relative distances between robots remain constant. However, rigid formations can be too restrictive, particularly in situations where the environment contains obstacles or when the formation needs to adapt in response to task-specific goals.

4.1.1 Formation Deformation

Recent studies have introduced the concept of formation deformation, which enables a formation to gradually alter its shape. In this framework, the desired relative offsets between robots are permitted to change over time. For instance, one can use a transition function to interpolate between two formation configurations. This approach offers greater flexibility, allowing the formation to adapt dynamically during navigation. As a result, the robot team can respond to shifting environments or optimize additional performance metrics.

4.1.2 Control Barrier Functions (CBF):

To ensure safety, such as avoiding collisions, the concept of Control Barrier Functions (CBFs) has become widely adopted. CBFs offer a structured approach to enforce state constraints, guaranteeing that safety-critical conditions, such as maintaining a minimum distance between robots, are always satisfied. In numerous studies, the CBF framework is integrated with Quadratic Programming (QP) to adapt the nominal control input in real-time, resulting in control actions that adhere to the imposed safety constraints.

Given the complexity and high-dimensional nature of multi-robot systems, centralized optimization quickly becomes computationally expensive. To mitigate this issue, recent research has focused on distributed optimization techniques, where each robot solves a local QP problem based on its own state and limited information from neighboring robots. A typical local QP problem for each robot aims to minimize a cost function that combines:

- A control deviation term, which ensures each robot follows the nominal control input (typically derived from a PD controller applied to the formation center).
- A formation term that minimizes deviations from the desired time-varying relative offsets, while accounting for the deformation of the formation.

This distributed approach enables the system to scale to larger teams and enhances resilience to communication delays or failures. Additionally, by utilizing consensus-based techniques, robots can coordinate their actions effectively without requiring a central coordinator.

4.1.3 Related Work

The problem of multi-robot formation control has been extensively studied, with applications spanning from autonomous vehicles to swarm robotics. Recent advancements in research have built upon traditional rigid formation approaches by introducing flexibility in formation shapes, allowing adaptation to dynamic environments.

Formation control remains a fundamental topic in multi-robot systems, with numerous methods proposed to maintain desired formations, particularly in dynamic settings. Early works concentrated on rigid formations, where the relative positions between robots are fixed. Techniques such as consensus algorithms and potential fields have been widely used in these contexts [52]. While efficient, these methods may not be applicable in scenarios where the formation needs to adjust, especially when robots must modify their formation shape.

The concept of formation deformation was introduced to enable robots to modify their formation shape over time, thus allowing for adaptation to obstacles and

changing environments. This approach facilitates gradual transitions between different formation shapes, enhancing flexibility and adaptability. Recent studies have proposed interpolation methods for transitioning between formations, often employing transition functions to regulate the deformation process. In [53], Control Barrier Functions (CBFs) are applied to multi-robot systems, ensuring safety during formation deformation, particularly in dynamic and constrained environments. Additionally, [54] discusses a motion planning strategy for coordinating multiple mobile robots to maintain formation, respect constraints, and avoid collisions. This method utilizes velocity optimization and online adjustments to address emergent scenarios, such as obstacle avoidance. Furthermore, [55] presents a novel strategy for addressing the multi-robot formation problem, employing a decentralized algorithm to guide holonomic robots toward a predefined formation while ensuring avoidance of non-convex and dynamic obstacles, as well as other robots. Local collision avoidance is handled using a modified variant of the ORCA (Optimal Reciprocal Collision Avoidance) algorithm, providing smooth and continuous control.

4.2 Problem Definition

4.2.1 Multi-Robot System with Double Integrator Dynamics

Consider a group of N robots, where each robot $i \in \{1, \ldots, N\}$ is described by a double-integrator dynamic model, described by the system of equations given below:

$$\begin{cases} \dot{p}_i = v_i, \\ \dot{v}_i = u_i, \end{cases} \tag{4.1}$$

where:

- $p_i \in \mathbb{R}^n$ represents the position of robot i,
- $v_i \in \mathbb{R}^n$ represents the velocity of robot i,
- $u_i \in \mathbb{R}^n$ represents the acceleration control input for robot i.

The state vector for each robot is given by $x_i = \begin{bmatrix} p_i^T & v_i^T \end{bmatrix}^T \in \mathbb{R}^{2n}$, with the corresponding dynamics:

$$\dot{x_i} = Fx_i + Gu_i, \tag{4.2}$$

where:

$$F = \begin{bmatrix} 0_{n\times n} & I_n \\ 0_{n\times n} & 0_{n\times n} \end{bmatrix} \in \mathbb{R}^{2n\times 2n}, \quad G = \begin{bmatrix} 0_{n\times n} \\ I_n \end{bmatrix} \in \mathbb{R}^{2n\times n}.$$

The global configuration of the robotic team is represented by the stacked state vector:

$$X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix} \in \mathbb{R}^{2nN}. \tag{4.3}$$

The global dynamics of the multi-robot team are expressed as follows

$$\dot{X} = \mathcal{F}X + \mathcal{G}U, \tag{4.4}$$

where:

$$\mathcal{F} = \begin{bmatrix} F & 0_{n\times n} & \cdots & 0_{n\times n} \\ 0_{n\times n} & F & \cdots & 0_{n\times n} \\ \vdots & \vdots & \ddots & \vdots \\ 0_{n\times n} & 0_{n\times n} & \cdots & F \end{bmatrix} \in \mathbb{R}^{2nN\times 2nN}, \quad \mathcal{G} = \begin{bmatrix} G & 0_{n\times n} & \cdots & 0_{n\times n} \\ 0_{n\times n} & G & \cdots & 0_{n\times n} \\ \vdots & \vdots & \ddots & \vdots \\ 0_{n\times n} & 0_{n\times n} & \cdots & G \end{bmatrix} \in \mathbb{R}^{2nN\times nN}.$$

The control input vector for all robots is given by:

$$U = \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_N \end{bmatrix} \in \mathbb{R}^{nN}. \tag{4.5}$$

This formulation describes the dynamics of a multi-robot system where each robot follows a double integrator model, and the global behavior of the system is governed by the collective states and control inputs.

4.2.2 Desired Formation

A *desired formation* is characterized by the desired relative positions $\delta_{ij}^d \in \mathbb{R}^n$ between robots i and j defined as follows:

$$p_i(t) - p_j(t) = \delta_{ij}^d, \quad \forall (i,j) \in \mathcal{E}, \tag{4.6}$$

where $\mathcal{E}$ represents the edges of the *formation graph*.

