Towards Safe and Resilient Autonomy in Multi-Robot Systems

Wenhao Luo

CMU-RI-TR-21-26

July 2021

Thesis submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Robotics

The Robotics Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

Thesis Committee:

Katia Sycara, Chair
Maxim Likhachev
Changliu Liu
Amanda Prorok, *University of Cambridge*

Abstract

Autonomous systems are envisioned to increasingly co-exist with humans in our daily lives, from household service to large-scale warehouse logistics, agriculture environment sampling, and smart city. Among them, networked cooperative systems such as autonomous multi-robot systems have been widely studied given their capability of accomplishing complex tasks through cooperative behaviors. Reliable interactions among robots as networked safety-critical systems often require provably correct guarantees about safety (e.g. collision avoidance) and resilience (e.g. capability of maintaining communication and operating in an unknown environment). As we strive to design and control such a large-scale system, robots are often assumed to have perfect information (e.g. ground-truth state, system dynamics, and environment model information), unconstrained inter-robot communication, and fault-free operation. However, the precomputed guarantees based on these assumptions could be easily broken when deploying robots in the real world that is uncertain, rapidly changing, and inherently stochastic.

In this thesis, we seek to develop and validate mathematically grounded algorithms to assure safe and resilient interactions among robots that adapt to uncertain and possibly hostile dynamic environments. To achieve the design objective, we discuss three research topics, including (1) safe control and learning under uncertainty, (2) resilient multi-robot interaction through networking, and (3) data-driven multi-robot coordination adapting to the unknown environment.

For (1), we first propose a reactive safe control framework for multi-robot systems under *known* robotic system dynamics with localization and motion noise. The framework generates multi-robot motions through centralized or decentralized computation to formally satisfy the collision-avoidance with lower bounded probability guarantee, while respecting the original robot behaviors for task efficiency. When the robotic system dynamics is *unknown* and *partially* modelled, we then develop a *sample efficient* safe learning framework for control that allows the robot to locally learn the unknown dynamics online while achieving sample efficiency in optimizing task performance with bounded regret and safety guarantee.

For (2), we design *provably correct* connectivity control frameworks utilizing the graph theoretic and control theoretic approaches for a team of robots to satisfy various global and local interaction network requirements while progressing towards mission goals. This allows the robot team to maintain, recover, or enhance user-defined network connectivity for smooth information exchange under possible adversaries and minimally deviate from their original behaviors. The proposed frameworks prove to be *minimally restrictive* towards maintaining the robots' original task-prescribed controllers subject to the connectivity requirements, thus optimally balancing between the required network redundancy and task performance.

With the assurance of safety and resilience in terms of retaining integrity of multi-robot systems through reliable networking, for (3) we address the problem of data-driven multi-robot coordination in the application of sensor coverage to achieve resilient cooperative behaviors in unknown environments. Specifically, a learning-enabled multi-robot control framework is proposed for robots to explore in the unknown environment and simultaneously optimize the task performance regarding sensor coverage using environment model learned online. This allows the robots to share locally collected data or model-related parameters through connected multi-robot network to learn a global environment model and develop task-related coverage controllers with this model to optimize the coverage performance.

Our approaches of safe control and resilient networking for multi-robot systems share a unified optimization-based control framework in real time, thus enabling the synthesis of certified control modules for different task-prescribed multi-robot behaviors with safety and connectivity guarantees. With our approach of sample efficient safe learning for control, we further extend the model-based safety analysis to partially modelled dynamical systems with learning-based behaviors, which enables strong synergies between learning and control with safety guarantee. The integration of data-driven methods into cooperative multi-robot control facilitates the design of adaptive multi-robot coordination behaviors for improved performance in unknown environments. The effectiveness of the proposed methods is demonstrated and evaluated through a set of simulations and realistic simulated robotic platforms.

To my family

Acknowledgements

First and foremost, I would like to express my sincere gratitude to my advisor Katia Sycara, who has been incredibly supportive throughout my journey at Carnegie Mellon University. Her profound knowledge, insightful visions, and patience not only helped me survive through struggling phases of PhD study and career development, but also greatly influenced me on how to keep passionate and be a deep thinker. It was always amazing to see how great ideas came from simple intuition during our numerous inspiring discussions. I feel so lucky to have the freedom to explore my own research ideas and stay happy when pursuing my goals. I sincerely wish I could be a great researcher and mentor as she is in the future.

I am also grateful to my thesis committee members: Maxim Likhachev, Changliu Liu, and Amanda Prorok, who have provided extremely helpful discussion and feedback. Their vision, guidance, and knowledge have been invaluable in my work and this thesis. In addition, I am thankful to Maxim Likhachev for his strongest support throughout my Master and PhD study in supervising my research progress and providing tremendous help with great advice, without which I would not have been where I am today.

I am fortunate to have spent two wonderful summers with Ashish Kapoor as a research intern at Microsoft. I learned a lot from the fruitful discussions with him that shaped my research in robot safety as one of the main parts of this thesis work. His insight, encouragement, and unique viewpoint on research and life have always been an inspiration for me. I am also thankful to my collaborators Nilanjan Chakraborty and Wen Sun, for always being available for many fruitful discussions that lead me to new research paths. I am also thankful to other faculty members at CMU and Pitt, especially Mike Lewis, George Kantor, John Dolan, Sebastian Scherer, Michael Erdmann, and Hartmut Geyer whom I am fortunate to collaborate with and get new perspectives.

My life at CMU could never be enjoyable without the support from my friends and labmates. They are the unsung heroes who have been with me through countless days and nights fighting against deadlines: Sasanka Nagavalli, Shehzaman Salim Khatib, Yulong Pei, Yuqing Tang, Anqi Li, Navyata Sanghvi, Ramitha Sundar, Sha Yi, Rui Liu, Dana Hughes, Changjoo Nam, Jaskaran Grover, Mengtian Li, Peiyun Hu, Chiyu Dong, Tianyu Gu, Rogerio Bonatti, Zhiqian Qiao, Weikun Zhen, Chen Fu, Jayanth Krishna Mogali, Chendi Lin, Yifan Ding, Fan Jia, Denise Li, Akshat Agarwal, Swaminathan Gurumurthy, Vigneshram Krishnamoorthy, Sumit Kumar. I would also like to thank Suzanne Muth, Keyla Cook, and Barbara Jean Fecich for making things easier for me. It has been a great pleasure to be part of this amazing place.

Going through the academic job search this past year is particularly challenging and I wouldn't be able to make it without the generous and strongest support from my mentors, collaborators, and friends. Many thanks to Katia Sycara, Mike Lewis, Ashish Kapoor, Maxim Likhachev, Wen Sun, Changliu Liu, Nilanjan Chakraborty, and Aaron Steinfeld for their tremendous help, giving me suggestions, and sharing their own experience with insightful thoughts. I would also like to thank Rui Liu for answering countless questions from me and for the incredible support the entire time.

Last but not least, I would express a deep sense of gratitude to my beloved parents and grandparents for their endless support and love throughout my life that made me who I am today. I am forever indebted to them. I would like to thank my girlfriend, Yiwei Lyu, for her unconditional love and for numerous discussions and encouragement that walk me through the most stressful time during my final stage at CMU, which is also the best part of my journey.

Contents

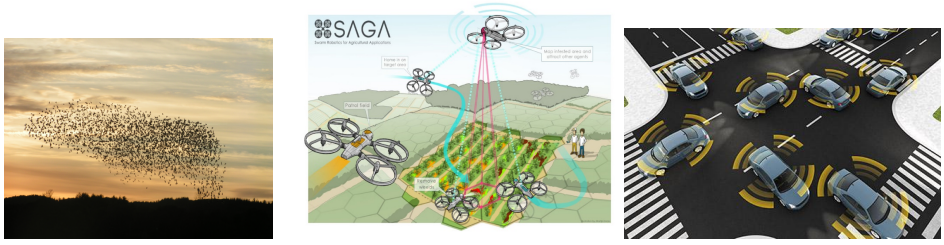
List of Figures

Chapter 1

Introduction

1.1 Multi-Robot Systems: Motivation and Challenges

Multi-robot systems have been widely studied for extending its capability of accomplishing complex tasks. Such systems are often comprised of a number of agents that collaborate autonomously in diverse applications through cooperative behaviors, e.g. environmental sampling [1, 2, 3], area coverage [4, 5, 6], search and rescue [7]. In those tasks, the robots are often assumed to be able to communicate and coordinate their individual actions to invoke collective behaviors as the overall missions evolve. To ensure successful mission execution, the robot team is expected to operate reliably from both individual and system perspectives: First, each robot as a safety-critical system should remain safe during movements. Second, the robot team functions as a system and requires integrity through robust networking to connect distributed robot components for information propagation and collective task executions. Third, in order to reliably operate in unknown environment, the robot team need to adapt their coordination behaviors as more data are observed during interaction with the environment.



(a) (b) (c)

Figure 1.1: Examples of multi-robot systems.

To achieve these capabilities, it is critical to develop algorithms along with mission-oriented design for the multi-robot system to satisfy these fundamental requirements while respecting the original tasks. For robot safety, existence of possible uncertainty and adversaries in the environment should be addressed when designing the safe robot controllers. For example, consider the problem of building an automatic collision avoidance system (ACAS) for aerial robots that would scale up as the autonomous aerial traffic increases. Such a system will override the robots original controllers to avoid potential collisions. While the pre-computed safety guarantees by design often relies on accurate perception and model information to derive reasonable actions that are safe in the predictable future, it could be easily broken when deploying robots to the physical world, e.g. self-driving car crash due to inaccurately localized obstacles. Thus it demands for more rigorous safe behavior designs that are robust to various real-world factors including uncertainty, non-determinism and approximations made in the formulation of the system. On the other hand, safety under *unmodelled* uncertainty remains challenging given the unstructured unknown components that are difficult to reason. Learning-enabled decision making process [8, 9, 10, 11, 12] has been widely used to characterize the uncertainty in the partially modelled system dynamics and unknown environments with safety analysis. This introduces new challenges about how to properly trade off between robust safety assurance (conservativeness) and optimal task performance (aggressiveness) during learning that could often inherently contradict each other.

Resilience in terms of 1) preserving the integrity of the multi-robot system itself through communication maintenance and 2) adapting robot behaviors to the unknown environment is another challenge when deploying robots to perform a set of tasks over long periods of time in unknown environment. As coordinated multi-robot behaviors are often achieved through proximity-based multi-robot information-exchange networks with limited communication range, the robots need to keep connected by staying close to each other while spreading out to perform their original tasks. Moreover, robots in adversarial environment will suffer from the risk of losing robot members due to attacks or increasing robot failures over time that pose additional difficulty for them to stay connected as a cohesive group. How to effectively retain the ability for the robots to recover network connectivity from faults while efficiently maintaining their original task-prescribed behaviors remains challenging. Besides, in presence of unknown environment it is necessary to integrate learning component to the standard model-based multi-robot behavior design such that robots could dynamically adjust their plans based on the data observed during interaction with the environment for improved task performance.

Hence, these fundamental challenges motivate the objective of research work

in this thesis, which is to **develop and validate *formally provable* multi-robot frameworks under uncertainty and adversaries to assure safe and resilient networked interactions among robots to accomplish mission goals**. In particular, we strive to address the following challenging research questions.

(1) How to ensure *provable* safety in terms of collision avoidance for multi-robot system with modelled dynamics under uncertainty, e.g. with localization and motion noises?

(2) How to ensure *provable* safety, sample efficiency and task performance when the dynamics of a robotic system is only partially modelled and needs to interact with the possibly risky environment to learn to perform a task optimally?

(3) How to *optimally* constrain robot motions during mission operation to ensure connectivity for successful networking within a moving multi-robot systems while respecting the original multi-robot behavior?

(4) In presence of possible robot failures, how to design a *resilient* framework for the robots to recover and increase network connectivity by *optimal* reconfiguration in a minimally disruptive manner to their primary tasks?

(5) How to develop adaptive model-based multi-robot coordination behaviors that allow robots to learn from interaction with an unknown environment and simultaneously optimize its primary task performance?

1.2 Thesis Overview

In light of the mentioned research questions, we have developed a set of methodologies in this thesis work contributing towards a safe and resilient multi-robot framework with literature review provided in Chapter 2.

As depicted in Fig. 1.2, the remainder of the thesis is organized as follows: Chapter 3-4 provide discussions of the proposed safe control and learning for autonomous robotic systems. Chapter 3 discusses centralized and decentralized safe behavior design for multi-robot systems. Chapter 4 provides a safe learning framework for a single robot nonlinear control task with partially modelled dynamics and shows how such systems could operate safely and optimally despite the model uncertainty. Chapter 5-6 propose connectivity control algorithms that enable robots to stay connected as one integrated group for smooth coordination. Chapter 5 develops an optimal method to preserve different levels of connectivity among robot members in a group, and Chapter 6 extends that to a resilient setting where the robot

team stay connected in presence of adversaries to maintain or enhance the desired multi-robot network connectivity. With the fulfilled assumptions of safety and connectivity, Chapter 7 discusses the learning-enabled multi-robot sensor coverage as an application of data-driven multi-robot coordination where cooperative multi-robot behaviors adapt to an initially unknown environment as mission evolves.

Contents and notations in each chapter are intended to be self-contained. The relationship among different chapters are shown in Fig. 1.2. Some of the discussed work have been published in [3, 6, 13, 14, 15, 16, 17, 18]. Next, we will give an overview of each chapter to briefly introduce the main ideas behind the work.

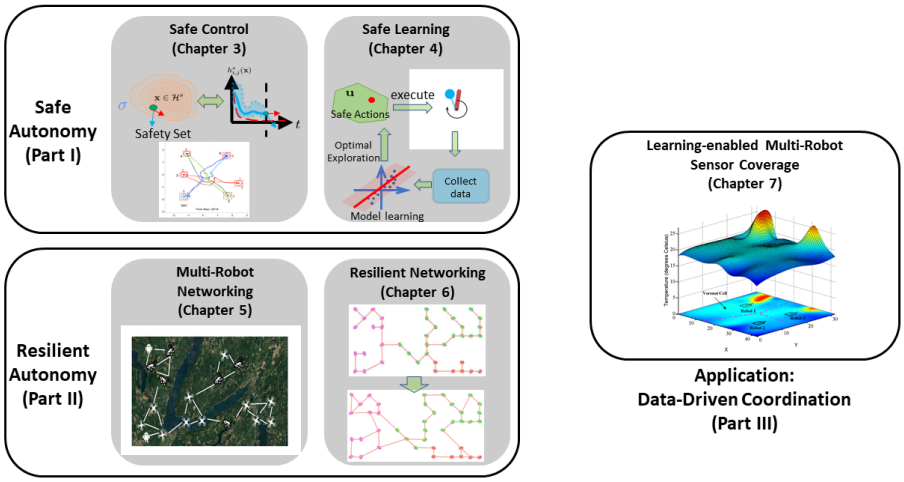


Figure 1.2: Thesis outline.

1.2.1 Multi-Robot Safe Control under Uncertainty

The focus of this part of the thesis work in Chapter 3 is on a safe control algorithm for homogeneous multi-robot system under uncertainties, addressing the Research Question 1. In particular, we are dealing with collision avoidance that accounts for both measurement and motion uncertainty over the multi-robot systems. While many prior approaches have attempted to address these different aspects of the problem, a complete solution addressing all the above aspects has been elusive. Many methods that attempt to address the measurement uncertainty often make restrictive assumptions, such as Gaussian representation of the uncertainties [19, 20, 21, 22, 23]. Approaches that consider bounded localization or control disturbance using conservative bounding volumes [24, 25, 26, 27] often overestimate the probability of collisions.

Our work proposes a novel approach that provides chance-constrained collision-free guarantees for crowded multi-robot team operating in a realistic environment. Akin to real-world we consider scenarios with both the measurement uncertainty as well as incomplete information about the dynamics. At the heart of the method is the idea of probabilistic safety barrier certificates (PrSBC) that enforces the chance constrained collision avoidance with deterministic constraints over controllers. With PrSBC constraints, the safety controller can be achieved by minimally modifying the existing controllers in real-time as done by other control barrier function approaches [28, 29]. This hence formally satisfy the collision-avoidance chance-constraints while staying as close to the original robot behaviors. Our goal here is to provide a real-time safety envelope around any existing controller that accounts for uncertainties and non-determinism in a probabilistic settings. The key assumption here is the finite support of the uncertainties arising due to sensor measurements, incomplete dynamics and other exogenous variables.

1.2.2 Sample-efficient Safe Learning for Control

In Chapter 4, we consider the safe reinforcement learning (safe RL) problem for a partially modelled dynamical system in nonlinear control tasks. Traditional model-based safe design [28, 29, 30, 31] are not applicable here due to the imperfect unmodelled information of the system dynamics. Safe RL approaches [32, 33, 34, 35] with constraints satisfaction have been proposed to encode safety consideration in a modified optimality criterion or in the constrained policy exploration process with external knowledge, e.g. an accurate probabilistic system model [35, 36]. However, the effectiveness in preventing risky behaviors relies on the sufficient period of policy learning where the unsafe situations could happen in the early learning stage. The key idea is to develop a learning method that encourages the robot to take actions that safely and efficiently explore the unknown while seeking to optimize the primary control task with the learned dynamics over time.