Remark 4.1 *In practical scenarios, achieving a perfectly rigid formation is challenging. The formation should be nearly rigid, meaning that the robots must uphold the desired or target formation, while permitting adjustments when required, such as for collision avoidance or trajectory modification to reach the goal.*

The formation constraint can be reformulated as follows[56]:

$$\|p_i(t) - p_j(t) - \delta_{ij}^d\| \leq \varepsilon \quad \forall (i,j) \in \mathcal{E}, \tag{4.7}$$

where ε represents a tolerance that provides flexibility in maintaining the distance between the robots. This constraint allows for small deviations from the desired distance, ensuring that the formation remains cohesive while accommodating necessary adjustments for practical purposes.

4.2.3 Controlled Formation Deformation and Recovery

When navigating through constrained environments, the formation adapts by scaling and rotating the relative offsets to avoid collisions with both obstacles and other robots:

$$\delta_{ij}^{adapt}(t) = \Gamma(t)\delta_{ij}^d, \tag{4.8}$$

where $\Gamma(t) \in \mathbb{R}^{n\times n}$represents the *deformation matrix*.
Once obstacles are avoided, the multi-robot system must temporarily adjust its formation shape and subsequently autonomously return to its original configuration, ensuring stable and guaranteed recovery.

4.2.4 Control Objectives

In the considered formation control problem, the multi-robot system is required to navigate safely from an initial configuration to a designated target while satisfying several critical objectives. The robots must preserve the desired formation structure, allow for controlled deformation when necessary, and subsequently return to the original formation once the maneuver is completed. Throughout the motion, safety constraints must be strictly enforced to guarantee collision avoidance and operational reliability. Moreover, the closed-loop system should ensure stability, while minimizing energy consumption and control effort to achieve efficiency. The framework must also exhibit robustness by adapting to uncertainties and external disturbances, and it should incorporate effective communication strategies to coordinate the robots and maintain cohesive group behavior.

4.3 Control Barrier Functions

In a multi-robot system, ensuring the safety of the robots, particularly with regard to collision avoidance, is essential. Control Barrier Functions (CBFs) provide a robust framework for ensuring safety in dynamic systems by enforcing state constraints that

must be continuously satisfied[53, 57].In this section, we apply CBFs to a system of N robots, each modeled using a double integrator dynamic, to guarantee that the robots maintain a minimum distance from one another and avoid collisions during their motion.

To ensure collision avoidance during movement, we define a Control Barrier Function (CBF) that enforces a minimum distance between each pair of robots. The CBF for each pair of robots i and j is a function $h_{ij} : \mathbb{R}^{2n} \to \mathbb{R}$ defined as:

$$h_{ij}(p_i, p_j) = \|p_i - p_j\|^2 - d_{min}^2, \tag{4.9}$$

where d_{min} represents the desired minimum distance between the robots.
To avoid collisions, the function must always be non-negative:

$$h_{ij}(p_i, p_j) \geq 0 \quad \forall i \neq j. \tag{4.10}$$

Therefore, the control input $u_i(t)$ must be adjusted to maintain the safety of the system. To ensure that the CBF constraint is satisfied over time, we compute the time derivative $\dot{h}_{ij}$ given by:

$$\dot{h}_{ij} = 2(p_i - p_j)^T(v_i - v_j). \tag{4.11}$$

In order to maintain safety, we require that the derivative of the CBF is non-negative:

$$\dot{h}_{ij} \geq -\alpha(h_{ij}), \tag{4.12}$$

where $\alpha(\cdot) : \mathbb{R} \to \mathbb{R}$ is a strictly increasing function, typically selected as a class $\mathcal{K}$ function with $\alpha(0) = 0$ [58], This ensures that when the robots approach one another too closely, the rate of change of $h_{ij}$is constrained, prompting the robots to move away from each other.

4.3.1 Higher-Order Control Barrier Functions

Higher-Order Control Barrier Functions (HOCBFs) [16] are essential for systems with a relative degree greater than one, such as double integrator dynamics, to ensure safety by incorporating higher-order state derivatives (e.g., acceleration inputs) into the constraints. This approach guarantees collision avoidance, even when the control input does not directly influence the first derivative of the safety condition.

$$\ddot{h}_{ij} = \frac{d}{dt}\dot{h}_{ij} = 2\|v_i - v_j\|^2 + 2(p_i - p_j)^T(u_i - u_j) \tag{4.13}$$

The safety constraint thus becomes:

$$\ddot{h}_{ij} + \alpha_1(\dot{h}_{ij}) + \alpha_2(h_{ij}) \geq 0 \tag{4.14}$$

where $\alpha_1(\cdot)$ and $\alpha_2(\cdot)$ are strictly increasing functions, typically selected as class $\mathcal{K}$ functions, with $\alpha_1(0) = 0$ and $\alpha_2(0) = 0$.

4.4 Case Studies in Multi-Robot Formation Control

4.4.1 Formation Preservation in Navigation

We consider a team of N robots that transition from an initial state x^{init} to a final state x^f while maintaining a desired formation along the trajectory. The initial and final states of each robot i are given by:

$$x_i^{init}(0) = \left[p_i(0)^T, v_i(0)^T\right]^T, \quad x_i^f = \left[p_i^f, v_i^f\right]^T$$

To ensure the robots maintain the desired formation, avoid collisions, and minimize control effort, we formulate the problem as a Quadratic Programming (QP) optimization problem:

$$\min_u \frac{1}{2}\sum_{i=1}^{N} \|u_i - u_{des}\|^2 + \lambda \sum_{i<j} \|p_i - p_j - \delta_{ij}^d\|^2 \tag{4.15}$$

Subject to the following constraints:

$$\|p_i - p_j - \delta_{ij}^d\| \leq \varepsilon \quad \forall i \neq j, \tag{4.16}$$

$$\|u_i\| \leq u_{max} \quad \forall i, \tag{4.17}$$

$$2(p_i - p_j)^T(u_i - u_j) \geq -2\|v_i - v_j\|^2 - \alpha_1(\dot{h}_{ij}) - \alpha_2(h_{ij}), \quad \forall i \neq j. \tag{4.18}$$

where u_{des} represents the desired control input, u_{max}denotes the maximum control input, and λ is a weighting factor.