In this part of work, we propose a *provably correct* method that handles both sample efficient safe learning and online nonlinear control task in partially unknown system dynamics. In particular, we develop an Optimism-based Safe Learning for Control framework that integrates 1) stochastic discrete-time control barrier functions (CBF) to ensure forward invariant safety under uncertainty, and 2) an optimism-based exploration strategy that enjoys a formally provable regret bound. To leverage between safe exploration and exploitation, the framework utilizes an optimism-based exploration strategy in face of uncertainty [37] to encourage efficient dynamics exploration and simultaneously synthesizes with model-based nonlinear control algorithms to safely optimize the policy performance under the learned dynamics. Such framework is proved to be safe and near optimal with

bounded regret, quantifying the sample efficiency and control performance. Compared to other existing works on safe learning or safe RL [8, 11, 12, 38, 39], our framework is able to simultaneously guarantee safety (remain in safe set during learning and execution), performance (task rewards maximization), and sample efficiency (near-optimal regret bound).

1.2.3 Multi-Robot Networking with Global and Subgroup Connectivity Maintenance

Chapter 5 addresses the problem of multi-robot networking in the context of connectivity control for a robotic team consisting of multiple task-oriented subgroups. The key idea is to develop a optimization-based multi-robot control framework that 1) select the optimal connectivity topology to constrain the robot motions that satisfy various connectivity requirements for successful networking while providing the greatest flexibility for the original mission operation, and 2) minimally modify the robot original task-related controllers subject to the invoked connectivity constraints so as to preserve the mission evolution at best. In particular, we consider the global and subgroup connectivity maintenance in this work for multi-robot behavior mixing, where robots *simultaneously* performing multiple behaviors in different subgroups while remaining connected. To ensure efficient collaboration and coordination, it is necessary for the multi-robot network to ensure connectivity within each subgroup and across subgroups as well as global connectivity.

We develop a generalized behavior mixing framework with minimum global and subgroup connectivity maintenance. Such framework is based on a bilevel optimization process that 1) incorporates a novel distributed *Minimum Connectivity Constraint Spanning Tree (MCCST)* to compute real-time *minimum connectivity constraints*, and 2) minimizes the revision to the original controllers subject to our invoked connectivity constraints and collision avoidance constraints formulated by the barrier certificates with control barrier functions (CBF) [28, 40]. In particular, MCCST computes the provably optimal set of communication links for the robots to maintain, which (a) has minimum number of links, and (b) invokes the connectivity constraints for global and subgroup connectivity least likely to be violated by the original controllers. Minimum connectivity maintenance is thus achieved by minimally modifying the original controllers to preserve these dynamic *least constraining* communication links and avoid collisions. This framework provides a way to characterize feasible control space for the robot team with guaranteed networking capabilities, and hence will be overarching with other safety and mission-oriented constraints so that the robots are able to interact with other robots at all times.

1.2.4 Resilient Multi-Robot Connectivity Maintenance

As the scale of the multi-robot team grows, the increasing number of unexpected robot failures necessities the recent research on robust multi-robot networking [16, 41, 42, 43] that seeks to ensure resilient inter-robot interaction in presence of adversaries. This is due to that some robot failures on the key robot nodes within multi-robot network could easily disconnect the whole system and hence prevent the robots from executing information exchange and collective behaviors. Moreover, considering the adversarial scenarios where the robot team is under continuous attacks and keeps losing members, it is critical to have a resilient algorithm that improves the connectivity among robots so that they can stay reliably connected at all times.

In Chapter 6, we have developed algorithms to maintain and enhance the connectivity of a multi-robot system with minimal disruption to the primary tasks that the robots are performing. In contrast to many existing work that can only maintain the current connectivity of the multi-robot graph, we propose a generalized connectivity control framework that allows for reconfiguration of the multi-robot system to provably satisfy any connectivity demand, while minimally disrupting the execution of their original tasks. In particular, we propose a novel $k-$Connected Minimum Resilient Graph (k-CMRG) algorithm to compute an optimal $k-$connectivity graph that minimally constrains the robots original task-related motion. The original controllers are then minimally modified to drive the robots and form the k-CMRG. This is an extension to the connectivity framework in Section 1.2.3 with resilience property adjustable based on human operator's input. Thus far, we completed the basic framework for multi-robot systems allowing for provably safety and resilient networking capabilities.

1.2.5 Learning-enabled Multi-Robot Sensor Coverage

Multi-robot sensor coverage is a classic distributed multi-robot coordination problem where a group of robots are deployed in an environment to coordinate their motions such that the sensing performance by the robot team over the environmental phenomenon from their final positions is maximized. Successful coordination strategy relies on the knowledge of the distribution of environmental phenomenon [44]. However, when such information is unknown beforehand, the robots need to explore in the environment and take observations to learn this model, which could be time-consuming. Hence, assuming assured multi-robot safety and network connectivity from our previous framework, the idea of this part of the work in Chapter 7 is to develop efficient multi-robot task-prescribed controllers that enable robots to take as few samples for efficient environmental modeling while ap-

proaching to their estimated optimal sensing locations in a coordinated manner.

We propose adaptive coverage control strategies with online data-driven environmental modeling for robots to take samples for leaning while approaching to the estimated optimal locations. To model significantly different components of the environmental phenomenon that is sensed locally by individual robots, we employ non-parametric inference methods such as Mixture of Gaussian Processes that provide good estimation of the environmental phenomenon distribution over the field from data collected at selected locations by the robots. Depending on the communication constraints, we develop both centralized and decentralized design for the learning-enabled multi-robot coverage controllers that integrate an information theoretic criterion to balance between model learning and coverage optimization, allowing robots to predict and move to the optimal sensing locations with improved performance. With the decentralized design, we propose a *distributed mixture of Gaussian Processes* algorithm that enables robots to collaboratively learn the global distribution of interests by exchanging only model-related parameters whose size is independent from the number of collected samples, and hence avoiding the transmission of all local data from every robot for communication efficiency. We empirically demonstrate the effectiveness of our algorithm via evaluation on real-world data gathered from agricultural field robot and indoor static sensors.

1.3 Contributions

This thesis provides the following contributions:

- **Multi-Robot Collision Avoidance with Probabilistic Safety Guarantee under Localization and Motion Uncertainty:** Chapter 3 proposes a Probabilistic Safety Barrier Certificates (PrSBC) method to define the space of admissible control actions for the robot team that are probabilistically safe with theoretical guarantee under uncertainty on localization and motion. The key advantage of the approach is that no assumptions about the form of uncertainty are required other than finite support, also enabling worst-case guarantees. We also derive a formal proof of existence of PrSBC in a closed form rendering feasible safety controller with probabilistic safety guarantee.

- **Sample-efficient Safe Learning for Control:** Chapter 4 proposes 1) a provably sample efficient episodic online learning framework that integrates safe model-based nonlinear control approaches with optimism-based exploration strategy to simultaneously achieve safe learning and policy optimization for online nonlinear control tasks, and 2) provides rigorous theoretical analy-

sis of guaranteed safety under learned uncertainty and near-optimal online learning and policy performance with proved regret bound.

- **Multi-Robot Networking with Global and Subgroup Connectivity Maintenance:** Chapter 5 proposes a generalized bilevel optimization based multi-robot networking framework to enable *simultaneous execution of different behaviors and sequences of behaviors within a single robot team,* while ensuring global and subgroup connectivity and collision avoidance. A real-time distributed Minimum Connectivity Constraint Spanning Tree (MCCST) algorithm is developed to select the minimum inter-robot connectivity constraints preserving subgroup and global connectivity that are *least likely to be violated* by the original controllers, hence providing greatest flexibility for original multi-robot task operations. The algorithm is *computationally efficient* and *scalable* to large number of robots.

- **Resilient Multi-Robot Connectivity Maintenance:** Chapter 6 proposes an algorithm to maintain and enhance the connectivity of a multi-robot system with minimal disruption to the primary tasks that the robots are performing. This algorithm is useful in a supervisory control setting when an operator wants to enhance the connectivity of the robot team to any desired value. A novel $k-$CMRG method is also developed to compute the optimal weighted $k-$node connected resilient graph for arbitrary initially connected multi-robot graph, imposing *least* connectivity constraints to the robots. We derive theoretical analysis and proof of the optimality of our algorithm with guaranteed, user-specified network resilient connectivity in presence of continuous robot failures.

- **Learning-enabled Multi-Robot Sensor Coverage:** Chapter 7 proposes an adaptive coverage controller that couples the adaptive sampling with information-theoretic criterion for efficient distributed model learning and coverage optimization with a reduced number of samples. This allows the robots in bandwidth-constrained environment to share model-related parameters learned from their own data to maintain important characteristics of local data distribution and eventually converge to consensus of global model parameters describing the learned environment in a uniform manner. It provides a principle information sharing strategy that can be easily embedded to other distributed multi-agent learning applications.

Chapter 2

Related Work

2.1 Multi-Robot Safe Control under Uncertainty

Safe control in terms of collision avoidance for robots operating in dynamic environments have been studied for safety consideration over the years. To avoid static and/or moving obstacles with perception uncertainty or motion disturbance, safe control and planning approaches such as [19, 45, 46] have been proposed to generate a sequence of tracking controllers over finite time horizon to achieve guaranteed probabilistic collision avoidance at run-time. In multi-robot applications, however, this could be computationally intractable due to the large scale of the multi-robot system. To address multi-robot collision avoidance, reactive methods such as reciprocal velocity obstacles (RVO) [47, 48, 49, 50], safety barrier certificates (SBC) [10, 29, 51], and buffered Voronoi cells [52] are presented to compute on-line multi-robot collision-free motions in a distributed manner. While all of them scale very well in large scale multi-robot team, they require perfect state information and/or accurate dynamics of the robots and the moving obstacles. In many practical cases, highly accurate state information and motion model may not be accessible to the robots.

To account for uncertainties associated to the robot state information and motion model, the mentioned collision avoidance methods have been extended to probabilistic representations. For example, to handle the bounded localization uncertainty, the concept of velocity obstacles is adopted to develop enlarged conservative bounding volumes around the robot [24, 25, 26, 27]. As mentioned in [20], this could often overestimate the probability of collisions. In other works [20, 21, 53], chance constraints are often employed to explicitly consider the collision probability. Similar ideas of probabilistic buffered Voronoi cells are utilized in [21, 53] to modify the buffered Voronoi boundary [52] based on the measurement uncer-

tainty of the other robots, so that the robots will never come across the designated Voronoi cell and so to avoid collisions. In those works, the chance constraints explicitly depend on the position level uncertainty of the robots. Probabilistic representation of reciprocal velocity obstacles [49, 50] are introduced in [22, 54] to develop first order constraint accounting for both measurement and actuation uncertainties. Key to the success of most of these chance constrained methods is the common assumption of Gaussian representation of uncertainties. It remains challenging when prior knowledge of the uncertainty model is not available or it is not necessarily Gaussian, e.g. readings from an on-board GPS sensor that only have an expected value with an finite support as accuracy.

Another family of reactive collision avoidance approaches is the recent optimization-based safety control using control barrier function [10, 28, 29, 31, 55, 56]. The safety controller is able to minimally revise the nominal controller in the context of quadratic programming and ensures the robots remain in the safety set at all time, leading to a minimally invasive safe control behavior. In [29], the control barrier function is employed to develop the Safety Barrier Certificates (SBC) for multi-robot systems, depicting a non-conservative safety envelope for the multi-robot controller from which the robots stay collision-free at all time. Extensions to higher order nonlinear system dynamics using SBC and Exponential Control Barrier Function (ECBF) have been introduced in [10, 31, 38]. In [38], the online safe learning and SBC-based collision avoidance is achieved by utilizing Gaussian Process to learn the motion disturbance while assuming perfect localization information. In this thesis, we propose the probabilistic safety barrier certificates (PrSBC), which extends the deterministic SBC [29] to a probabilistic setting to account for both localization and motion uncertainties of the ego robot and other robots/obstacles. No assumptions about the uncertainty model are required other than finite support. We show the PrSBC could handle other uncertainty models as well.

2.2 Safe Learning for Control

The control of safety-critical system such robotic systems is a difficult challenge under uncertainty and lack of complete information in the real world applications. While Reinforcement Learning (RL) algorithms that seek for long-term reward maximization has achieved significant results in many continuous control tasks [57, 58], it has not yet been widely applied to safety-critical control tasks as the rigorous safety requirements may be easily violated by intermediate policies during policy learning. Safe RL approaches [32, 33, 34, 35] with constraints satisfaction have been proposed to encode safety consideration in a modified optimality

criterion or in the constrained policy exploration process with external knowledge, e.g. an accurate probabilistic system model [35, 36]. However, the effectiveness in preventing risky behaviors relies on the sufficient period of policy learning where the unsafe situations could happen in the early learning stage.

Model-based approaches utilizing Model Predictive Control (MPC) or Lyapunov-based methods have seen a number of success in demanding control tasks under different constraints with accurate system models [20, 30, 59] or approximated dynamics [60]. For provable long-term safety guarantee, safety in terms of *set forward invariance* has become an active research area using Lyapunov functions [8] and control barrier functions (CBFs) with perfect system model [28, 30, 55, 56] or noisy model with known distributions [13, 61, 62, 63]. However, these control-based approaches still require known system model uncertainty and could be overly conservative for system behaviors in presence of large uncertainty. Consider an autonomous mobile robot operating in an unknown environment, it is desired to have a strategic exploration strategy that enables the robots to safely collect data for modelling the uncertainty and improve the control performance over time.

For this purpose, very recently integrating data-driven learning-based approach with model-based safe control approaches has received significant attention to achieve model uncertainty reduction while ensuring provable safety [8, 9, 11, 12, 38, 39, 64, 65]. The process often involves safe policy exploration with data collection from a nominal dynamics model and iteratively reduce learned model uncertainty over time to expand certified safety region of the system's state space [8, 11, 38, 39, 64]. However, such exhaustive data collection for safe learning could suffer from poor scalability and low efficiency for primary task. For example, instead of densely sampling over the space, it may be more beneficial to guide the safe exploration process towards task-prescribed policy optimization. Recent work [12] incorporates the safe learning using Gaussian Process (GP) and CBF into a model-free RL framework (RL-CBF) so that the guided exploration process will not only learn model uncertainty impacting safe behaviors but also optimizing the policy performance. Nevertheless, there is no theoretical guarantee on the learning performance in terms of sample efficiency or the control performance for the primary task. In this thesis, we propose a *provably correct* method that handles both sample efficient safe learning and online nonlinear control task in partially unknown system dynamics.

2.3 Multi-Robot Connectivity Maintenance

The general problem of connectivity maintenance has been widely studied in the past decade due to its importance in enabling local information sharing and collaboration for multi-robot systems in performing complex tasks. Given an initially connected multi-robot spatial communication graph, the goal of continual connectivity control is to couple the task-related controllers of robots with a connectivity controller such that the communication graph over time remains connected. There have been two major classes of connectivity control methods: 1) *local methods* that seek to preserve the initial connectivity graph topology over time [66, 67, 68], and 2) *global methods* that aim to preserve the global algebraic connectivity of the communication graph by deriving controllers to keep the second smallest eigenvalue of the graph Lapacian positive at all times [69, 70, 71, 72]. While the global connectivity control provides better flexibility over local methods as it allows for changing network topology, neither of them is able to deal with flexible global connectivity and subgroup connectivity maintenance *at the same time*. Moreover, for both of the methodologies there is no guarantee that the perturbation from the connectivity controllers is minimum over the original robot controllers.

To achieve more flexible connectivity control with multiple behaviors, i.e. simultaneously exploring different regions, recent work [73, 74, 75] have explored the idea of redeploying a certain number of robots to act as communication relays, while aiming to allow the rest of the robots to perform their original tasks. In particular, the communication relays can be derived by following certain structured behaviors such as lattice-based formations [74, 76], or by separate optimization process that explicitly assigns some of the robots as connectors [73, 75]. In order to find a more flexible communication relay structure with quantified pairwise connectivity, [77] proposed to employ minimum spanning tree topology and uses pairwise distance as heuristic to provide better freedom of robot motion, i.e. robots closer to each other are less restrictive. However, these heuristic methods have no theoretical guarantee that the selected connectivity constraints are minimum to the original task-related robot controllers.

For minimally invasive multi-robot control revision with constraint satisfaction, control barrier functions [28] have been employed to encode a variety of inter-robot constraints and the resulting constrained control outputs lead to forward invariance of the satisfying set, i.e. robots remain collision free and connected. However, these capabilities are achieved by predefining the connectivity constraints so as to preserve fixed predefined communication topology [40, 78]. This could weaken the minimal invasiveness to robots controllers and limit the motion of the robots during execution. On the contrary, we do not define communication topology to preserve beforehand, but instead calculate the optimal topology

online during execution. In our work, we are optimizing both the real-time connectivity constraints to enforce and the modifications to the original controllers to achieve minimum connectivity maintenance, thus rendering the least perturbation to the original multi-robot behaviors.

2.4 Multi-Robot Robust and Resilient Connectivity Maintenance

The problem of $k-$connectivity control or $k-$redundancy control has also been studied [79, 80, 81, 82]. [81] introduced distributed algorithms for detecting $k-$connectivity of multi-robot graph. Work in [79] addressed the $k-$hop connectivity control where the robots stay connected with its $k-$hop neighbors at all time. In [80, 82] the robots are tasked to reconfigure their positions for meeting certain redundant connectivity constraints. These approaches often consider the connectivity maintenance as a separate optimization problem and hence has no optimal guarantee over the original robot's controllers. For less restrictive multi-robot control with constraint satisfaction, control barrier functions have been employed to encode a variety of inter-robot constraints and the resulting constrained control outputs lead to forward invariance of the satisfying set, i.e. robots remain collision free and connected under predefined fixed communication topology [29, 40, 78]. Although the resultant control outputs are optimized to stay as close to the original controllers with constraints, the predefined fixed communication topology has no guarantee regarding its optimality to the robot behaviors. In our work, we are optimizing both the activated $k-$connectivity constraints together with the controllers with proven optimality guarantees, so that the control revision with the invoked connectivity constraints is minimally invasive to the original behavior-prescribed controllers, thus allowing for flexible multi-robot behaviors with required network redundancy.

2.5 Multi-Robot Coordination in Sensor Coverage

In the multi-robot sensor coverage problem [4, 44], the sensing performance to optimize is determined by the distance between each robot and its assigned point to sense assuming negative correlation as well as the density function of the points. Solutions of such a locational optimization problem are known as the centroid of the Voronoi tessellation [83] and the algorithm is often referred to as the move-to-centroids controller navigating the robots towards the centroids of their Voronoi cells. However, most of them assume the prior knowledge of either the environ-

mental phenomenon distribution (often modelled as density function) [4, 84], or basis functions consisting of density function [85], which could be impractical in real-world application. To allow for online density function learning and adaptive coverage control, recent works [86] proposed to use two-stage decoupled processes that embed an on-line sampling process to first obtain an estimate of the density function and then follow the move-to-centroid control law in performing the multi-robot coverage. As mentioned in [1], this approach could demand unnecessarily larger number of samples to take before reaching the optimal locations.

To improve sampling efficiency, GP-based adaptive sampling methods [1, 87, 88] with Bayesian optimization framework [89] have been studied for information gathering to maximize the total value (e.g. utility or informativeness) of sequentially collected samples. [90, 91] extends adaptive sampling in multi-robot systems where the robots make sequential decisions regarding the next best waypoint to sample and then perform the path planning. The sampling criterion is often determined by predicted utility using GP model or information-theoretic criterion such as mutual information gain [92, 93] to maximize the sampled utility or model uncertainty reduction respectively. Besides GP-based adaptive sampling approaches, ergodic control methods have been proposed in [94, 95] to track the unknown spatial distribution by using ergodicity metrics to optimize time averaged trajectory in accordance with the expected spatial distribution, with the final trajectory statistics matching to the initially unknown distribution.

A recent work [1] proposed an efficient voronoi-based multi-robot informative adaptive sampling, where each robot only takes the best samples within its assigned partitioned region. [87] developed input-dependent model using the general approach of mixture of GPs [96] to accurately represent complex distributions with the linear combination of different GP models learned on-line. However, these approaches still require global information of the collected samples by all the robots. [97] proposed a distributed EM algorithm for classification tasks with Gaussian mixture model. Inspired by this work, in this thesis we first propose a centralized learning method to model the environment and employ the distributed EM algorithm with consensus learning as a heuristic method to develop a distributed learning framework using mixture of GPs to improve the local prediction accuracy, which leads to better multi-robot coverage performance.

Part I

Safe Control and Learning

Safety for large-scale autonomous system is critical yet challenging under real-world factors such as uncertainty, non-determinism and lack of complete information. Current model-based safe design often assumes perfect sensing and dynamics modeling in order to derive provable safety assurance [28, 29, 30, 31]. However, the pre-computed guarantees could be easily broken when deploying the system to the physical world, e.g. self-driving car crash due to inaccurately localized obstacles. On the other hand, learning-enabled decision making process [8, 9, 10, 11, 12] has been widely used to characterize the uncertainty with safety analysis for planning and control in presence of incomplete information, e.g. with partially modelled system dynamics and unknown environments. This introduces new challenges about how to properly trade off among robust safety assurance, optimal task performance (exploitation), and sample efficient exploration for leaning that inherently contradict each other.

In this part, we will focus on two problems as follows:

- ***Robust safety assurance under uncertainty*** *(Chapter 3)* by explicitly reasoning about the chance-constrained safety analysis that incorporates measurement and motion uncertainties into the multi-robot safe control design.
- ***Sample efficient safe reinforcement learning*** *(Chapter 4)* by leveraging the existing knowledge of partially modeled system dynamics for safe optimism-based model exploration and control with bounded learning efficiency and optimality guarantee.

Chapter 3

Multi-Robot Safe Control under Uncertainty

Safe operation is one of the most important task that needs to be addressed in the realm of autonomous systems. In real-world implementation, safe control in terms of collision avoidance for a large-scale multi-robot systems is a difficult challenge under uncertainty, non-determinism and lack of complete information. For example, consider the problem of building an automatic collision avoidance system (ACAS) for aerial robots that would scale up as the autonomous aerial traffic increases. Such a system needs to be computationally efficient for execution in real-time and robust to various real-world factors that include uncertainty, non-determinism and approximations made in the formulation of the system. Measurement uncertainty in the system arises from various estimation or prediction procedures in real-world that rely on sensory information (e.g. LIDARS, on-board GPS) being collected in real-time to get robots state information. On the other hand, non-determinism often arises from our in-ability to model various exogenous variables that are part of our operating environment, e.g. phenomena such as wind gusts. Ability to pro-actively deal with such measurement and motion uncertainty is fundamental in the safety considerations.

In this chapter, we develop a collision avoidance method for centralized and decentralized multi-robot systems that accounts for both measurement uncertainty and motion uncertainty. In particular, we propose Probabilistic Safety Barrier Certificates (PrSBC) using Control Barrier Functions [28] (See Section 3.2) to define the space of possible control actions that are probabilistically safe with theoretical guarantee. By formulating the chance constrained safety set into deterministic control constraints with PrSBC, the safety controllers can be computed by minimally modifying the existing unconstrained controller via a quadratic program subject

to the PrSBC constraints. This hence formally satisfies the collision-avoidance chance-constraints while staying as close to the original robot behaviors as possible. The key advantage of the approach is that no assumptions about the form of uncertainty are required other than finite support, also enabling worst-case guarantees.

Our work is most closely related to the work on safety barrier certificates (SBC) for multi-robot collision avoidance [29] using permissive control barrier functions (CBF) [28, 55]. While the prior work focused on deterministic settings, our goal here is to provide a safety envelope around an existing controller that accounts for uncertainties and non-determinism in a probabilistic setting. There are several **advantages** of the proposed PrSBC. First, in contrast of other probabilistic collision avoidance approaches that directly constrain the inter-robot distance [20, 21, 53], the proposed method produces a more permissive set for the controllers with a tighter bound. Second, the PrSBC naturally inherits the forward invariance from CBF, e.g. robots staying in the collision-free set at all time, and thus enabling us to prove guarantees throughout the continuous time scale. Finally, it is natural to apply the chance constrained collision avoidance with PrSBC under both centralized and decentralized settings to bridge learning based methodologies and model based safety-critical control with provable safety guarantee. For example, one may use learning techniques such as Gaussian Processes [98] to learn one or more partially unknown dynamical systems with noisy uncertainties and use our PrSBC approach to compute certified probablistically safe policies to collect more data for further improving models. We believe integrating dynamical system learning with our PrSBC framework to guarantee safe learning to control is an important future direction (See Chapter 4).

The key underlying assumption in our method is that the uncertainties arising due to sensor measurements, incomplete dynamics and other exogenous variables have finite support. This is a reasonable assumption for many of the multi-robot scenarios. For example, we can safely assume that true positions of robots, or the amount of wind gusts etc. are bounded within certain sensor specifications or physical parameters respectively. We use the task similar to automatic collision avoidance system for aerial robots as a motivating application. Our experiments explore the proposed computation of PrSBC controller in both centralized and decentralized settings, which can handle both the uncertainties as well as environmental disturbances while continuously guaranteeing safety. In summary, the core contributions of this chapter are as follows:

- A novel chance-constrained collision avoidance method with Probabilistic Safety Barrier Certificates (PrSBC) ensuring provable forward invariance under uncertainties with bounded support.

- Formal proof of existence of PrSBC in a closed form.
- Experimental results on the task similar to automatic collision avoidance for aerial robots that demonstrate efficiency, scalability and distributed computation.

3.1 Chance-Constrained Multi-Robot Collision Avoidance Problem

Consider a team of N robots moving in a shared d-dimensional workspace. Each robot $i \in \mathcal{I} = \{1, \dots, N\}$ is centered at the position $\mathbf{x}_i \in \mathcal{X}_i \subset \mathbb{R}^d$ and enclosed with a uniform safety radius $R_i \in \mathbb{R}$. The stochastic dynamical system $\dot{\mathbf{x}}_i$ in control affine form with noise and the noisy observation $\hat{\mathbf{x}}_i \in \mathbb{R}^d$ of each robot i are described as follows.

$$\begin{aligned} \dot{\mathbf{x}}_i &= f_i(\mathbf{x}_i, \mathbf{u}_i) + \mathbf{w}_i = F_i(\mathbf{x}_i) + G_i(\mathbf{x}_i)\mathbf{u}_i + \mathbf{w}_i \ , \quad \mathbf{w}_i \sim U(-\Delta\mathbf{w}_i, \Delta\mathbf{w}_i) \\ \hat{\mathbf{x}}_i &= \mathbf{x}_i + \mathbf{v}_i \ , \quad \mathbf{v}_i \sim U(-\Delta\mathbf{v}_i, \Delta\mathbf{v}_i) \end{aligned} \tag{3.1}$$

where $\mathbf{u}_i \in \mathcal{U}_i \subseteq \mathbb{R}^m$ denotes the control input. F_i and G_i are locally Lipschitz continuous. The deterministic system dynamics $f_i(\mathbf{x}_i, \mathbf{u}_i) = F_i(\mathbf{x}_i) + G_i(\mathbf{x}_i)\mathbf{u}_i$ in control affine form is general and could describe a large family of nonlinear systems, e.g. 3-dof differential drive vehicles with unicycle dynamics ([29, 99]), 12-dof quadrotors with underactuated system ([38, 100]), bipedal robots, automotive vehicle, and Segway robots [11, 28]. $\mathbf{w}_i, \mathbf{v}_i \in \mathbb{R}^d$ are the uniformly distributed process noise and the measurement noise respectively and considered as continuous independent random variables with finite support. A uniform distribution is a natural choice for these noise processes, however, most of our analysis does not require the exact form except that the support is finite. This finite support can vary at each time-point and come from a perception module, a state estimator or other physical parameters of the system.

Obstacle Model: Similar to the robots, other static or moving obstacles $k \in \mathcal{O} = \{1, \dots, K\}$ are also modeled as a rigid sphere located at $\mathbf{x}_k \in \mathbb{R}^d$ with the safety radius $R_k \in \mathbb{R}$. The measurement of obstacle location via robot sensor is modeled as $\hat{\mathbf{x}}_k = \mathbf{x}_k + \mathbf{v}_k \in \mathbb{R}^d$ with bounded uniformly distributed noise $\mathbf{v}_k \sim U(-\Delta\mathbf{v}_k, \Delta\mathbf{v}_k)$. As commonly assumed in other collision avoidance work ([20, 22, 101]), we consider the piece-wise constant obstacle' velocity to be detected by the robots as $\hat{\mathbf{u}}_k$ with a bounded noise, rendering the obstacle dynamics as $\dot{\mathbf{x}}_k = \mathbf{u}_k = \hat{\mathbf{u}}_k + \mathbf{w}_k \in \mathbb{R}^d,\ \mathbf{w}_k \sim U(-\Delta\mathbf{w}_k, \Delta\mathbf{w}_k)$. The finite supports of $\mathbf{v}_k, \mathbf{w}_k$ are also assumed to be known by the robots.

Denote the joint robot states as $\mathbf{x} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\} \in \mathcal{X} \subset \mathbb{R}^{d\times N}$ and the joint obstacle states as $\mathbf{x}_o = \{\mathbf{x}_1, \ldots, \mathbf{x}_K\} \in \mathcal{X}_o \in \mathbb{R}^{d\times K}$. For any pair-wise inter-robot or robot-obstacle collision avoidance between robots $i, j \in \mathcal{I}$ and obstacles $k \in \mathcal{O}$, the following condition define the safety of $\mathbf{x}$.

$$\begin{aligned} h^s_{i,j}(\mathbf{x}) &= \|\mathbf{x}_i - \mathbf{x}_j\|^2 - (R_i + R_j)^2, \qquad \forall i > j \\ h^s_{i,k}(\mathbf{x}, \mathbf{x}_o) &= \|\mathbf{x}_i - \mathbf{x}_k\|^2 - (R_i + R_k)^2, \qquad \forall i, k \end{aligned} \tag{3.2}$$

$$\begin{aligned} \mathcal{H}^s_{i,j} &= \{\mathbf{x} \in \mathbb{R}^{d\times N} : h^s_{i,j}(\mathbf{x}) \geq 0\} \qquad \forall i > j \\ \mathcal{H}^s_{i,k} &= \{\mathbf{x} \in \mathbb{R}^{d\times N} : h^s_{i,k}(\mathbf{x}, \mathbf{x}_o) \geq 0\}, \qquad \forall i, k \end{aligned} \tag{3.3}$$

The condition of $\forall i > j$ ensures each pairwise collision will be considered only once for the robot team. The sets of $\mathcal{H}^s_{i,j}$ and $\mathcal{H}^s_{i,k}$ indicate the safety set from which robots i and j, robot i and obstacle k will never collide. For the entire robotic team, the safety set is hence determined by the intersection of all $\mathcal{H}^s_{i,j}, \mathcal{H}^s_{i,k}$ as follows

$$\mathcal{H}^s = \bigcap_{\substack{i,j\in\mathcal{I} \\ i>j}} \mathcal{H}^s_{i,j} \bigcap_{\substack{i\in\mathcal{I} \\ k\in\mathcal{O}}} \mathcal{H}^s_{i,k} \tag{3.4}$$

As the robots only have access to the noisy measurements on the states of the robots and obstacles, the positions of the robots and obstacles are modeled as random variables with a finite support. The collision avoidance constraints can then be considered in a chance-constrained setting for each pairwise robots i, j and robot-obstacle i, k. Formally, given the minimum admissible probability of safety $\sigma, \sigma_o \in [0, 1]$ predefined by the user, it is required that:

$$\begin{aligned} \Pr(\mathbf{x}_i, \mathbf{x}_j \in \mathcal{H}^s_{i,j}) \geq \sigma\ , \quad \forall i > j \\ \Pr(\mathbf{x}_i, \mathbf{x}_k \in \mathcal{H}^s_{i,k}) \geq \sigma_o\ , \quad \forall i, k \end{aligned} \tag{3.5}$$

$\Pr(\cdot)$ indicates the probability of an event. Note that when σ, σ_o are set to 1, the conditions naturally lead to the worst-case collision avoidance with enlarged bounded volume as discussed in Section 3.3. Such worst-case guarantees can lead to a conservative behavior, thus often there are advantages in maintaining a probabilistic safety.

Assume that each robot has a task-related controller $\mathbf{u}^*_i \in \mathbb{R}^m$. We consider the chance-constrained collision avoidance as an one-step optimization problem that minimally modifies $\mathbf{u}^*_i$ for each robot i, while satisfying the desired probabilistic safety in Eq. (3.5). Formally we solve the following Quadratic Program (QP) under

the safety constraints:

$$\min_{\mathbf{u}\in\mathbb{R}^{mN}} \quad \sum_{i=1}^{N} \|\mathbf{u}_i - \mathbf{u}_i^*\|^2 \tag{3.6}$$

$$\text{s.t.} \quad \Pr(\mathbf{x}_i, \mathbf{x}_j \in \mathcal{H}_{i,j}^s) \geq \sigma \ , \quad \forall i > j \tag{3.7}$$

$$\Pr(\mathbf{x}_i, \mathbf{x}_k \in \mathcal{H}_{i,k}^s) \geq \sigma_o \ , \quad \forall i, k \tag{3.8}$$

$$\|\mathbf{u}_i\| \leq \alpha_i, \forall i, j \in \{1, \ldots, N\}, \ k \in \{1, \ldots, K\} \tag{3.9}$$

where $\mathbf{u} \in \mathcal{U} \subset \mathbb{R}^{mN}$ is the joint control inputs of all the robots with bounded magnitude $\alpha_i, \forall i$. Next, we briefly describe the background of Safety Barrier Certificates (SBC) [29]. Section 3.3 then presents our method of Probabilistic Safety Barrier Certificates (PrSBC) that utilizes control barrier functions [28] to remap the probabilistic safety set constraints Eq. (3.5) from the state space $\mathcal{X} \subset \mathbb{R}^{d\times N}$ to the control space $\mathcal{U} \subset \mathbb{R}^{mN}$.