Remark 4.2 *As the number of robots increases, the global quadratic optimization problem in* (4.15) *becomes computationally intensive. To mitigate this, each robot independently solves its own QP problem in a distributed manner, taking into account only local interactions with its neighbors.*

Each robot minimizes its local cost function subject to constraints, while exchanging information with its neighbors (e.g., relative positions p_j, velocities v_j, and desired formation offsets δ_{ij}^d). For each robot i, the distributed optimization problem is formulated as follows[56]:

$$\min_{u_i} \frac{1}{2}\|u_i - u_{des}\|^2 + \lambda \sum_{j \in \mathcal{N}_i} \|p_i - p_j - \delta_{ij}^d\|^2 \tag{4.19}$$

where:

- $\delta_{ij}^d \in \mathbb{R}^n$ is the desired relative position between robots i and j (from the formation graph $\mathcal{E}$),
- $\mathcal{N}_i = \{j \mid (i,j) \in \mathcal{E}\}$ is the set of neighbors of robot i in the formation.

Subject to:

$$\|p_i - p_j - \delta_{ij}^d\| \leq \varepsilon \quad \forall j \in \mathcal{N}_i, \tag{4.20}$$

$$\|u_i\| \leq u_{\max} \quad \forall i, \tag{4.21}$$

$$2(p_i - p_j)^T(u_i - u_j) \geq -2\|v_i - v_j\|^2 - \alpha_1(\dot{h}_{ij}) - \alpha_2(h_{ij}), \quad \forall j \in \mathcal{N}_i. \tag{4.22}$$

Example 4.1 *Consider a team of 5 robots positioned on a circle with center at* $(1,1)$ *and radius* $r = 2$*. The robots navigate from an initial state at* $(1,1)$ *to a final state at* $(10,10)$ *while maintaining the desired formation.*

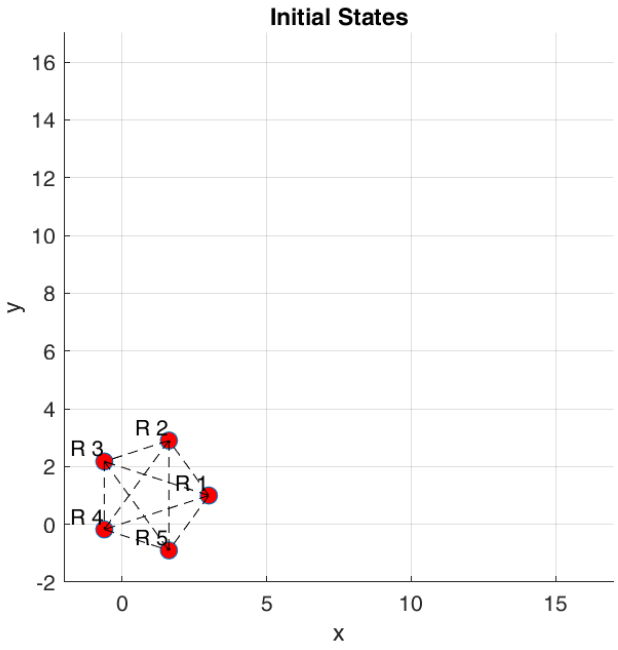

Figure 4.1: Initial Formation

The five-robot pentagon moves rigidly toward the goal while preserving its geometric shape. Non-intersecting paths ensure collision-free motion, with minimal formation error.

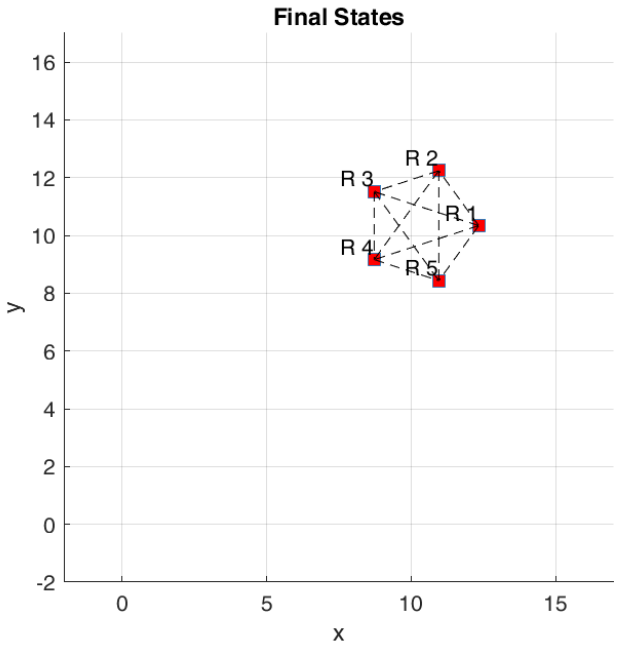

Figure 4.2: Final Formation

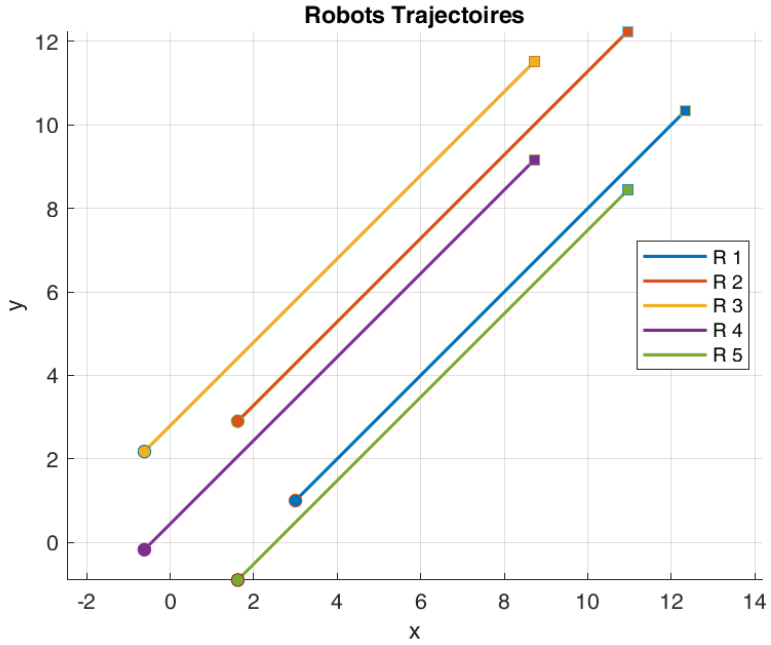

Figure 4.3: Robot Trajectories

4.4.2 Formation Reconfiguration via Scaling and Rotation

In practice, achieving a strictly rigid deformation is not feasible. Therefore, we introduce a deformation tolerance $\varepsilon^{adapt} > 0$ defined as follows[56]:

$$\|\delta_{ij}^{adapt}(t) - \Gamma(t)\delta_{ij}^{d}\| \leq \varepsilon^{adapt} \tag{4.23}$$