3.2 Background: Safety Barrier Certificates using Control Barrier Functions

Recent advances in permissive control barrier functions [28, 29, 55] enable mechanisms that guarantee forward invariance of desired safety sets for robots, e.g. robots staying collision-free at all times by constraining the controllers. Here we first describe the formulation of the deterministic safety constraints utilizing the safety barrier certificates [29]. Without loss of generality, we can represent the desired safety set $\mathcal{H}^s$ in Eq. (3.4) using the function $h^s(\mathbf{x})$ from Eq. (3.2) as:

$$\mathcal{H}^s = \{\mathbf{x} \in \mathbb{R}^{d\times N} \quad | \quad h^s(\mathbf{x}) \geq 0\} \tag{3.10}$$

First, we summarize the conditions on controllers $\mathbf{u} \in \mathcal{U} \subseteq \mathbb{R}^{mN}$ based on Zeroing Control Barrier Functions (ZCBF) [55] and the Safety Barrier Certificates (SBC) [29] to guarantee *forward invariance* of safety. Formally, a safety condition is forward-invariant if $\mathbf{x}(t = 0) \in \mathcal{H}^s$ implies $\mathbf{x}(t) \in \mathcal{H}^s$ for all $t > 0$ with the designed satisfying controller at each time step. Readers are referred to [29, 55] for details. The Theorem of ZCBF and forward invariance from [29, 55] is summarized as the following Lemma.

Lemma 1. *Given the dynamical system in Eq. (3.1) without uncertainties, i.e.* $\mathbf{w}_i = 0, \forall i \in \mathcal{I}$ *and the set* $\mathcal{H}^s$ *defined by Eq. (3.10) for the continuously differentiable function* $h^s : \mathbb{R}^{d\times N} \to \mathbb{R}$. *The function* h^s *is a ZCBF and the admissible control space* $S(\mathbf{x})$ *for each time step can be defined as*

$$S(\mathbf{x}) = \{\mathbf{u} \in \mathcal{U} \mid \dot{h}^s(\mathbf{x}, \mathbf{u}) + \kappa(h^s(\mathbf{x})) \geq 0\}, \ \mathbf{x} \in \mathcal{X}, \tag{3.11}$$

where κ is an extended class-$\mathscr{K}$ function. Then any Lipschitz continuous controller satisfying $\mathbf{u} \in S(\mathbf{x})$ at each time step for the system Eq. (3.1) renders the set $\mathcal{H}^s$ forward invariant, i.e. robots stay collision-free at all times.

As described in [29], the extended class-$\mathscr{K}$ function κ such as $\kappa(r) = r^P$ with any positive odd integer P leads to different behaviors of the state of the system approaching the boundary of safety set $\mathcal{H}^s$ in Eq. (3.10). Similar to [29], in our case we use the particular choice of $\kappa(h^s(\mathbf{x})) = \gamma h^s(\mathbf{x})$ with $\gamma > 0$. In order to render a larger admissible control space $S(x)$, a very large value of $\gamma >> 0$ will be adopted. Thus the admissible control space in Eq. (3.11) induces the following pairwise constraints over the controllers, referred as *Safety Barrier Certificates (SBC)* [29]:

$$\begin{aligned} \mathcal{B}^s(\mathbf{x}) &= \{\mathbf{u} \in \mathbb{R}^{mN} : \dot{h}^s_{i,j}(\mathbf{x}, \mathbf{u}) + \gamma h^s_{i,j}(\mathbf{x}) \geq 0,\ \forall i > j\} \\ \mathcal{B}^o(\mathbf{x}, \mathbf{x}_o) &= \{\mathbf{u} \in \mathbb{R}^{mN} : \dot{h}^s_{i,k}(\mathbf{x}, \mathbf{x}_o, \mathbf{u}, \mathbf{u}^o) + \gamma h^s_{i,k}(\mathbf{x}, \mathbf{x}_o) \geq 0,\ \forall i, k\} \end{aligned} \tag{3.12}$$

where $\mathbf{u}^o \in \mathbb{R}^{dK}$ is the joint control input of all the obstacles not controllable by the robots. Here $\mathcal{B}^s(\mathbf{x}), \mathcal{B}^o(\mathbf{x}, \mathbf{x}_o)$ define the SBC for the inter-robot and robot-obstacle collision avoidance respectively, rendering the safety set $\mathcal{H}^s$ forward invariant: the robots will always stay safe, i.e. satisfying Eq. (3.3) at all times if they are initially collision free and the robots' joint control input $\mathbf{u}$ lies in the set $\mathcal{B}^s(\mathbf{x}) \cap \mathcal{B}^o(\mathbf{x}, \mathbf{x}_o)$. One of the useful properties of Eq. (3.12) is that they induce linear constraints over both the pair-wise control inputs $\mathbf{u}_i$ and $\mathbf{u}_j$ (inter-robot) and control input $\mathbf{u}_i$ (robot-obstacle).

3.3 Probabilistic Safety Barrier Certificates (PrSBC)

We seek a probabilistic version of Lemma 1 that implies the SBC in Eq. (3.12) as a sufficient condition for the forward invariance of $\mathcal{H}^s$ in Eq. (3.10). Given the assumption that each pairwise robots are initially collision-free, i.e. $\mathbf{x}_i, \mathbf{x}_j \in \mathcal{H}^s_{i,j}$ at $t = 0$ and the sufficiency condition in Lemma 1, we have $\mathbf{u}_i, \mathbf{u}_j \in \mathcal{B}^s_{i,j}(\mathbf{x}) \implies \mathbf{x}_i, \mathbf{x}_j \in \mathcal{H}^s_{i,j}$ and $\mathbf{u}_i, \mathbf{u}_j \notin \mathcal{B}^s_{i,j}(\mathbf{x}) \not\implies \mathbf{x}_i, \mathbf{x}_j \notin \mathcal{H}^s_{i,j}$. Hence it is straightforward to show that $\Pr(\mathbf{u}_i, \mathbf{u}_j \in \mathcal{B}^s_{i,j}(\mathbf{x})) \leq \Pr(\mathbf{x}_i, \mathbf{x}_j \in \mathcal{H}^s_{i,j})$ and $\Pr(\mathbf{u}_i, \mathbf{u}_k \in \mathcal{B}^o_{i,k}(\mathbf{x}, \mathbf{x}_o)) \leq \Pr(\mathbf{x}_i, \mathbf{x}_k \in \mathcal{H}^s_{i,k})$. Consequently, we can derive the following inter-robot and robot-obstacle probabilistic collision free sufficiency conditions corresponding to Eq. (3.5):

$$\begin{aligned} \Pr(\mathbf{u}_i, \mathbf{u}_j \in \mathcal{B}^s_{i,j}(\mathbf{x})) \geq \sigma &\implies \Pr(\mathbf{x}_i, \mathbf{x}_j \in \mathcal{H}^s_{i,j}) \geq \sigma, \quad \forall i > j \\ \Pr(\mathbf{u}_i, \mathbf{u}_k \in \mathcal{B}^o_{i,k}(\mathbf{x}, \mathbf{x}_o)) \geq \sigma_o &\implies \Pr(\mathbf{x}_i, \mathbf{x}_k \in \mathcal{H}^s_{i,k}) \geq \sigma_o, \quad \forall i,\ k \end{aligned} \tag{3.13}$$

Intuitively, these conditions allow us to translate the probabilistic safety constraints from the state-space directly to the the controls, thereby enabling consideration of safety when reasoning about the next control action. Note that the above condition is over the joint control space of multiple robots, hence far less restrictive than other methods that only constrain ego robots motion.

Given these reformulated collision-free chance constraints over controllers, we now formally define the Probabilistic Safety Barrier Certificates (PrSBC):

Definition 2. ***Probabilistic Safety Barrier Certificates (PrSBC):*** *Given a confidence level $\sigma \in [0, 1]$, PrSBC determines the admissible control space $\mathcal{S}_u^\sigma$ at each time-step guaranteeing the chance-constrained safety condition in Eq. (3.5) and are defined as the intersection of n different half-spaces where n is the total number of pairwise deterministic inter-robot constraints.*

$$\mathcal{S}_u^\sigma = \{\mathbf{u} \in \mathbb{R}^{mN} \mid A_{ij}^\sigma \mathbf{u} \leq b_{ij}^\sigma, \quad \forall i > j,\ A^\sigma \in \mathbb{R}^{n \times mN},\ b^\sigma \in \mathbb{R}^n\} \tag{3.14}$$

Here we first introduce the definition and form of PrSBC. The computation of $A^\sigma \in \mathbb{R}^{n \times mN}, b^\sigma \in \mathbb{R}^n$ determined by σ will be given in the latter part of Eq. (3.24) and (3.25) for inter-robot and robot-obstacle collision avoidance. The PrSBC hence characterizes the admissible safe control space for the multi-robot team with probabilistic safety guarantee.

Theoretical Analysis of PrSBC Next, we provide theoretical analysis that discusses existence of PrSBC, justifies representation of PrSBC as intersection of half-spaces, and shows how they can be computed and enable us to derive probabilistic safe controllers.

Theorem 3. ***Existence of PrSBC:*** *Assuming all pairwise robots are initially collision-free at $t = 0$, i.e. Eq. (3.2) holds true for all possible value of random state variables $\mathbf{x}_i \in [\hat{\mathbf{x}}_i - \Delta\mathbf{v}_i, \hat{\mathbf{x}}_i + \Delta\mathbf{v}_i], \forall i \in \mathcal{I}$, then the PrSBC defined in Eq. (3.14) is guaranteed to exist for any given confidence level $\sigma \in [0, 1]$.*

Proof. We start by proving the existence of PrSBC between each pairwise robots i and j with any user-defined confidence level $\sigma \in [0, 1]$. Consider the sufficiency condition of $\Pr(\mathbf{u}_i, \mathbf{u}_j \in \mathcal{B}_{i,j}^s(\mathbf{x})) \geq \sigma$ in Eq. (3.13) with pairwise version of Eq. (3.12) that renders desired chance constrained safety $\Pr(\mathbf{x}_i, \mathbf{x}_j \in \mathcal{H}_{i,j}^s) \geq \sigma$. With $\dot{h}_{i,j}^s(\mathbf{x}, \mathbf{u}) = \frac{\partial h_{i,j}^s}{\partial \mathbf{x}}(\mathbf{x})(\Delta F_{i,j}(\mathbf{x}) + G_{i,j}(\mathbf{x})\mathbf{u}_{i,j} + \Delta\mathbf{w}_{i,j})$, we can then re-write the sufficiency condition $\Pr(\mathbf{u}_i, \mathbf{u}_j \in \mathcal{B}_{i,j}^s(\mathbf{x})) \geq \sigma$ in Eq. (3.13) using Eq. (3.12) as follows:

$$\begin{aligned}
&\Pr(\mathbf{u}_i, \mathbf{u}_j \in \mathcal{B}_{i,j}^s(\mathbf{x})) \geq \sigma : \\
&\iff \Pr\left(\frac{\partial h_{i,j}^s}{\partial \mathbf{x}}(\mathbf{x}) G_{i,j}(\mathbf{x})\mathbf{u}_{i,j} \geq -\gamma h_{i,j}^s(\mathbf{x}) - \frac{\partial h_{i,j}^s}{\partial \mathbf{x}}(\mathbf{x})\big(\Delta F_{i,j}(\mathbf{x}) + \Delta\mathbf{w}_{i,j}\big)\right) \geq \sigma
\end{aligned} \tag{3.15}$$

where

$$\frac{\partial h_{i,j}^s}{\partial \mathbf{x}}(\mathbf{x})G_{i,j}(\mathbf{x})\mathbf{u}_{i,j} = 2(\mathbf{x}_i - \mathbf{x}_j)^T\Big(G_i(\mathbf{x}_i)\mathbf{u}_i - G_j(\mathbf{x}_j)\mathbf{u}_j\Big)$$
$$\frac{\partial h_{i,j}^s}{\partial \mathbf{x}}(\mathbf{x})\Big(\Delta F_{i,j}(\mathbf{x}) + \Delta\mathbf{w}_{i,j}\Big) = 2(\mathbf{x}_i - \mathbf{x}_j)^T\Big(F_i(\mathbf{x}_i) - F_j(\mathbf{x}_j) + \mathbf{w}_i - \mathbf{w}_j\Big)$$

Let's denote the process noise difference $\Delta\mathbf{w}_{i,j} = \mathbf{w}_i - \mathbf{w}_j \sim Q_{i,j}$ with the finite support $\text{supp}(Q_{i,j}) = \Big[-(\Delta\mathbf{w}_i + \Delta\mathbf{w}_j),\ (\Delta\mathbf{w}_i + \Delta\mathbf{w}_j)\Big]$ and state difference $\Delta\mathbf{x}_{i,j} = \mathbf{x}_i - \mathbf{x}_j \sim T_{i,j}$ with the finite support $\text{supp}(T_{i,j}) = \Big[(\hat{\mathbf{x}}_i - \hat{\mathbf{x}}_j) - (\Delta\mathbf{v}_i + \Delta\mathbf{v}_j),\ (\hat{\mathbf{x}}_i - \hat{\mathbf{x}}_j) + (\Delta\mathbf{v}_i + \Delta\mathbf{v}_j)\Big]$. Moreover, given the assumed uniform distributions of $\mathbf{w}_i, \mathbf{w}_j, \mathbf{x}_i, \mathbf{x}_j$, the distributions $T_{i,j}, Q_{i,j}$ are hence two different symmetric trapezoid distributions with finite supports. Then by substituting Eq. (3.16) into Eq. (3.15) and after re-organization, we have

$$\Pr\Bigg(\Bigg[\Delta\mathbf{x}_{i,j} + \frac{\overbrace{G_{i,j}\mathbf{u}_{i,j} + \Delta F_{i,j} + \Delta\mathbf{w}_{i,j}}^{\dot{\mathbf{x}}_i - \dot{\mathbf{x}}_j}}{\gamma}\Bigg]^2 \geq R_{ij}^2 + \Bigg[\frac{\overbrace{G_{i,j}\mathbf{u}_{i,j} + \Delta F_{i,j} + \Delta\mathbf{w}_{i,j}}^{\dot{\mathbf{x}}_i - \dot{\mathbf{x}}_j}}{\gamma}\Bigg]^2\Bigg) \geq \sigma \tag{3.16}$$

where

$$G_{i,j}\mathbf{u}_{i,j} = G_i(\mathbf{x}_i)\mathbf{u}_i - G_j(\mathbf{x}_j)\mathbf{u}_j\ , \Delta F_{i,j} = F_i(\mathbf{x}_i) - F_j(\mathbf{x}_j)\ , R_{ij} = R_i + R_j > 0 \tag{3.17}$$

Thus consider the following set of random variable $\Delta\mathbf{x}_{i,j}$ from its own finite support and Eq. (3.16):

$$\begin{aligned}
\Omega_{i,j}(\Delta\mathbf{x}_{i,j}) &= \text{supp}(T_{i,j}) \\
&= \Big[(\hat{\mathbf{x}}_i - \hat{\mathbf{x}}_j) - (\Delta\mathbf{v}_i + \Delta\mathbf{v}_j),\quad (\hat{\mathbf{x}}_i - \hat{\mathbf{x}}_j) + (\Delta\mathbf{v}_i + \Delta\mathbf{v}_j)\Big] \\
\Omega_{i,j}^{\mathbf{u}}(\Delta\mathbf{x}_{i,j}) &= \Bigg\{\Delta\mathbf{x}_{i,j} \in \mathbb{R}^d \ \Bigg|\ \Bigg[\Delta\mathbf{x}_{i,j} + \frac{\overbrace{G_{i,j}\mathbf{u}_{i,j} + \Delta F_{i,j} + \Delta\mathbf{w}_{i,j}}^{\dot{\mathbf{x}}_i - \dot{\mathbf{x}}_j}}{\gamma}\Bigg]^2 \\
&\qquad\qquad \geq R_{ij}^2 + \Bigg[\frac{\overbrace{G_{i,j}\mathbf{u}_{i,j} + \Delta F_{i,j} + \Delta\mathbf{w}_{i,j}}^{\dot{\mathbf{x}}_i - \dot{\mathbf{x}}_j}}{\gamma}\Bigg]^2\Bigg\}
\end{aligned} \tag{3.18}$$