Remark 4.3 *The deformation matrix $\Gamma(t)$ remains invertible for all t. This invertibility guarantees that the formation can always be restored to its original shape once external constraints or collision-avoidance maneuvers are no longer necessary.*

For simplicity, we focus on the two-dimensional case. However, the same approach can be extended to higher dimensions $n \geq 2$ without fundamental changes in the methodology.
The deformation matrix is expressed as:

$$\Gamma(t) = s(t) \cdot R(\theta(t)), \tag{4.24}$$

where:

- $s(t)$ is a scalar function (the scale factor) that enables the formation to expand or contract within the plane.
- $R(\theta(t))$ is the rotation matrix, defined as:

$$R(\theta(t)) = \begin{bmatrix} \cos(\theta(t)) & -\sin(\theta(t)) \\ \sin(\theta(t)) & \cos(\theta(t)) \end{bmatrix}.$$

There are various methods to define the time-dependent scale $s(t)$ and rotation $\theta(t)$. Two illustrative cases are:

1. Linear Variations:

$$s(t) = 1 + \beta_1 t, \quad \beta_1 \text{ is a small constant rate of expansion or contraction,}$$

$$\theta(t) = \beta_2 t, \quad \beta_2 \text{ is a constant rotational speed.}$$

2. Sigmoidal Variations:

$$s(t) = \frac{1}{1 + e^{-\beta(t-t_0)}}, \quad \beta > 0, \quad t_0 \text{ represents the midpoint of the switch,}$$

$\theta(t) = \theta_0 + \gamma s(t), \quad \theta_0$ is the initial angle of the formation, $\quad \gamma$ is a rotation factor (or rotation

Example 4.2 *Consider 5 robots initially arranged in a circle with a radius of $r = 2$ and the formation center at $(1, 1)$.In this example, a simple deformation, consisting of both scaling and rotation, is applied to adjust the formation as the robots navigate toward a final state with the center at $(10, 10)$ while maintaining the desired formation.*

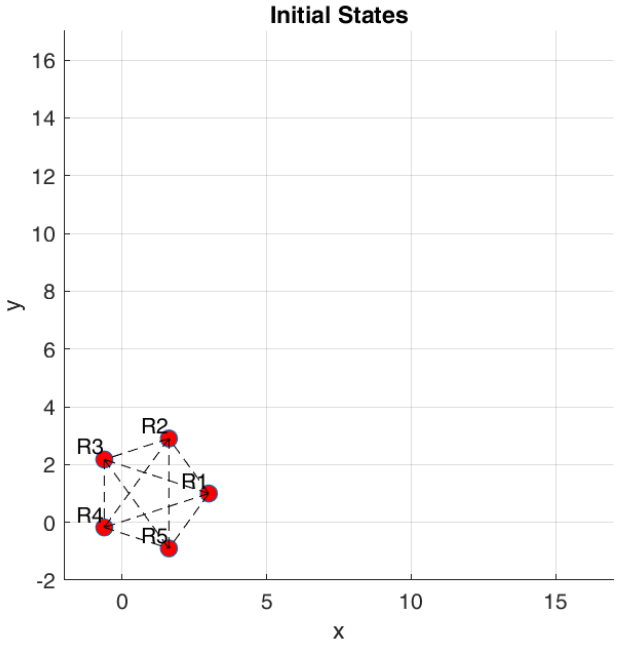

Figure 4.4: Initial Formation

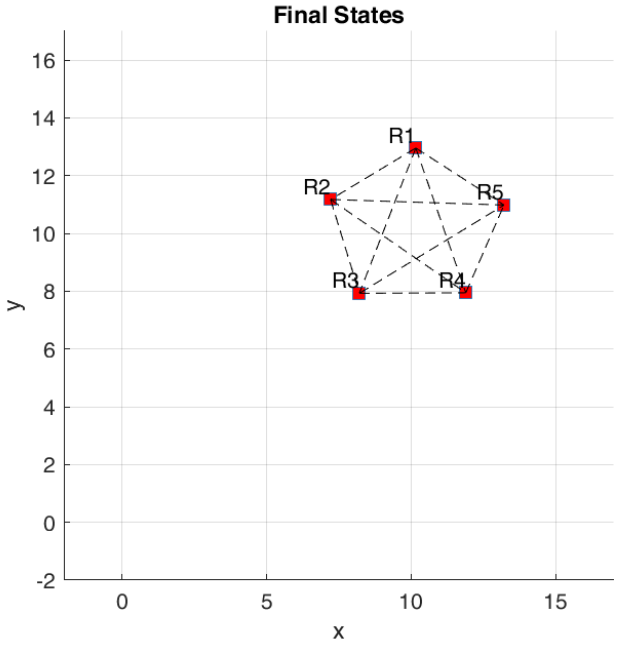

Figure 4.5: Final Formation

The multi-robot team moves from its initial coordinates to the target location, simultaneously scaling and rotating the formation. During this process, strict inter-agent separation is maintained, ensuring collision-free trajectories throughout the maneuver.

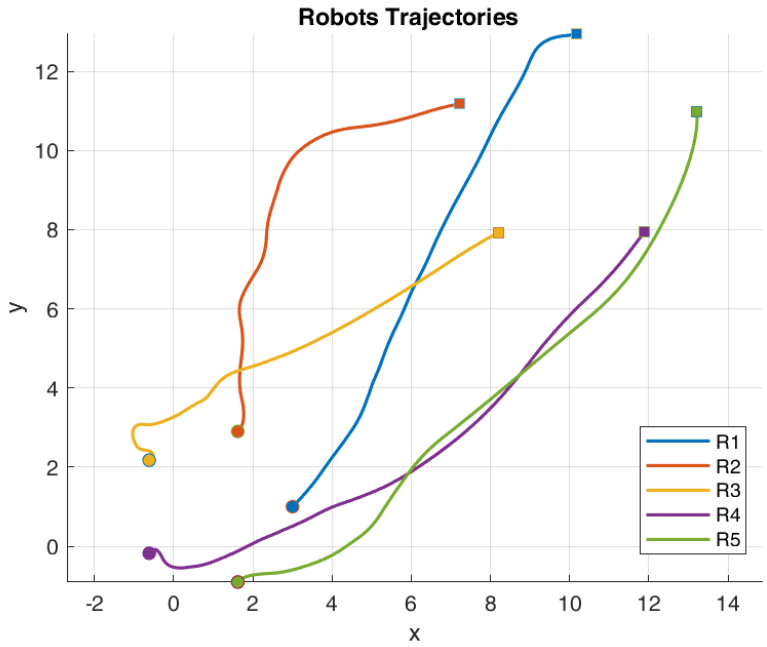

Figure 4.6: Robot Trajectories

4.4.3 Formation Reconfiguration via Shape Change

Let $\delta_{ij}^{Sh_1}$ and $\delta_{ij}^{Sh_2}$ represent the desired relative positions between robots i and j corresponding to the shapes Sh_1 and Sh_2, respectively. We introduce a continuous transition function $\beta(t) \in [0, 1]$, defined as follows:

- $\beta(0) = 0$ (initially, the formation is Sh_1),
- $\beta(t) = 1$ for $t \geq T_{trans}$ (After a specified transition time, the formation transitions to Sh_2).

The desired relative offset is given by:

$$\delta_{ij}^{d} = (1 - \beta(t))\, \delta_{ij}^{Sh_1} + \beta(t)\delta_{ij}^{Sh_2}. \tag{4.25}$$

Remark 4.4 *The following functions can be utilized as transition functions:*

- $\beta(t) = \min\left\{1, \dfrac{1}{T_{trans}}\right\}$,
- $\beta(t) = \begin{cases} 0 & t \leq t_{start}, \\ \dfrac{1}{1 + e^{-k(t-t_0)}} & t_{start} < t < t_{end}, \\ 1 & t \geq t_{end}. \end{cases}$

where $k > 0$ is a parameter, t_0 is the midpoint (indicating that the transition is halfway complete at $t = t_0$), and t_{start}, t_{end} represent the start and end times of the transition, respectively.

The distributed QP problem is solved locally by each robot, using information from its neighbors:

$$\min_{u_i} \frac{1}{2}\|u_i - u_{des}\|^2 + \lambda \sum_{j \in \mathcal{N}_i} \|p_i - p_j - \delta_{ij}^d\|^2 \tag{4.26}$$

This formulation enables each robot to utilize local information (through distributed consensus) to adjust its control input and drive the formation from shape Sh_1 to shape Sh_2 over time.

Example 4.3 *Consider 5 robots initially arranged in a circle with a radius of $r = 2$ and the formation center at $(1, 1)$. In this example, the formation undergoes a transition from a circular to a triangular shape, while simultaneously navigating toward the target.*

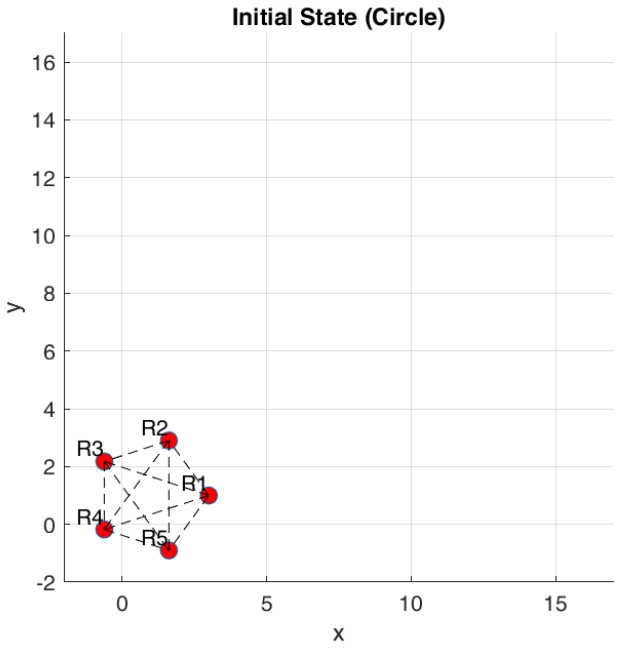

Figure 4.7: Initial Formation

As demonstrated in Figures 4.7-4.9,the five-robot team successfully performs a dual-objective task: translating toward the goal while continuously morphing from an initial circular lattice (radius $r = 2$, center at $(1, 1)$) to a final triangular formation, all without any inter-robot collisions. The smooth, non-intersecting trajectories validate that the CBF-QP supervisor enforces the prescribed minimum separations at all times, even while accommodating the coupled inertial dynamics and non-convex deformation path. These results highlight the controller's ability to manage complex, real-time formation reshaping while ensuring strict safety guarantees.

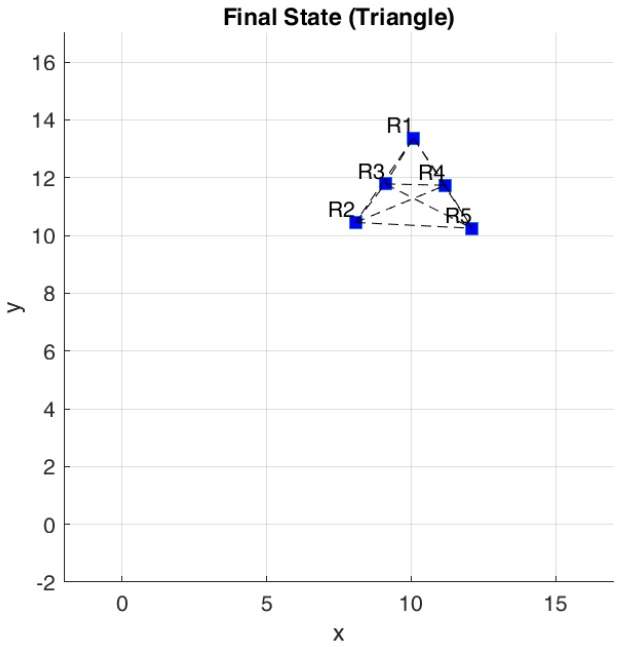

Figure 4.8: Final Formation

4.5 Conclusion

In this work, we have explored the problem of multi-robot formation control, focusing on both rigid and deformable formations. We presented methods for maintaining a desired formation, ensuring collision avoidance, and minimizing control effort, while also considering the dynamic environments that multi-robot systems operate in. Through various case studies, we demonstrated how the robots can navigate while preserving their formation and adapting to changes in the environment, such as scaling, rotating, and even reshaping the formation. A distributed Quadratic Programming (QP) framework was utilized to enable decentralized control, ensuring that each robot adjusts its behavior based on local information from its neighbors.