Note that the set of $\Omega^{\mathbf{u}}_{i,j}(\Delta\mathbf{x}_{i,j})$ representing the space outside a $(d-1)-$sphere for $\Delta\mathbf{x}_{i,j}$ in $d-$dimensional space. It is determined by the pairwise value of $\mathbf{u}_i, \mathbf{u}_j$ through $G_{i,j}\mathbf{u}_{i,j} = G_i(\mathbf{x}_i)\mathbf{u}_i - G_j(\mathbf{x}_j)\mathbf{u}_j$ as defined in Eq. (3.17). It is thus straightforward to show that the condition in Eq. (3.16) is equivalent to:

$$\Pr\Big(\Delta\mathbf{x}_{i,j} \in \Omega_{i,j} \cap \Omega^{\mathbf{u}}_{i,j}\Big) \geq \sigma \tag{3.19}$$

To prove the guaranteed existence of PrSBC, we need to show there always exists at least one solution of pairwise $\mathbf{u}_i, \mathbf{u}_j$ such that Eq. (3.19) holds for any given value of $\sigma \in [0, 1]$. First let's consider any pairwise $\mathbf{u}_i = \mathbf{u}^0_i, \mathbf{u}_j = \mathbf{u}^0_j$ leading to the joint control inputs $\mathbf{u}^0$ such that $\dot{\mathbf{x}}_i - \dot{\mathbf{x}}_j = 0$ in Eq. (3.18), then we have the following condition representing the space outside the $(d-1)-$sphere for $\Delta\mathbf{x}_{i,j}$:

$$\Omega^{\mathbf{u}^0}_{i,j}(\Delta\mathbf{x}_{i,j}) = \left\{ \Delta\mathbf{x}_{i,j} \in \mathbb{R}^d \;\middle|\; \Delta\mathbf{x}^2_{i,j} \geq R^2_{ij} \right\} \tag{3.20}$$

Recall that all pairwise robots are assumed to be initially collision-free, i.e. $\Delta\mathbf{x}^2_{i,j} \geq R^2_{ij}$, thus Eq. (3.19) holds true at all times for any given $\sigma \in [0, 1]$ since $\Pr\Big(\Delta\mathbf{x}_{i,j} \in \Omega_{i,j} \cap \Omega^{\mathbf{u}^0}_{i,j}\Big) = 1$ under one possible solution of joint control inputs $\mathbf{u} = \mathbf{u}^0$ that leads to $\dot{\mathbf{x}}_i - \dot{\mathbf{x}}_j = 0$. More generally, as the value of $||\dot{\mathbf{x}}_i - \dot{\mathbf{x}}_j||$ grows from 0 with other value of $\mathbf{u} \neq \mathbf{u}^0$, the corresponding $(d-1)-$sphere of $\Omega^{\mathbf{u}}_{i,j}(\Delta\mathbf{x}_{i,j})$ in Eq. (3.18) will continuously shift from the origin and gradually intersect with the bounding box of $\Omega_{i,j}(\Delta\mathbf{x}_{i,j}) = \text{supp}(T_{i,j})$ in Eq. (3.18). This leads to $\Pr\Big(\Delta\mathbf{x}_{i,j} \in \Omega_{i,j} \cap \Omega^{\mathbf{u}}_{i,j}\Big)$ continuously decrease from 1 to 0. Hence for any given value $\sigma \in [0, 1]$, it is always feasible to solve for at least a particular pairwise $\mathbf{u}_i, \mathbf{u}_j$ such that $\Pr\Big(\Delta\mathbf{x}_{i,j} \in \Omega_{i,j} \cap \Omega^{\mathbf{u}}_{i,j}\Big) = \sigma$, or $\Pr\Big(\Delta\mathbf{x}_{i,j} \in \Omega_{i,j} \cap \Omega^{\mathbf{u}}_{i,j}\Big) > \sigma$ so that Eq. (3.19) holds true. This pairwise $\mathbf{u}_i, \mathbf{u}_j$ could then serve as a hyperplane dividing the corresponding subspace of joint control space of $\mathbf{u}$ with one side in the form of Eq. (3.14) rendering the satisfying probabilistic safety between robot i, j. And by repeatedly updating the hyperplane at each time step in Eq. (3.14), the constrained step-wise controllers $\mathbf{u}_i, \mathbf{u}_j$ ensure the probabilistic safety is guaranteed at all times given the forward invariance in Eq. (3.13). It is then straightforward to extend to all pairwise inter-robot collision avoidance constraints and thus concludes the proof. □

Computation of PrSBC With the proved existence of PrSBC from Theorem 3, we will provide computation of PrSBC that yields the solution in Eq. (3.24). Lets consider inter-robot collision avoidance first. Given any confidence level $\sigma \in [0, 1]$,

the equivalent chance constraint of $\Pr(\mathbf{u}_i, \mathbf{u}_j \in \mathcal{B}^s_{i,j}(\mathbf{x})) \geq \sigma$ in Eq. (3.15) and its re-written form in Eq. (3.16) can be transformed into a deterministic linear constraint over pairwise controllers $\mathbf{u}_i, \mathbf{u}_j$ in the form of Eq. (3.14). While it is computationally intractable to get closed form solutions from Eq. (3.16), we obtain an approximate solution by considering the condition on each individual dimension $\Delta\mathbf{x}^l_{i,j} \in \{\Delta\mathbf{x}^1_{i,j}, \ldots, \Delta\mathbf{x}^d_{i,j}\} \subset \mathbb{R}^d$ of $\Delta\mathbf{x}_{i,j}, \forall l = 1, \ldots, d$ for Eq. (3.16). Hence, we introduce a sufficiency condition to Eq. (3.16) in each dimension as follows, so that ensuring Eq. (3.21) $\implies$ Eq. (3.16).

$$\Pr\left((\Delta\mathbf{x}^l_{i,j})^2 + 2 \cdot \frac{(G_{i,j}\mathbf{u}_{i,j})_l + \Delta F^l_{i,j}}{\gamma}\Delta\mathbf{x}^l_{i,j} \geq R^2_{ij} - B^l_{i,j}\right) \geq \sigma \tag{3.21}$$

where $(G_{i,j}\mathbf{u}_{i,j})_l = (G_i\mathbf{u}_i - G_j\mathbf{u}_j)_l \in \mathbb{R}$ and $\Delta F^l_{i,j} = F^l_i - F^l_j \in \mathbb{R}$ denote the lth element of $G_{i,j}\mathbf{u}_{i,j} \in \mathbb{R}^{d\times 1}$ and $\Delta F_{i,j} \in \mathbb{R}^{d\times 1}$ respectively. $B^l_{i,j} = -\frac{2}{\gamma}\max||\Delta\mathbf{w}^l_{i,j}|| \cdot ||\Delta\mathbf{x}^l_{i,j}|| \in \mathbb{R}$ with $\Delta\mathbf{w}^l_{i,j} \in \mathbb{R}$ as the lth element in $\Delta\mathbf{w}_{i,j} \in \mathbb{R}^d$. To simplify the discussion we assume piece-wise $G_i, G_j \in \mathbb{R}^{d\times m}, F_i, F_j \in \mathbb{R}^{d\times 1}$ in Eq. (3.1) can be approximated by Taylor theorem at points $\hat{x}_i, \hat{x}_j$ respectively. Then, we have equivalent condition of Eq. (3.21) as follows

$$\Pr\left(\Delta\mathbf{x}^l_{i,j} \leq -\frac{(G_{i,j}\mathbf{u}_{i,j})_l + \Delta F^l_{i,j}}{\gamma} - D^l_{i,j} \;\text{ OR }\; \Delta\mathbf{x}^l_{i,j} \geq -\frac{(G_{i,j}\mathbf{u}_{i,j})_l + \Delta F^l_{i,j}}{\gamma} + D^l_{i,j}\right) \geq \sigma \tag{3.22}$$

where

$$D^l_{i,j} = \sqrt{\frac{\left((G_{i,j}\mathbf{u}_{i,j})_l + \Delta F^l_{i,j}\right)^2}{\gamma^2} + R^2_{ij} - B^l_{i,j}}$$

Recall the finite support of $\Delta\mathbf{x}_{i,j}$ with its symmetric trapezoid distribution $T_{i,j}$ in Eq. (3.18), we can find alternative condition to enforce either of the condition in Eq. (3.22), e.g. $\Pr(\Delta\mathbf{x}^l_{i,j} \leq \cdot) \geq \sigma$ or $\Pr(\Delta\mathbf{x}^l_{i,j} \geq \cdot) \geq \sigma$ so that Eq. (3.22) is definitely lower bounded by σ. We assume $\sigma > 0.5$ and denote $e^{l,1}_{i,j} = \Phi^{-1}(\sigma)$ and $e^{l,2}_{i,j} = \Phi^{-1}(1-\sigma)$ with $\Phi^{-1}(\cdot)$ as the inverse cumulative distribution function (CDF) of the random variable $\Delta\mathbf{x}^l_{i,j} = \mathbf{x}^l_i - \mathbf{x}^l_j$ in Eq. (3.16) along each lth dimension. We have $\sigma > 0.5 \implies e^{l,1}_{i,j} > e^{l,2}_{i,j}$. Thus, we derive a formal sufficiency condition for Eq. (3.22) as follows.

$$\exists l = 1, \ldots, d: \quad -2\mathbf{e}^l_{i,j}(G_{i,j}\mathbf{u}_{i,j})_l/\gamma \leq (\mathbf{e}^l_{i,j})^2 - R^2_{ij} + B^l_{ij} + 2\mathbf{e}^l_{i,j}\Delta F^l_{i,j}/\gamma \tag{3.23}$$

where

$$\mathbf{e}^l_{i,j} = \begin{cases} e^{l,2}_{i,j}, & e^{l,2}_{i,j} > 0 \\ e^{l,1}_{i,j}, & e^{l,1}_{i,j} < 0 \\ 0, & e^{l,2}_{i,j} \leq 0 \text{ and } e^{l,1}_{i,j} \geq 0 \end{cases}$$

Note that $\mathbf{e}_{i,j}^l = 0$ implies the two robots i and j overlap along the lth dimension, e.g. two drones flying to the same 2D locations but with different altitudes. As it is assumed any pairwise robots are initially collision free and from the forward invariance property discussed above, $\mathbf{e}_{i,j}^l = 0$ only happens along at most $d-1$ dimensions. To that end, we can formally construct the PrSBC as follows in the closed form of Eq. (3.14), by adding up the linear constraints in Eq. (3.23) for all d dimensions.

$$\mathcal{S}_u^\sigma = \{\mathbf{u} \in \mathbb{R}^{mN} | -2\mathbf{e}_{i,j}^T(G_i\mathbf{u}_i - G_j\mathbf{u}_j)/\gamma \leq ||\mathbf{e}_{i,j}||^2 - d\cdot R_{ij}^2 + B_{ij} + 2\mathbf{e}_{i,j}^T \Delta F_{i,j}/\gamma, \quad \forall i > j\} \tag{3.24}$$

where $\mathbf{e}_{i,j} = [\mathbf{e}_{i,j}^1, \ldots, \mathbf{e}_{i,j}^d]^T \in \mathbb{R}^{d\times 1}$ and $B_{ij} = \Sigma_{l=1}^d B_{ij}^l$. This invokes a set of pairwise linear constraints over the robot controllers such that the inter-robot probabilistic collision avoidance in Eq. (3.5) holds true at all times. Note the PrSBC constraint in Eq. (3.24) is a conservative approximation of Eq. (3.23) by adding up the constraints for each dimension, and therefore guarantee $\Pr(\mathbf{u}_i, \mathbf{u}_j \in \mathcal{B}_{i,j}^s(\mathbf{x})) \geq \sigma$.

Remark 1. *For other forms of distribution than uniform but with finite support for the noise models, the only change is the computation of inverse CDF to specify different $e_{i,j}^{l,1}, e_{i,j}^{l,2}$ and the rest of the derivations of PrSBC still holds and ensure chance-constrained safety. For Gaussian distribution with infinite support, we can still compute a finite support based on the corresponding inverse CDF from σ for $\Delta\mathbf{x}_{i,j}$ at each dimension.*

Proposition 4. ***PrSBC for Robot-Obstacle Collision Avoidance:*** *Consider the dynamic obstacle model and the PrSBC for pairwise robots in Eq. (3.24), the PrSBC for robot-obstacle collision avoidance with a given confidence level $\sigma_o \in [0,1]$ can be defined as follows.*

$$\begin{aligned}\mathcal{S}_u^{\sigma_o} = \{\mathbf{u} \in \mathbb{R}^{mN} | -2\mathbf{e}_{i,k}'^T G_i\mathbf{u}_i/\gamma \leq -2\mathbf{e}_{i,k}'^T\hat{\mathbf{u}}_k/\gamma + ||\mathbf{e}_{i,k}'||^2 - d\cdot R_{ik}^2 \\ + B_{ik} + 2\mathbf{e}_{i,k}'^T F_i/\gamma, \forall i,k\}\end{aligned} \tag{3.25}$$

where the intermediate variables of $\mathbf{e}_{i,k}', B_{ik}$ are computed the same way as for inter-robot case Eq. (3.24).

Proposition 5. *PrSBC in Eq. (3.24) can be considered as a generalized SBC when the dynamics model in Eq. (3.1) is deterministic and without any uncertainty, i.e. $\mathbf{w}, \mathbf{v} = 0$. In this case, we have $\mathbf{e}_{i,j} = \Delta\mathbf{x}_{i,j} = \Delta\hat{\mathbf{x}}_{i,j} = \hat{\mathbf{x}}_i - \hat{\mathbf{x}}_j$ and $B_{ij} = 0$ in Eq. (3.24), and then it degenerates to the constraint in Eq. (3.12) same as SBC ([29]).*

Proposition 6. ***Worst-case Collision Avoidance:*** *when confidence level is set to be $\sigma = 1$, the PrSBC in Eq. (3.24) hence leads to the worst-case driven collision avoidance with $\mathbf{e}_{i,j}$ specified by the boundary of finite support of $\Delta\mathbf{x}_{i,j}$, yielding most conservative motions of $\mathbf{u}$ for all of the robots.*

3.4 Optimization-based Controllers with Probabilistic Safety

The constrained control space specified by PrSBC in Eq. (3.24) and (3.25) ensures the forward invariance of probabilistic safety in Eq. (3.13). Hence, we can reformulate the original QP problem in Eq. (3.6) with the PrSBC constraints. The probabilistic safety controller can thus obtained by minimally modifying the original controller $\mathbf{u}^*$ that the system wished to execute. Formally, we can write this as:

$$\mathbf{u} = \underset{\mathbf{u}\in\mathbb{R}^{mN}}{\arg\min} \sum_{i=1}^{N} \|\mathbf{u}_i - \mathbf{u}_i^*\|^2 \tag{3.26}$$

$$\text{s.t.} \quad \mathbf{u} \in \mathcal{S}_{\mathbf{u}}^{\sigma} \bigcap \mathcal{S}_{\mathbf{u}}^{\sigma_o}, \quad \|\mathbf{u}_i\| \leq \alpha_i, \forall i = 1, \ldots, N \tag{3.27}$$

As mentioned the PrSBC constraints Eq. (3.27) invoke a set of linear constraints over robot controllers and hence the probabilistic safety controller Eq. (3.26) can be solved efficiently in real-time with guaranteed specified probability of safety. The resulting safe controller per time step ensures for all $t \in [0, \tau]$, $\mathbf{u} \in \mathcal{S}_{\mathbf{u}}^{\sigma} \bigcap \mathcal{S}_{\mathbf{u}}^{\sigma_o}$, then our approach guarantees chance constrained safety along the entire time horizon $[0, \tau]$.

Remark 2. *(Probability of collision for the full trajectory) Denoting n_t as the total number of time steps during execution, the probability of collision avoidance between robot i, j for the whole trajectory is lower bounded as $Pr\big(\bigcap_{t=1}^{n_t}(\mathbf{x}_i^t, \mathbf{x}_j^t \in \mathcal{H}_{i,j}^s(t))\big) = \prod_{t=1}^{n_t} Pr(\mathbf{x}_i^t, \mathbf{x}_j^t \in \mathcal{H}_{i,j}^s(t)) \geq \sigma^{n_t}$. Here we assume the probability of collision avoidance at each time step is independent for practical purposes as done in [20]. In theory, by selecting $\sigma = exp(\frac{ln\sigma_{all}}{n_t})$ one could achieve a lower bounded joint collision free threshold of σ_{all} for the full trajectory. However, it could be over-conservative in the long run, and hence we use step-wise threshold to construct local collision constraints. An alternative is to impose discounting factor $\beta < 1$ so that the penalty of future violation probabilities is relaxed, i.e. step-wise threshold σ renders the same bounded joint threshold for the whole trajectory $\sum_{t=1}^{n_t}(\beta)^t Pr(\mathbf{x}_i^t, \mathbf{x}_j^t \in \mathcal{H}_{i,j}^s(t)) \geq \sigma$ if given discounting factor $\beta > 0.5$ (see [20]).*

While the controller in Eq. (3.26) is in a centralized setting, we can also derive a decentralized version of the PrSBC and the controllers. The mechanism is similar to [29] which was originally applied to deterministic SBC.

Consider the PrSBC in Eq. (3.24) and denote $b_{ij}^{\sigma} = ||\mathbf{e}_{i,j}||^2 - d \cdot R_{ij}^2 + B_{ij} + 2\mathbf{e}_{i,j}^T \Delta F_{i,j}/\gamma$. We can then separate the linear pairwise PrSBC constraint between

robot i and j in the following two inequalities:

$$-2\mathbf{e}_{i,j}^T G_i/\gamma \cdot \mathbf{u}_i \leq p_{ij}/(p_{ij}+p_{ji}) \cdot b_{ij}^{\sigma}, \quad 2\mathbf{e}_{i,j}^T G_j/\gamma \cdot \mathbf{u}_j \leq p_{ji}/(p_{ij}+p_{ji}) \cdot b_{ij}^{\sigma} \tag{3.28}$$

Here $p_{ij}, p_{ji} \in [0,1]$ represents the responsibility that each of the two robot takes regarding satisfying this pairwise probabilistic safety constraint. The knowledge of p_{ij}, p_{ji} can be either predefined and assumed known by all robots, in which case each robot does not need to communicate and simply avoid collision in a reciprocal manner, or can be communicated locally between pairwise robots in a more cooperative manner. Note that Eq. (3.28) is a sufficient condition of Eq. (3.24) and hence still guarantees the required probabilistic safety.

With such decentralized constraints, we have the decentralized probabilistic safety controller for each robot i as follows.

$$\mathbf{u}_i = \underset{\mathbf{u}_i \in \mathbb{R}^m}{\arg\min} \|\mathbf{u}_i - \mathbf{u}_i^*\|^2 \tag{3.29}$$

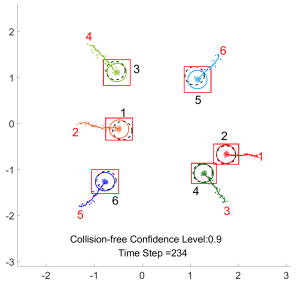
$$\text{s.t.} \quad \mathbf{u}_i \in \mathcal{S}_{\mathbf{u}_i}^{\sigma} \bigcap \mathcal{S}_{\mathbf{u}_i}^{\sigma_o}, \quad \|\mathbf{u}_i\| \leq \alpha_i \tag{3.30}$$

with $\mathcal{S}_{\mathbf{u}_i}^{\sigma} = \{\mathbf{u}_i \in \mathbb{R}^m | -2\mathbf{e}_{i,j}^T G_i/\gamma \cdot \mathbf{u}_i \leq p_{ij}/(p_{ij}+p_{ji}) \cdot b_{ij}^{\sigma}, \forall j \in \mathcal{N}_i\}$ and $\mathcal{S}_{\mathbf{u}_i}^{\sigma_o} = \{\mathbf{u}_i \in \mathbb{R}^m | -2\mathbf{e}_{i,k}'^T G_i \mathbf{u}_i/\gamma \leq -2\mathbf{e}_{i,k}'^T \hat{\mathbf{u}}_k/\gamma + ||\mathbf{e}_{i,k}'||^2 - R_{ik}^2 + B_{ik} + 2\mathbf{e}_{i,k}'^T F_i/\gamma, \forall k \in \mathcal{K}\}$. $\mathcal{N}_i$ denotes the set of neighboring robots around robot i.

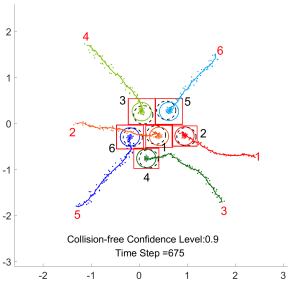
This decentralized PrSBC controller does not require centralized optimization process as for Eq. (3.26), but may thus lead to more conservative motion of robots or infeasible solution in extreme cases. In this case the robots will simply decelerate to zero velocities to ensure safety, which may cause the deadlock preventing the robots from achieving the goals. Some deconfliction policies for deterministic SBC can thereby be employed, such as the one suggested in [102]. Readers are referred to [102] for detailed solutions.

3.5 Results

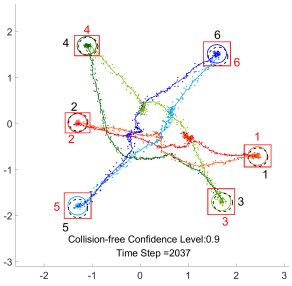
To evaluate the performance of our PrSBC method with optimization-based controllers, we designed four sets of experiments in Matlab simulation and a near-realistic simulation environment: i) a simulation example using 6 simulated mobile robots with unicycle dynamics to show inter-robot probabilistic collision avoidance in a centralized manner, ii) a simulation example using 7 simulated mobile robots with unicycle dynamics to show inter-robot and robot-obstacle probabilistic collision avoidance in a decentralized manner, iii) 50 random trials of simulations with various number of mobile robots to show the guaranteed probabilistic safety, and



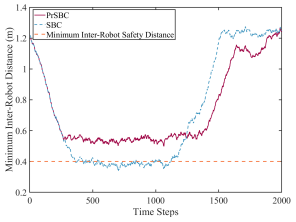
(a) Time Step = 234 (PrSBC)



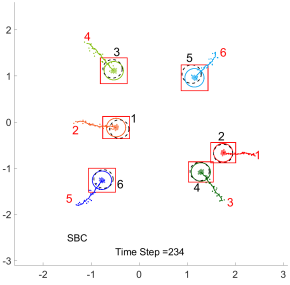
(b) Time Step = 675 (PrSBC)



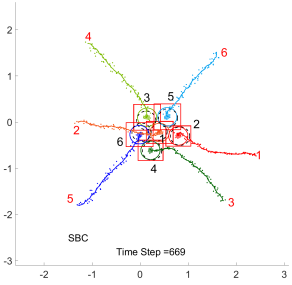
(c) Time Step = 2037 (PrSBC)



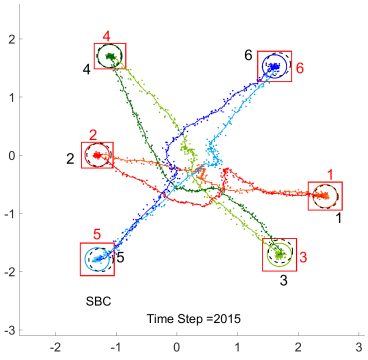
(d) Minimum true inter-robot distance



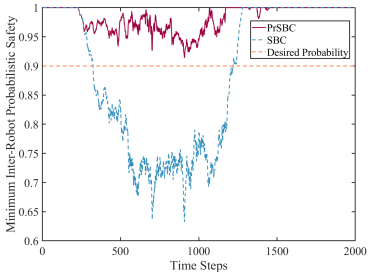
(e) Time Step = 234 (SBC)



(f) Time Step = 669 (SBC)

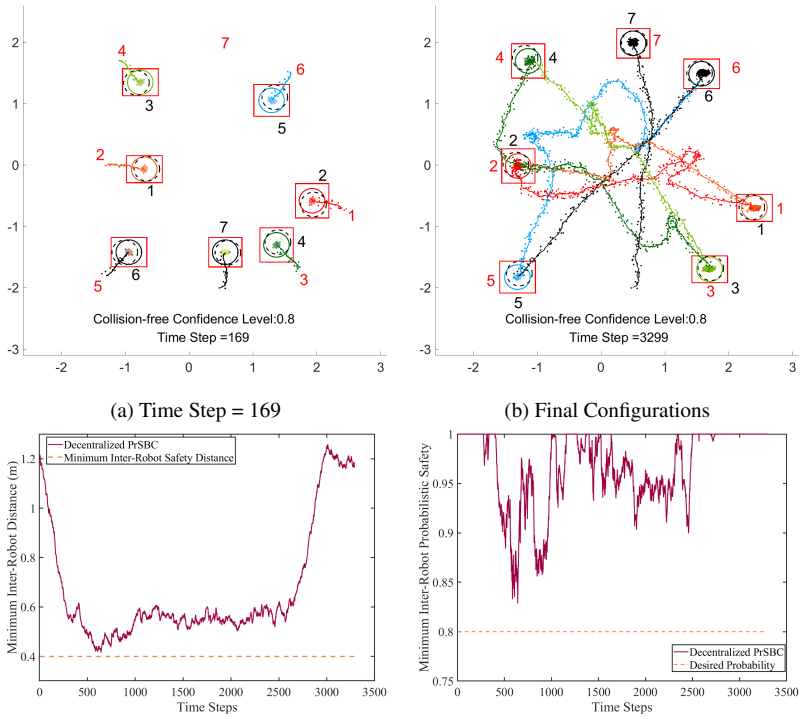


(g) Time Step = 2015 (SBC)



(h) Minimum inter-robot probabilistic safety

Figure 3.1: Simulation example of 6 robots swapping positions while maintain the collision-free confidence level $\sigma = 0.9$. Each labeled robot is covered with a red bounded error box implying the bounded real-time measurement uncertainty. The dashed black circle on each robot represents the real-time measurement of the robot and solid circle with the robot color as the ground-truth robot position surrounded by the safety radius. Labels in red are the final goal positions for the robots. Robot trajectories are covered by points in the same color from all past noisy measurements. (a)-(c) and (e)-(g) are results from our proposed PrSBC and SBC [29] respectively, with numerical results shown in (d) and (h).



(a) Time Step = 169

(b) Final Configurations

(c) Minimum true inter-distance

(d) Minimum probabilistic safety

Figure 3.2: Decentralized PrSBC with 7 robots. Robots 6 and 7 marked in black serve as passive moving obstacles without interaction to other robots.

iv) an experiment in a near-realistic simulation environment [103] with 11 simulated drones driven by Unity physical engine to demonstrate collision avoidance performance.

Simulation Example: Fig. 3.1 demonstrates the first set of simulations performed on a team of $N = 6$ mobile robots constrained by the unicycle dynamics using our PrSBC from Eq. (3.26) and the comparing deterministic SBC from [29], with both in centralized setting. We employ nonlinear inversion method [99] to map the desired velocity to the unicycle dynamics of mobile robots without compromising the safety guarantee. All of the robots employ the gradient based controller $\mathbf{u}_i^* = -K_p(\mathbf{x}_i - \mathbf{x}_{i,goal})$ to swap their positions with the robot on the opposite side, e.g. robot 1 with 2, 3 with 4, and 5 with 6 shown in Fig. 3.1a. Locations indexed in red are the goal positions for the corresponding robots. The robot safety radius is set to be $R_i = 0.2$m and has bounded uniformly distributed localization error denoted by the red error box accounting for the safety radius. At each time step, each robot only has access to the noisy measurement marked by dashed black circle covering

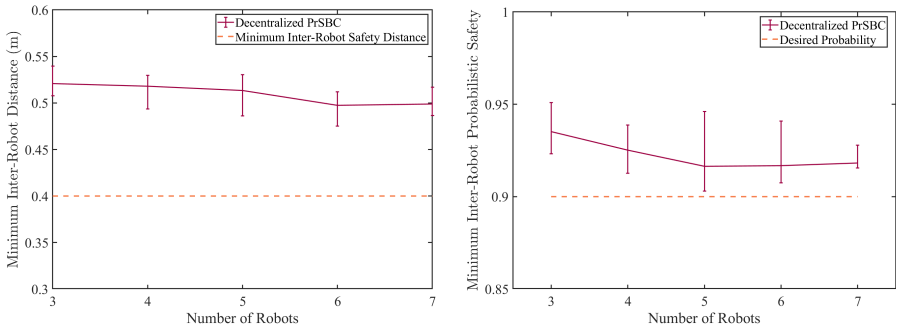
each robot. Maximum velocity limit is 0.1m/sec for the robots and robots motion is disturbed by randomly generated bounded noise with magnitude up to 0.07m/sec. The inter-robot collision-free confidence level σ set to be 0.9.

As the SBC [29] is designed for a deterministic system, here it takes the noisy measurement of the robots directly as the robot states to compose the SBC for collision avoidance controller. We observe from Fig. 3.1f that collisions occur (robot 1 and 5) due to uncertainty in measured robot states as well as the motion disturbances. While with our PrSBC controller in Eq. (3.26), robots safely navigate through the work space (Fig. 3.1d) (but not too conservatively as it still allows interaction between bounding error box shown in Fig. 3.1b for probabilistic safety). In particular, results in Fig. 3.1h indicates our PrSBC method successfully ensures the satisfying probabilistic safety ($\sigma = 0.9$). This is computed by the minimum ratio between non-overlapping area and the whole area within each robot's bounding error box shown in red.

Scenario with Dynamic Obstacles: To account for dynamic obstacles, we add robot 7 to the previous scenario and make robot 6 and 7 serve as the non-cooperating passive moving obstacles. Fig. 3.2 highlights our observations from this experiment. We assume robots can identify them as obstacles instead of cooperating robots. With the same set-up except for the two obstacles, we demonstrate the performance of our controller based on decentralized PrSBC in Eq. (3.29) and set the inter-robot, robot-obstacle collision-free confidence $\sigma = \sigma_o = 0.8$ to encourage more flexible motion. In the decentralized settings, robots are set to assume equal responsibility in collision avoidance, i.e. $p_{ij} = p_{ji} = 0.5$ in Eq. (3.28) for each robot, and thus no communication is needed between robots. Results in Fig. 3.2c and 3.2d indicate the inter-robots and robot-obstacle are collision free and with a satisfying probabilistic safety close to $\sigma = 0.8$ (thus not overly conservative). From Fig. 3.2b it is noted that robot 5 with light blue trajectory took a large detour before reaching the goal position. This is caused by the non-cooperating obstacle robot 6 and 7 in the way, where the PrSBC for obstacles Eq. (3.25) forces the robot 5 to obey the more restrictive constraints to adapt to the momentum in order to guarantee the satisfying probabilistic collision avoidance performance.

Quantitative Results: We performed 50 random trials with different number of robots under a required confidence $\sigma = 0.9$ to validate the effectiveness of our decentralized PrSBC controller in presence of random measurement and motion noise. Fig. 3.3a and 3.3b shows that the robots are always safe and satisfy the probabilistic safety guarantee using PrSBC.

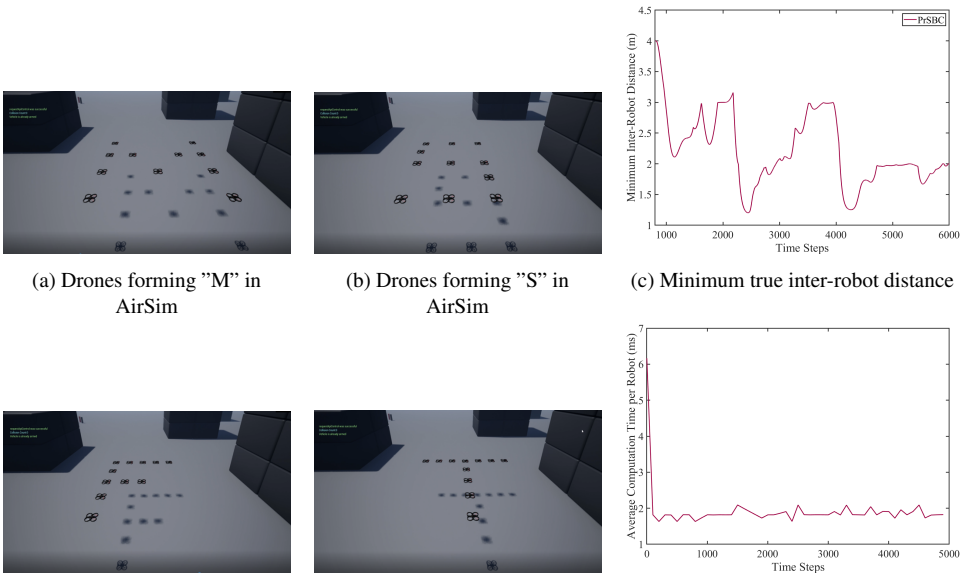
Experimental Results: Finally, as shown in Fig. 3.4, we carried out experiments with 11 simulated drones in AirSim [103], an open-source near-realistic simulation environment. The primary task for the drones is to sequentially form the letters of M-S-F-T while avoiding collisions with each other with the minimum probability



(a) Minimum true inter-distance

(b) Minimum probabilistic safety

Figure 3.3: Quantitative results summary of PrSBC from 50 random trials.



(a) Drones forming "M" in AirSim

(b) Drones forming "S" in AirSim

(c) Minimum true inter-robot distance

(d) Drones forming "F" in AirSim

(e) Drones forming "T" in AirSim

(f) Average Computation Time per Robot

Figure 3.4: AirSim [103] experiment snapshot with 11 drones using our PrSBC for collision avoidacne.

of 0.9. Each of the drones has the pre-defined target position in the letter formation and they execute the gradient based controller to move towards it. The safety radius between pairwise drones is $1m$ and the state estimation noise is between $[-0.2m, 0.2m]$. We then employ our PrSBC controller to compute the linear velocity for each drone and feed it to the drone controller in the simulator. During the task, no collisions are observed as shown in Fig. 3.4c. The simulations are on

personal laptop with Intel Core i7-8750H CPU of 2.20 GHz. The average computation time per robot is below $2ms$ as reported in Fig. 3.4f, demonstrating the efficiency of our PrSBC in real-time computation.