However, several challenges remain to be addressed in future work. One key difficulty is navigating in unknown or highly dynamic environments, where the formation may need to adapt continuously to avoid obstacles or handle unexpected changes in the surroundings. Additionally, the scalability of the proposed methods needs to be further explored, as the computational cost increases significantly as the number of robots grows. Future studies should also consider integrating more advanced machine learning techniques, such as neural networks, to facilitate real-time decision-making and improve the system's robustness to disturbances and uncertainties. These techniques could enable the robots to learn and adapt to new environments without requiring predefined models or strict supervision. Finally, further research into robust communication strategies is essential to ensure coordination and cohesion among robots in environments with unreliable or intermittent communication channels.

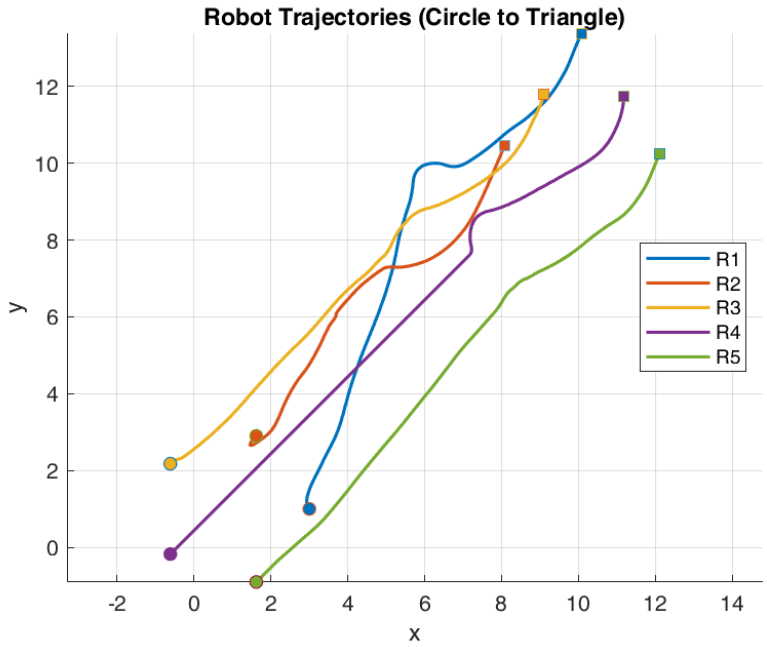

Figure 4.9: Robot Trajectories

By addressing these challenges, future advancements could lead to more efficient and resilient multi-robot systems capable of handling complex, real-world scenarios.

General Conclusion

In this thesis, we have explored the coordination of multi-robot systems with a focus on safe and adaptive formation control. By combining Higher-Order Control Barrier Functions (HOCBFs) with distributed optimization techniques, the proposed approach ensures that robots can maintain a desired formation while adapting to dynamic environments, avoiding collisions, and minimizing control effort. Through mathematical modeling, theoretical analysis, and a series of case studies, we demonstrated the ability of the proposed framework to handle rigid and deformable formations, as well as real-time formation reshaping. The results demonstrate the effectiveness of integrating safety-critical control with decentralized decision-making, enabling robots to autonomously adapt their behavior while maintaining global coordination.

The contributions of this work extend the state of the art in multi-robot coordination by providing a unified framework that guarantees safety even in the presence of formation deformation and dynamic obstacles. By utilizing distributed control and local information exchange, the approach is scalable and robust, suitable for large teams operating in complex environments. The case studies presented illustrate the practical applicability of the approach in real world scenarios, highlighting its potential for a variety of applications in fields such as autonomous vehicle fleets, swarm robotics, and multi-agent exploration.

Perspectives and Challenges

Despite the promising results, several challenges and open questions remain. One of the key difficulties lies in improving the scalability of the proposed methods, especially as the number of robots in a team increases. As the system size grows, the computational burden and communication overhead for maintaining coordination and safety constraints become more significant. To address this, further research into more efficient optimization algorithms, distributed consensus methods, and reduced-order

models is needed to enhance performance in large-scale systems.

Additionally, multi-robot systems often operate in dynamic and uncertain environments, where obstacles, communication failures, and environmental changes introduce new complexities. Future work should focus on integrating advanced machine learning techniques, such as reinforcement learning or neural networks, to enable robots to learn from experience and adapt to unknown environments without relying on pre-programmed models. Furthermore, ensuring robust performance in the face of network delays, intermittent communication, and sensor inaccuracies is crucial for practical deployment. By tackling these challenges, future research can enable more robust, scalable, and adaptive multi-robot systems capable of autonomously performing complex tasks in real world scenarios.

Another critical challenge is ensuring resilience against attacks or disruptions. Multi-robot systems are vulnerable to malicious interference, such as cyber attacks, spoofing, or targeted jamming of communication channels. Developing robust strategies for securing communication, detecting and mitigating attacks, and ensuring the continued functionality of the system despite adversarial behavior is essential. Future work could explore the integration of security measures, such as secure multi-party computation, cryptographic techniques, and fault tolerant protocols, to enhance the resilience of the system to both external and internal attacks. Ensuring that these systems can recover from attacks without significant performance degradation will be critical for applications in security-sensitive domains like military operations or critical infrastructure monitoring.

Bibliography

[1] F. Bullo, J. Cortés, and S. Martínez, *Distributed Control of Robotic Networks: A Mathematical Approach to Motion Coordination Algorithms*. Princeton University Press, 2009.

[2] W. Ren and R. W. Beard, *Distributed Consensus in Multi-vehicle Cooperative Control: Theory and Applications*, ser. Communications and Control Engineering. London: Springer, 2007.

[3] M. Mesbahi and M. Egerstedt, "Graph theoretic methods in multiagent networks," 2010.

[4] M. de Queiroz, X. Cai, and M. Feemster, *Formation Control of Multi-Agent Systems: A Graph Rigidity Approach*, ser. Wiley Series in Dynamics and Control of Electromechanical Systems. Wiley, 2019.

[5] K.-K. Oh, M.-C. Park, and H.-S. Ahn, "A survey of multi-agent formation control," *Automatica*, vol. 53, pp. 424–440, 2015.

[6] D. Panagou and V. Kumar, "Cooperative visibility maintenance for leader–follower formations in obstacle environments," *IEEE Transactions on Robotics*, vol. 30, no. 4, pp. 831–844, 2014.

[7] X. Zhang, Q. Yang, F. Xiao, H. Fang, and J. Chen, "Linear formation control of multi-agent systems," *Automatica*, vol. 171, p. 111935, 2025.

[8] A. Fagiolini, M. Pellinacci, G. Valenti, G. Dini, and A. Bicchi, "Consensus-based distributed intrusion detection for multi-robot systems," in *2008 IEEE International Conference on Robotics and Automation*, 2008, pp. 120–127.