3.6 Conclusions and Discussions

We presented a probabilistic approach to address chance constrained collision avoidance for a system of multiple robots in real-world settings. We address the complexities that arise due to uncertainty in perception and incompleteness in modeling the underlying dynamics of the system. The key idea is to induce probabilistic constraints via safety barriers, which are then used to minimally modify an existing controller via a constrained quadratic program. We formally define Probabilisitc Safety Barrier Certificates that guarantee forward-invariance in time continuously and also can be decomposed so as to enable de-centralized computation of the safe controllers. Note that the proposed safe control method in this chapter is step-wise and reactive as other control barrier function based methods (e.g. [28, 29]), thus could be sub-optimal in terms of optimality for the entire mission period. Although the minimized deviation from a nominal controller in context of quadratic programming could reflect the *minimal invasion* nature to the task-prescribed robot behaviors, the step-wise optimization-based safe control in Eq. (3.26) does not optimize the primary task directly over the entire trajectories of the robots. In order to derive *optimal* and *safe* multi-robot behaviors over the entire mission time, one could combine planning algorithms or Model Predictive Control (MPC) frameworks with our PrSBC constraints to search for an optimal sequence of control actions within the safe envelope of robot controllers defined by PrSBC. In Chapter 4, we will discuss the example that optimizes a particular objective function over a finite time horizon while ensuring safety and task performance.

Future work entails extensions to model-based and model-free controllers trained via Reinforcement Learning and implementation to solve real-world tasks, such as Automatic Collision Avoidance System for manned and unmanned aircraft. We plan to employ variants of CBF such as ECBF [31] to explicitly handle higher relative degree system. On the other hand, extending the expressivity of the PrSBC formulation with different forms of the safe set $h^s(\mathbf{x})$ to address other uncertainty-aware safety consideration beyond collision avoidance is also an important future direction, e.g. limiting the number of drones within a volume, and adapting to temporal safety tasks using signal temporal logic (STL) formulations.

Chapter 4

Sample-efficient Safe Learning for Control

Reinforcement Learning (RL) and continuous nonlinear control have been successfully deployed in multiple domains of complicated sequential decision-making tasks. However, given the exploration nature of the learning process and the presence of model uncertainty, it is challenging to apply them to safety-critical control tasks due to the lack of safety guarantee. In Chapter 3 we have discussed how to provide safety assurance for autonomous systems under *given* uncertainty models. When the uncertainty information is unknown beforehand or overly conservative using prior knowledge, e.g. system dynamics is only partially modelled, then one would expect robots to safely explore and interact with the environment to learn the unknown dynamics and simultaneously improve task performance, often known as *safe Reinforcement Learning* (RL) problem [32]. While combining control-theoretical approaches with learning algorithms has shown promise in safe RL applications, the sample efficiency of safe data collection process for control is not well addressed.

In this chapter, we propose a *provably* sample efficient episodic safe learning framework for online control tasks that leverages safe exploration and exploitation in an unknown, nonlinear dynamical system. In particular, the framework 1) extends control barrier functions (CBF) in a stochastic setting to achieve provable safety under uncertainty during model learning and 2) integrates an optimism-based exploration strategy to efficiently guide the safe exploration process with learned dynamics for *near optimal* control performance. We provide formal analysis on the episodic regret bound against the optimal controller and probabilistic safety with theoretical guarantees.

4.1 Safe Learning for Control Problem

Consider the following partially *unknown* discrete-time control-affine system dynamics with state $x \in \mathcal{X} \subset \mathbb{R}^n$ and control input $u \in \mathcal{U} \subset \mathbb{R}^m$ for a discrete time index $h \in \mathbb{N}$

$$x_{h+1} = \hat{f}(x_h, u_h) + d(x_h, u_h) + \varepsilon_h, \quad \varepsilon_h \sim \mathcal{N}(0, \Sigma_\sigma) \tag{4.1}$$

where $\hat{f} : \mathcal{X} \times \mathcal{U} \mapsto \mathbb{R}^n$ is the known nominal discrete dynamics affine in the control input as $\hat{f}(x_h, u_h) = \hat{F}(x_h) + \hat{G}(x_h)u_h$. We assume $\hat{F} : \mathbb{R}^n \mapsto \mathbb{R}^n$, $\hat{G} : \mathbb{R}^n \mapsto \mathbb{R}^{n\times m}$ are locally Lipschitz continuous and the relative degrees of the nominal model and the actual system are the same, which are common assumptions as in [11, 39]. $d : \mathcal{X} \times \mathcal{U} \mapsto \mathbb{R}^n$ denotes the unmodelled part of the system dynamics which is unknown, and ε_h is i.i.d noise sampled from a known Multivariate Gaussian distribution with the covariance matrix $\Sigma_\sigma = \text{diag}(\sigma_1^2, \ldots, \sigma_n^2)$, i.e., $\sigma_1, \ldots, \sigma_n$ are known to the learner. For notation simplicity, we denote the stochastic state transition as $P(\cdot|x, u)$.

The unmodelled part $d(x, u)$ could represent state (action)-dependent external motion disturbances [38] or system model error due to parameter mismatch [11, 12]. With the partially unknown system dynamics, in order to control the system to achieve some tasks and also satisfy some safety constraints, we need data-driven approaches to *safely* learn the unknown part of the dynamical system $d(x, u)$ by collecting data as the system executes the task. In particular, we assume that $d(x, u)$ is modelled by the following nonlinear model, $d(x, u) := W^\star\phi(x, u)$ where $\phi : \mathcal{X} \times \mathcal{U} \mapsto \mathbb{R}^r$ is a known nonlinear feature mapping and the linear mapping $W^\star \in \mathbb{R}^{n\times r}$ is the unknown system parameters that need to be learned. Such model has been studied in [37] for unconstrained online control and in [104] for pure system identification. The model is flexible enough to describe the classic linear dynamical systems, nonlinear dynamical systems such as high order polynomials, and piece-wise linear systems [104].

The control task is described by a cost function. Given an immediate cost function $c : \mathcal{X} \times \mathcal{U} \mapsto \mathbb{R}^+$, the primary task-prescribed objective can be defined as

$$\min_{\pi\in\Pi} J^\pi(x_0; c, W^\star) = \mathbb{E}\Big[\sum_{h=0}^{H-1} c(x_h, u_h)|\pi, x_0, W^\star\Big] \tag{4.2}$$

where $\mathbf{x}_0 \in \mathcal{X}$ is a given starting state and Π is some set of pre-defined feasible controllers. Each controller (or a policy) is a mapping $\pi \in \Pi : \mathcal{X} \mapsto \mathcal{U}$. We denote $J^\pi(x; c, W)$ as the expected total cost of a policy π under cost function c, initial state x_0, and the dynamical system in Eq. (4.1) whose $d(x, u)$ is parameterized by W. To achieve optimal task performance, we need to learn the unmodelled

$d(x, u)$ by taking samples to approximate the true linear mapping $W^\star$. On the other hand, simply optimizing the cost function is not enough for safety critical application. Below we consider a specific formulation that uses Control Barrier Functions (CBF) [28] to enforce safety constraints during learning.

4.2 Discrete-time Control Barrier Functions For Gaussian Dynamical Systems

To ensure safe control and learning, we introduce CBFs in this section that are specialized to discrete-time Gaussian stochastic dynamics and prove the forward invariant safety guarantee in a stochastic setting. Note that most existing continuous or discrete-time CBFs are defined only for deterministic system, e.g. [11, 12, 28, 38, 39, 105]. For stochastic system, we introduce a new CBF definition and show that it satisfies forward invariance *with high probability* where the high probability is corresponding to the system Gaussian noise (Proposition 8).

Consider a stochastic Gaussian discrete-time dynamics described in Eq. (4.1). A desired safety set $x \in \mathcal{S} \subset \mathcal{X}$ can be denoted by the following safety function $h^s : \mathbb{R}^n \mapsto \mathbb{R}$

$$\mathcal{S} = \{x \in \mathbb{R}^n : h^s(x) \geq 0\} \tag{4.3}$$

Formally, a safety condition is forward invariant if $x_{h=0} \in \mathcal{S}$ implies $x_h \in \mathcal{S}$ for all $h > 0$ with some designed controller $u \in \mathcal{U}$. Control barrier functions (CBF) [28] are often used to derive such designed controllers that enforce the forward invariance of a set of the system state space. Due to the stochasticity of the dynamics, we need to take the noise into consideration and thus we define a stochastic discrete-time Control Barrier Function $h^s(\cdot)$ if the following condition holds.

Definition 7. *[Discrete-time Control Barrier Function under Known Gaussian Dynamics] Assume $h^s(\cdot)$ is L-Lipschitz continuous when $x \in \mathcal{X}$ is bounded. Given $\delta \in (0, 1)$ and horizon H, let $\mathcal{S}$ be the 0-superlevel set of $h^s : \mathbb{R}^n \to \mathbb{R}$ which is a continuously differentiable function. We call $h^s(\cdot)$ a stochastic discrete-time control barrier function (CBF) for dynamical system Eq. (4.1) if there exists a $\eta \in (0, 1)$, such that for all time steps $h = 0, \ldots, H-1$, given any $x \in \mathcal{S}$:*

$$\sup_{u \in \mathcal{U}} \left[h^s\left(\hat{f}(x, u) + d(x, u)\right) - L\bar{\sigma}\sqrt{2n \ln\left(\frac{Hn}{\delta}\right)} - h^s(x) \right] \geq -\eta h^s(x) \tag{4.4}$$

where $\bar{\sigma} = \max\{\sigma_1, \ldots, \sigma_n\}$. Note that different from conventional CBF [12, 28] that is defined with respect to deterministic transition, the above definition takes the stochasticity into consideration. Also note that when $\bar{\sigma} \to 0$, i.e., the Gaussian

dynamical system in Eq. (4.1) becomes a near deterministic system, then the above definition converges to the usual discrete-time CBF definition [12, 30]. We now show that under Definition 7, the state $\mathcal{S}$ is forward invariant with probability at least $1-\delta$.

Proposition 8 (Forward Invariant with High Probability)**.** *Consider a control barrier function $h^s(\cdot)$ in Definition 7. Given $x_0 \in \mathcal{S}$, consider any policy $\pi : \mathcal{X} \to \mathcal{U}$ such that at any state x, this policy outputs an action $u = \pi(x)$ that satisfies the constraint Eq. (4.4). Then executing π to generate a trajectory starting at x_0: $\tau = \{x_0, u_0, \ldots, x_{H-1}, u_{H-1}\}$, with probability at least $1-\delta$ we have $h^s(x_h) \geq 0$ for all $h \in [H]$, i.e., all states on the trajectory belong to the safe set $\mathcal{S}$.*

Proof. For notation simplicity, let us denote $f(x,u) := \hat{f}(x,u) + d(x,u)$. First consider the event: $\exists h \in [H], i \in [n]$ such that $\epsilon_t[i] \geq p$. Via union bound and the normal distribution's property, we have:

$$\mathbb{P}\left(\exists h \in [H], i \in [n], \text{ s.t., } \epsilon_t[i] \geq p\right) \leq Hn \exp(-p^2/(2\bar{\sigma}^2)).$$

Let us set the failure probability to be δ, i.e., $Hn\exp(-p^2/(2\bar{\sigma}^2)) = \delta$. Solve for p and we indeed have $p := \bar{\sigma}\sqrt{2\ln\left(\frac{Hn}{\delta}\right)}$. Below we conditioned the event that for all $h \in [H]$ and $i \in [n]$, we have $|\epsilon_h[i]| \leq p$. Note that this event happens with probability at least $1-\delta$.

Due to the Lipschitz assumption with a Lipshcitz constant L and the assumption that $h^s(\cdot)$ is differentiable, we have that for the system $x_{h+1} = f(x_h, u_h) + \epsilon_h$ at any time $h \in [H]$,

$$h^s(x_{h+1}) = h^s(f(x_h,u_h)) + \nabla h^s(\xi)^\top \epsilon_h \geq h^s(f(x_h,u_h)) - L\bar{\sigma}\sqrt{2n\ln\left(\frac{Hn}{\delta}\right)}$$

Since the control policy satisfies the control barrier function constraint in Eq. (4.4), we must have:

$$h^s(x_{h+1}) - h^s(x_h) \geq \left[h^s(f(x_h,u_h)) - L\bar{\sigma}\sqrt{2n\ln\left(\frac{Hn}{\delta}\right)} - h^s(x_h)\right] \geq -\eta h^s(x_h)$$

Namely, we have:

$$h^s(x_{h+1}) \geq (1-\eta)h^s(x_h) \geq (1-\eta)^{h+1}h^s(x_0) \geq 0,$$

under the condition that $h^s(x_0) \geq 0$. The above argument holds for all $h \in [H]$ which thus concludes the proof. □

Remark 3. *In general for nonlinear function $h^s(\cdot)$ and nonlinear dynamical system, the constraint in Eq. (4.4) is nonlinear with respect to the control u. Similar to [12], one special case have been considered where $h^s(\cdot)$ is linear with respect to x (affine barrier functions), and both $\hat{f}$ and d are affine control functions in the form of $g_1(x) + g_2(x)u$, though our method could apply to more general barrier functions. In this case, the constraint in Eq. (4.4) becomes linear with respect to u.*

Proof. Here we discuss how to derive the control constraints with nonlinear control barrier function $h^s(\cdot)$ from Eq. (4.8) that fulfills Eq. (4.4) (and hence fulfills Proposition 8). Recall the constraint Eq. (4.8) as follows (we have $u_h = \pi(x_h)$).

$$h^s\left(\hat{f}(x_h, u_h) + d(x_h, u_h)\right) - L\bar{\sigma}\sqrt{2n\ln\left(\frac{Hn}{\delta}\right)} - h^s(x_h) \geq -\eta h^s(x_h) \quad (4.5)$$

Given that both the known nominal discrete dynamics $\hat{f}$ and the unknown part d are affine in control as $\hat{f}(x_h, u_h) = \hat{F}(x_h) + \hat{G}(x_h)u_h$ and $d(x_h, u_h) = g_1(x_h) + g_2(x_h)u_h$, where $\hat{F}, \hat{G}, g_1, g_2$ are assumed locally Lipschitz continuous. Then with the continuously differentiable function $h^s(\cdot)$, we have $h^s\left(\hat{f}(x_h, u_h) + d(x_h, u_h)\right) - h^s(x_h) = L^{\Delta}_{\hat{F}+g_1} h^s(x_h) + L^{\Delta}_{\hat{G}+g_2} h^s(x_h)u_h$ and hence Eq. (4.5) can be re-written as

$$L^{\Delta}_{\hat{F}+g_1} h^s(x_h) + L^{\Delta}_{\hat{G}+g_2} h^s(x_h)u_h \geq -\eta h^s(x_h) + L\bar{\sigma}\sqrt{2n\ln\left(\frac{Hn}{\delta}\right)} \quad (4.6)$$

where $L^{\Delta}_{\hat{F}+g_1} h^s(x_h)$ and $L^{\Delta}_{\hat{G}+g_2} h^s(x_h)$ are discrete-time Lie-derivatives of $h^s(x_h)$ obtained through Taylor's theorem along $\hat{F}(x_h) + g_1(x_h)$ and $\hat{G}(x_h) + g_2(x_h)$ respectively. To that end, the condition in Eq. (4.4) and Eq. (4.8) hold by enforcing the linear control constraint Eq. (4.6) on u_h. Thus, we conclude the proof. □

4.3 Learning Objective

If we had knew the unmodelled dynamics $d(\cdot)$, i.e., the whole stochastic Gaussian dynamical system in Eq. (4.1) is known, then the safe nonlinear control problem can be modeled as follows:

$$\min_{\pi \in \Pi} J^{\pi}(x_0; c), \quad (4.7)$$

$$s.t. \quad h^s\left(\hat{f}(x, \pi(x)) + d(x, \pi(x))\right) - L\bar{\sigma}\sqrt{2n\ln\left(\frac{Hn}{\delta}\right)} - h^s(x) \geq -\eta h^s(x),$$
$$\forall x \in \mathcal{X} \quad (4.8)$$

Assume the above constrained optimization problem is feasible, and denote $\pi^\star$ as the optimal policy. Then, via Proposition 8, we know that with probability at least $1-\delta$, $\pi^\star$ generates a trajectory whose states are all in the safe set $\mathcal{S}$. However, as the unmodelled part $d(x,u)$ is initially unknown, we cannot directly solve the above constrained optimization program using standard RL or MPC approaches. Instead we need to learn d (more specifically, the unknown linear mapping $W^\star \in \mathbb{R}^{n\times r}$) online using data-driven approach for safety guarantee as well as cost minimization.