[9] D. Koung, I. Fantoni, O. Kermorgant, and L. Belouaer, "Consensus-based formation control and obstacle avoidance for nonholonomic multi-robot system," in *2020 16th International Conference on Control, Automation, Robotics and Vision (ICARCV)*, 2020, pp. 92–97.

[10] K. Griparic, M. Polic, M. Krizmancic, and S. Bogdan, "Consensus-based distributed connectivity control in multi-agent systems," *IEEE Transactions on Network Science and Engineering*, vol. 9, no. 3, pp. 1264–1281, 2022.

[11] X. Xu, P. Tabuada, J. W. Grizzle, and A. D. Ames, "Robustness of control barrier functions for safety critical control," *IFAC-PapersOnLine*, vol. 48, no. 27, pp. 54–61, 2015.

[12] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs for safety critical systems," *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 3861–3876, Aug. 2017.

[13] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control barrier functions: Theory and applications," in *Proceedings of the 18th European Control Conference (ECC)*. Naples, Italy: IEEE, Jun. 2019, pp. 3420–3431.

[14] L. Wang, A. D. Ames, and M. Egerstedt, "Safety barrier certificates for collisions-free multirobot systems," *IEEE Transactions on Robotics*, vol. 33, no. 3, pp. 661–674, 2017.

[15] W. Xiao and C. Belta, "Control barrier functions for systems with high relative degree," in *2019 IEEE 58th Conference on Decision and Control (CDC)*, 2019, pp. 474–479.

[16] ——, "High-order control barrier functions," *IEEE Transactions on Automatic Control*, vol. 67, no. 7, pp. 3655–3662, 2022.

[17] X. Tan, W. S. Cortez, and D. V. Dimarogonas, "High-order barrier functions: Robustness, safety, and performance-critical control," *IEEE Transactions on Automatic Control*, vol. 67, no. 6, pp. 3021–3028, 2022.

[18] J. K. Verma and V. Ranga, "Multi-robot coordination analysis, taxonomy, challenges and future scope," *Journal of Intelligent and Robotic Systems*, vol. 102, no. 1, p. 10, 2021.

[19] A. Khamis, A. Hussein, and A. Elmogy, *Multi-robot Task Allocation: A Review of the State-of-the-Art*. Cham: Springer International Publishing, 2015, pp. 31–51.

[20] A. Madridano, A. Al-Kaff, D. Martin, and A. de la Escalera, "Trajectory planning for multi-robot systems: Methods and applications," *Expert Systems with Applications*, vol. 173, p. 114660, 2021.

[21] A. Erokhin, V. Erokhin, S. Sotnikov, A. Gogolevsky, Z. Prokopova, P. Silhavy, and R. Silhavy, "Optimal multi-robot path finding algorithm based on a," in *Intelligent Systems in Cybernetics and Automation Control Theory*. Springer International Publishing AG, 2018, vol. 860, pp. 172–182.

[22] S. Banik, S. C. Banik, and S. S. Mahmud, "Path planning approaches in multi-robot system: A review," *Engineering Reports*, vol. 7, no. 1, p. e13035, 2025.

[23] D.-H. Lee, "Resource-based task allocation for multi-robot systems," *Robotics and Autonomous Systems*, vol. 103, pp. 151–161, 2018.

[24] F. Li, Y. Ding, and K. Hao, "A dynamic leader-follower strategy for multi-robot systems," in *2015 IEEE International Conference on Systems, Man, and Cybernetics*, 2015, pp. 298–303.

[25] D. Gu and Z. Wang, "Leader-follower flocking: Algorithms and experiments," *IEEE Transactions on Control Systems Technology*, vol. 17, no. 5, pp. 1211–1219, 2009.

[26] M. Chen, H. Chen, D. Yang, L. Li, K. Liao, and W. Wei, "Formation obstacle avoidance control for multi-agent systems based on leader-follower strategy," in *2025 37th Chinese Control and Decision Conference (CCDC)*, 2025, pp. 627–632.

[27] S. Huang, H. Zeng, W. Lan, and X. Yu, "Leader-follower formation tracking control of mobile robots: A visual observer-based approach," *IEEE Transactions on Control Systems Technology*, vol. 33, no. 5, pp. 1937–1945, 2025.

[28] Y. Li and C. Tan, "A survey of the consensus for multi-agent systems," *Systems Science and Control Engineering*, vol. 7, no. 1, pp. 468–482, 2019.

[29] H. T. Do, H. T. Hua, M. T. Nguyen, C. V. Nguyen, H. T. Nguyen, H. T. Nguyen, and N. T. Nguyen, "Formation control algorithms for multiple-uavs: A comprehensive survey." *EAI Endorsed Trans. Ind. Networks Intell. Syst.*, vol. 8, no. 27, p. e3, 2021.

[30] A. Farooq, Z. Xiang, W. J. Chang, and M. S. Aslam, "Recent advancement in formation control of multi-agent systems: A review," *Computers, Materials and Continua*, vol. 83, no. 3, 2025.

[31] Z. Lin, L. Wang, Z. Han, and M. Fu, "Distributed formation control of multi-agent systems using complex laplacian," *IEEE Transactions on Automatic Control*, vol. 59, no. 7, pp. 1765–1777, 2014.

[32] P. Lin and Y. Jia, "Distributed rotating formation control of multi-agent systems," *Systems and Control Letters*, vol. 59, no. 10, pp. 587–595, 2010.

[33] A. Dorri, S. S. Kanhere, and R. Jurdak, "Multi-agent systems: A survey," *IEEE Access*, vol. 6, pp. 28 573–28 593, 2018.

[34] V. Singhal and D. Dahiya, "Distributed task allocation in dynamic multi-agent system," in *International Conference on Computing, Communication and Automation*, 2015, pp. 643–648.

[35] B. P. Gerkey and M. J. Matari, "A formal analysis and taxonomy of task allocation in multi-robot systems," *The International Journal of Robotics Research*, vol. 23, no. 9, pp. 939–954, 2004.

[36] C. Ju, J. Kim, J. Seol, and H. I. Son, "A review on multirobot systems in agriculture," *Computers and Electronics in Agriculture*, vol. 202, p. 107336, 2022.

[37] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: Modelling, Planning and Control*. Springer-Verlag London, 2009.

[38] J. Fax and R. Murray, "Information flow and cooperative control of vehicle formations," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1465–1476, 2004.

[39] W. Ren and R. W. Beard, "Consensus seeking in multiagent systems under dynamically changing interaction topologies," *IEEE Transactions on Automatic Control*, vol. 50, no. 5, pp. 655–661, May 2005.

[40] L. Asimow and B. Roth, "The rigidity of graphs," *Transactions of the American Mathematical Society*, vol. 245, pp. 279–289, 1978.

[41] L. Krick, M. E. Broucke, and B. A. Francis, "Stabilisation of infinitesimally rigid formations of multi-robot networks," *International Journal of Control*, vol. 82, no. 3, pp. 423–439, 2009.

[42] R. Olfati-Saber and R. Murray, "Consensus problems in networks of agents with switching topology and time-delays," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1520–1533, 2004.

[43] Y. Zheng, S. E. Li, K. Li, F. Borrelli, and J. K. Hedrick, "Distributed model predictive control for heterogeneous vehicle platoons under unidirectional topologies," *IEEE Transactions on Control Systems Technology*, vol. 25, no. 3, pp. 899–910, 2017.

[44] S. Prajna and A. Jadbabaie, "Safety verification of hybrid systems using barrier certificates," in *Hybrid Systems: Computation and Control (HSCC 2004)*, ser. Lecture Notes in Computer Science, R. Alur and G. J. Pappas, Eds., vol. 2993. Berlin, Heidelberg: Springer, 2004, pp. 477–492.

[45] M. Nagumo, "Über die lage der integralkurven gewöhnlicher differentialgleichungen," *Proceedings of the Physico-Mathematical Society of Japan. 3rd Series*, vol. 24, pp. 551–559, 1942.

[46] F. Blanchini, "Set invariance in control," *Automatica*, vol. 35, no. 11, pp. 1747–1767, 1999.

[47] J.-P. Aubin, *Viability Theory*. New York, NY, USA: Springer Science & Business Media, 2009.

[48] ——, "A survey of viability theory," *SIAM Journal on Control and Optimization*, vol. 28, no. 4, pp. 749–788, 1990.

[49] H. K. Khalil, *Nonlinear Systems*, 3rd ed. Prentice Hall, 2002.

[50] E. D. Sontag, "A lyapunov-like characterization of asymptotic controllability," *SIAM journal on control and optimization*, vol. 21, no. 3, pp. 462–471, 1983.

[51] A. D. Ames, K. Galloway, K. Sreenath, and J. W. Grizzle, "Rapidly exponentially stabilizing control lyapunov functions and hybrid zero dynamics," *IEEE Transactions on Automatic Control*, vol. 59, no. 4, pp. 876–891, 2014.

[52] R. Olfati-Saber, "Flocking for multi-agent dynamic systems: algorithms and theory," *IEEE Transactions on Automatic Control*, vol. 51, no. 3, pp. 401–420, 2006.

[53] A. D. Ames, X. Xu, J. W. Grizzle, and P. Tabuada, "Control barrier function based quadratic programs for safety critical systems," *IEEE Transactions on Automatic Control*, vol. 62, no. 8, pp. 3861–3876, 2017.

[54] S. Liu, D. Sun, and C. Zhu, "Motion planning of multirobot formation," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010, pp. 3848–3853.

[55] L. Ruiz-Fernandez, J. Ruiz-Leon, D. Gomez-Gutierrez *et al.*, "Decentralized multi-robot formation control in environments with non-convex and dynamic obstacles based on path planning algorithms," *Intelligent Service Robotics*, 2025.

[56] W. Guettaf, B. Bouderah, and M. H. Zaitri, "Distributed formation control of double-integrator systems via adaptive cbfs with dynamic safety margins," *Mathematical Modelling of Engineering Problems*, vol. 12, no. 7, pp. 2245–2252, 2025.

[57] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control barrier functions: Theory and applications," in *2019 18th European Control Conference (ECC)*, 2019, pp. 3420–3431.

[58] X. Xu, P. Tabuada, J. W. Grizzle, and A. D. Ames, "Robustness of control barrier functions for safety critical control," *IFAC-PapersOnLine*, vol. 48, no. 27, pp. 54–61, 2015.

ملخص:

تتناول هذه الأطروحة تنسيق نظام متعدد الربوتات مع التركيز بشكل خاص على التحكم في التشكيلات ودمج آليات السلامة في بيئة ديناميكية. الطريقة التي تدمج توابع الحواجز التحكمية CBF وتوابع الحواجز ذات رتب عليا HOCBF لضمان تجنب الإصطدامات والسماح بتشويه التشكيلات بشكل متحكم فيه. دراسة هذه الحالة أثبتت فعالية الإطار المقترح لضمان السلامة و التنسيق في السيناريوهات المعقدة. تفتح هذه الأطروحة آفاقا للتطبيقات في ماجالات الروبوتات المستقلة والمراقبة. ستضمن الأبحاث المستقبلية دمج التعليم الآلي و تحسين المرونة ضد الهجمات.

الكلمات المفتاحية: الأنظمة المتعددة الروبوتات، التحكم في التشكيلات، توابع الحواجز، السلامة، التشويه المتحكم فيه، التنسيق اللامركزي.

Abstract:

This thesis focuses on the coordination of multi-robot systems, particularly on formation control and the integration of safety mechanisms in dynamic environments. A methodology combining Control Barrier Functions (CBFs) and Higher-Order Control Barrier Functions (HOCBFs) is developed to ensure collision avoidance and allow controlled formation deformation. Case studies demonstrate the effectiveness of the proposed framework in maintaining safety and coordination in complex scenarios. This work opens up prospects for applications in areas such as autonomous robotics and surveillance. Future research will explore the integration of machine learning and improving resilience against attacks.

Key Words: Multi-robot systems, Formation control, CBF and HOCBF, Safety, Controlled deformation, Decentralized coordination.

Résumé:

Cette thèse porte sur la coordination des systèmes multi-robots, en particulier sur le contrôle de formations et l'intégration de mécanismes de sécurité dans des environnements dynamiques. Une méthodologie combinant des fonctions de barrière de contrôle (CBFs) et des fonctions de barrière d'ordre supérieur (HOCBFs) est développée pour garantir l'évitement des collisions et permettre des déformations contrôlées des formations. Des études de cas montrent l'efficacité du cadre proposé pour maintenir la sécurité et la coordination dans des scénarios complexes. Ce travail ouvre des perspectives pour des applications dans des domaines comme la robotique autonome et la surveillance. Des recherches futures incluront l'intégration de l'apprentissage automatique et l'amélioration de la résilience face aux attaques.

Mots Clés: Systèmes multi-robots, Contrôle de formations, CBF et HOCBF, Sécurité, Déformation contrôlée, Coordination décentralisée.