In our episodic finite horizon learning framework, we start with some initialization $\overline{W}_0$ which is used to parameterize $d_0(x,u) := \overline{W}_0\phi(x,u)$ (we will discuss conditions on $\overline{W}_0$ in Section 4.4 that can ensure safety during the entire learning process). At every episode t, the learner will propose a policy $\pi_t \in \Pi$ (probably based on the current guess $d_t(x,u)$ with $\overline{W}_t$), execute π_t in the real system to generate one trajectory $\{x_0^t, u_0^t, \ldots, x_{H-1}^t, u_{H-1}^t\}$ for H time steps; the learner then incrementally updates model parameter to $\overline{W}_{t+1}$ using observations from all of the past trajectories, and move to the next episode $t+1$ starting from the same initial state $x \leftarrow x_0$. The ideal goal of the learner is to ensure that π_t is safe (with high probability) in terms of satisfying CBF constraint Eq. (4.4), and also optimize the cost function over episodes:

$$\text{Regret}_T := \sum_{t=0}^{T-1}\sum_{h=0}^{H-1} c(x_h^t, u_h^t) - \sum_{t=0}^{T-1} J^{\pi^\star}(x_0; c) = o(T) \tag{4.9}$$

Namely, comparing to the best policy $\pi^\star$ (i.e., the optimal solution of the constrained optimization program in Eq. (4.7) if assuming perfect model information), the cumulative regret grows sublinearly with respect to the number of episodes T. This goal implies that when $T \to \infty$ in a long run, the average episodic cost incurred by the learner is the same as the episodic cost incurred by the best policy $\pi^\star$. In other words, the policy generated with the learned dynamics model over time performs as good as the optimal policy generated with the ground-truth dynamics Eq. (4.1) in a long run.

To that end, the goal is to minimize the cumulative regret in Eq. (4.9) subject to safety constraint in Eq. (4.8) at all times in each episode. Next, we will discuss how to enforce such safety constraint with d learned online and provide the episodic safe learning algorithm to achieve bounded regret in Eq. (4.9) with rigorous analysis.

4.4 Calibrated Model and Approximate Safety Guarantee

Due to the unknown d, our initial guess d_0 using $\overline{W}_0$ estimated from some pre-collected data could be arbitrarily bad, and hence it is impossible to achieve safe

learning while simultaneously optimizing the regret without further assumptions. Note that ensuring small regret as Eq. (4.9) requires us to perform optimism-based exploration (e.g., multi-armed bandits and PAC/no-regret learning in RL), while ensuring safety under model uncertainty requires us to be conservative.* Intuitively, exploration via optimism and safety via conservativeness are contradict to each other. While existing safe learning work often made the required standard assumption that the prior model of d could yield initial safe policy to start the exploration process (e.g. [8, 11, 12, 38]), it is non-trivial to characterize the initial uncertainty region over d to enforce safety guarantee for an optimism-based exploration. Hence it is challenging to derive a safe exploration strategy with small regret for efficient sampling and high control performance. In this work, we will first give the assumption of a calibrated model $\overline{W}_0$ that derives a reasonable initial estimate of d, as typically assumed in [8, 12, 38, 39], and also we characterize the data pre-collection to derive such $\overline{W}_0$ that fulfills this assumption for safe optimism-based exploration discussed in Section 4.5.

Given N triples $(x_i, u_i, x_i')_{i=1}^N$ with $x' \sim P(\cdot|x, u)$ as the set of pre-collected initial training data , we can compute $\overline{W}_0$— the initialization parameters of $d(x, u)$ via ridge linear regression under known feature mapping $\phi : \mathcal{X} \times \mathcal{U} \mapsto \mathbb{R}^r$:

$$\overline{W}_0 = \arg\min_W \sum_{i=1}^{N} \left\| W\phi(x_i, u_i) - (x_i' - \hat{f}(x_i, u_i)) \right\|_2^2 + \lambda \|W\|_F^2 \tag{4.10}$$

where λ is a regularizer parameter and $\|W\|_F$ is the Frobenius norm of the model parameter matrix $W \in \mathbb{R}^{n \times r}$. Denote the initial empirical regularized covariance matrix as

$$V_0 = \sum_{i=1}^{N} \phi(x_i, u_i)\phi(x_i, u_i)^\top + \lambda I \tag{4.11}$$

To initialize safe learning, the following assumption shows that by using polynomially number of samples drawn from an appropriate distribution μ as the initial training data, we will have $d_0(x, u) = \overline{W}_0\phi(x, u)$ as a reasonable good estimate of $d(x, u) = W^\star\phi(x, u)$ for all $x, u \in \mathcal{X} \times \mathcal{U}$ (note that however we cannot guarantee $\overline{W}_0$ will be close to $W^\star$ in terms of ℓ_2 norm).

Assumption 9. *(Calibrated model from pre-collected data) Fix a pair (ϵ, δ) with $\epsilon, \delta \in (0, 1)$. Draw N triples $\{x_i, u_i, x_i'\}_{i=1}^N$ from some distribution μ with $x_i, u_i \sim \mu, x_i' \sim P(\cdot|x_i, u_i)$, and set $\overline{W}_0 = \sum_{i=1}^N (x_i' - \hat{f}(x_i, u_i))\phi(x_i, u_i)^\top V_0^{-1}$ with*

*Achieving safety via conservative behavior is easy: the robot can stay still and it will be safe during this episode, e.g. avoiding static obstacles for a mobile robot. However this does not optimize the task's cost function. From simple multi-arm bandit problems, it is not hard to show an example where being conservative rather than optimistic fails to optimize the reward.

$V_0 = \sum_{i=1}^N \phi(x_i, u_i)\phi(x_i, u_i)^\top + \lambda I$. *Then with probability at least* $1-\delta$, *we have:*

$$\forall x, u \in \mathcal{X} \times \mathcal{U}, \quad \left\|(\overline{W}_0 - W^\star)\phi(x,u)\right\|_2 = O(\epsilon),$$

with polynomially number of samples, i.e., N *scaling polynomially with respect to the relevant parameters:*

$$N = \mathcal{O}\Big(\frac{r\|W^\star\|^2\lambda + r\bar{\sigma}^2 n + \ln(1/\delta) + r^2\bar{\sigma}^2}{\epsilon^2} + \frac{\|W^\star\|^2 r^2 \ln(r/\delta)}{\epsilon^4}\Big) \tag{4.12}$$

After deriving $\overline{W}_0, V_0$ from the initial data $(x_i, u_i, x_i')_{i=1}^N$, we can build the initial confidence ball describing the uncertain region of $W^\star$ as follows:

$$\text{Ball}_0 = \left\{W : \left\|(W - \overline{W}_0)V_0\right\|_2 \le \beta, \quad \|W\|_2 \le \|W^\star\|_2\right\} \tag{4.13}$$

where β is the confidence radius as $\beta := \sqrt{\lambda}\|W^\star\|_2 + \bar{\sigma}\sqrt{8n\ln(5) + 8d\ln(1+N/\lambda) + 8\ln(1/\delta)}$.

Proof. Here we discuss how to find the distribution μ using John's ellipsoid [106] for sampling initial training data and deriving $\overline{W}_0$ that fulfills Assumption 9.

Denote $\Phi \in \mathbb{R}^{r\times N}$ where each column of Φ corresponds to the feature vector $\phi(x,u)$ for $(x,u) \in \mathcal{X}\times\mathcal{U}$. Assume $\text{span}(\Phi) = r$. Via John's theorem, denote $\mathcal{B} \subset \mathcal{X}\times\mathcal{U}$ as the core set of John's ellipsoid, and μ as the corresponding sampling distribution with support on $\mathcal{B}$ defined by $\mu = \arg\max_{\rho\in\Delta(\mathcal{X}\times\mathcal{U})} \ln\det\left(\mathbb{E}_{x,u\sim\rho}\phi(x,u)\phi(x,u)^\top\right)$ from John's ellipsoid. Then draw N triples $\mathcal{D} = \{x_i, u_i, x_i'\}_{i=1}^N$ as pre-collected offline dataset with $x_i, u_i \sim \mu, x_i' \sim P(\cdot|x_i,u_i)$, and compute the initialization $\overline{W}_0 = \sum_{i=1}^N (x_i' - \hat{f}(x_i,u_i))\phi(x_i,u_i)^\top V_0^{-1}$ with $V_0 = \sum_{i=1}^N \phi(x_i,u_i)\phi(x_i,u_i)^\top + \lambda I$.

First note that we can compute the exact difference between the least square solution $\overline{W}_0$ and $W^\star$:

$$\overline{W}_0 - W^\star = -\lambda W^\star (V_0)^{-1} + \sum_{i=1}^N \epsilon_i \phi(x_i,u_i)^\top V_0^{-1}.$$

Continue, we have

$$\begin{aligned}
\left\|(\overline{W}_0 - W^\star)V_0^{1/2}\right\|_2 &\le \left\|\lambda W^\star V_0^{-1/2}\right\|_2 + \left\|\sum_{i=1}^N \epsilon_i\phi(x_i,u_i)^\top V_0^{-1/2}\right\|_2 \\
&\le \sqrt{\lambda}\|W^\star\|_2 + \bar{\sigma}\sqrt{8n\ln(5) + 8\ln\left(\det(1+V_0/\lambda)\right) + 8\ln(1/\delta)} \\
&\le \underbrace{\sqrt{\lambda}\|W^\star\|_2 + \bar{\sigma}\sqrt{8n\ln(5) + 8r\ln(1+N/\lambda) + 8\ln(1/\delta)}}_{:=\beta}
\end{aligned}$$

Denote $\Sigma = \mathbb{E}_{x,u\sim\mu}\phi(x,u)\phi(x,u)^\top$. Via matrix Bernstein's inequality, we get that with probability at least $1-\delta$, for any x with $\|x\|_2 \leq 1$,

$$\left| x^\top \left(\sum_{i=1}^{N} \phi(x_i,u_i)\phi(x_i,u_i)^\top/N - \Sigma \right) x \right| \leq \frac{2\ln(8r/\delta)}{3N} + \sqrt{\frac{2\ln(8r/\delta)}{N}} := \varepsilon.$$

Thus we will have that for any x with $\|x\|_2 \leq 1$ and with the standard assumption of bounded norm $\|W^\star\|_2 \leq W$:

$$x^\top(\overline{W}_0 - W^\star)V_0(\overline{W}_0 - W^\star)^\top x \geq x^\top(\overline{W}_0 - W^\star)(\Sigma N + \lambda)(\overline{W}_0 - W^\star)^\top x - 2\varepsilon NW,$$

which means that:

$$\begin{aligned}\left\|(\overline{W}_0 - W^\star)(\Sigma+\lambda/N)^{1/2}\right\|_2^2 \leq& \beta^2/N + 2W\varepsilon \\ \leq& \frac{\lambda W^2}{N} + \frac{\bar{\sigma}^2(n + r\ln(1+N/\lambda+\ln(1/\delta)))}{N} + \frac{2W\sqrt{\ln(8r/\delta)}}{\sqrt{N}}\end{aligned}$$

For any x, u, we have:

$$\left|(\overline{W}_0 - W^\star)\phi(x,u)\right|^2 \leq \left\|(\overline{W}_0 - W^\star)(\Sigma+\lambda/N)^{1/2}\right\|_2^2 \left\|(\Sigma+\lambda/N)^{-1/2}\phi(x,u)\right\|_2^2$$

Note that for any x, we have:

$$x^\top\Sigma^{-1}x \geq x^\top(\Sigma+\lambda/N)^{-1}x.$$

Using the John's theorem, we get that:

$$\phi(x,u)^\top(\Sigma+\lambda/N)^{-1}\phi(x,u) \leq \phi(x,u)^\top\Sigma^{-1}\phi(x,u) \leq r$$

Hence, we have:

$$\begin{aligned}\left|(\overline{W}_0 - W^\star)\phi(x,u)\right| \leq& \sqrt{\left(\frac{\beta^2}{N} + 2W\varepsilon\right)r} \\ \leq& \sqrt{\frac{r\lambda W^2}{N}} + \sqrt{\frac{r\bar{\sigma}^2(n + r\ln(1+N/\lambda+\ln(1/\delta)))}{N}} + \sqrt{\frac{2Wr\sqrt{\ln(8r/\delta)}}{\sqrt{N}}}\end{aligned}$$

Now setting $N = \mathcal{O}\Big(\frac{r\|W^\star\|^2\lambda + r\bar{\sigma}^2 n + \ln(1/\delta) + r^2\bar{\sigma}^2}{\epsilon^2} + \frac{\|W^\star\|^2 r^2\ln(r/\delta)}{\epsilon^4}\Big)$, we ensure that:

$$\left|(\overline{W}_0 - W^\star)\phi(x,u)\right| \leq O(\epsilon).$$

This concludes the proof. □

Then in the following we can show that for any $\widetilde{W} \in \text{Ball}_0$, its prediction $\widetilde{d}(x,u) = \widetilde{W}\phi(x,u)$ for any given x, u is close to the true prediction $d(x,u) = W^\star\phi(x,u)$ from $W^\star$.

Lemma 10. *Consider the setting in Assumption 9 and Ball$_0$ defined in Eq. (4.13). For all $\widetilde{W} \in$ Ball$_0$, we have that with probability at least $1-\delta$:*

$$\forall x, u \in \mathcal{X} \times \mathcal{U} : \ \left|\left(\widetilde{W} - W^\star\right)\phi(x,u)\right| \leq \mathcal{O}\left(\epsilon\right).$$

Proof. Starting from triangle inequality, we get:

$$\begin{aligned}
\left|(\widetilde{W} - W^\star)\phi(x,u)\right| &\leq \left|(\widetilde{W} - \overline{W}_0)\phi(x,u)\right| + \left|(\overline{W}_0 - W^\star)\phi(x,u)\right| \\
&\leq \left\|(\widetilde{W} - \overline{W}_0)(\Sigma + \lambda/N)^{1/2}\right\|_2 \left\|(\Sigma + \lambda/N)^{-1/2}\phi(x,u)\right\|_2 \\
&\quad + \left\|(\overline{W}_0 - W^\star)(\Sigma + \lambda/N)^{1/2}\right\|_2 \left\|(\Sigma + \lambda/N)^{-1/2}\phi(x,u)\right\|_2 \\
&\leq \left\|(\widetilde{W} - \overline{W}_0)(\Sigma + \lambda/N)^{1/2}\right\|_2 \sqrt{r} + \left\|(\overline{W}_0 - W^\star)(\Sigma + \lambda/N)^{1/2}\right\|_2 \sqrt{r}
\end{aligned}$$

We also know that for any two W_1 and W_0 with $\|W_i\|_2 \leq W$ with $i \in \{1,2\}$, we have:

$$x^\top (W_1 - W_2) V_0 (W_1 - W_2)^\top x \geq x^\top (W_1 - W_2)(\Sigma N + \lambda)(W_1 - W_2)^\top x - 2\varepsilon N W,$$

which means that:

$$\begin{aligned}
\left\|(\overline{W}_0 - \widetilde{W})(\Sigma + \lambda/N)^{1/2}\right\|_2^2 &\leq \beta^2/N + 2W\varepsilon, \\
\left\|(\overline{W}_0 - W^\star)(\Sigma + \lambda/N)^{1/2}\right\|_2 &\leq \beta^2/N + 2W\varepsilon.
\end{aligned}$$

This implies that:

$$\left|(\widetilde{W} - W^\star)\phi(x,u)\right| \leq 2\sqrt{r}\sqrt{\beta^2/N + 2W\varepsilon}.$$

Now recall the setup of N, β, and ε in Assumption 9 and its proof, we conclude the proof. □

The calibrated initial model from pre-collected data $(x_i, u_i, x_i')_{i=1}^N$ and confidence region Ball$_0$ ensure that when we control our dynamical system using CBF with any model $\widetilde{W} \in \text{Ball}_0$, we can ensure safety update to $\mathcal{O}(\epsilon)$.

Theorem 11 (Policy for Approximate Safety Guarantee with Learned Dynamics)**.** *Suppose the conditions in Assumption 9 hold. Consider any $\widetilde{W} \in$ Ball$_0$, and define any policy $\pi_s : \mathcal{X} \mapsto \mathcal{U}$ that satisfies the CBF constraint parameterized by $\widetilde{W}$, i.e.,*

$$\forall x \in \mathcal{X} : \pi_s(x) \in \mathcal{U}_x := \Big\{ u : h^s\left(\hat{f}(x,u) + \widetilde{W}\phi(x,u)\right) - L\bar{\sigma}\sqrt{2n\ln\left(\frac{Hn}{\delta}\right)} \geq (1-\eta)h^s(x) \Big\} \quad (4.14)$$

Then with probability at least $1-\delta$, starting at any safe initial state $h^s(x_0) \geq 0$, π_s generates a safe trajectory $\{x_0, u_0, \ldots, x_{H-1}, u_{H-1}\}$, such that for all time steps $h \in [H]$, $h^s(x_h) \geq -\mathcal{O}(\frac{L\epsilon}{\eta})$, where L is the Lipschitz constant of $h^s(\cdot)$ under bounded $x \in \mathcal{X}$.

Proof. Start from Lemma 10, we know that for any $\widetilde{W} \in$ Ball$_0$, we have:

$$\left|\left(\widetilde{W} - W^\star\right)\phi(x,u)\right| \leq \mathcal{O}(\epsilon), \forall x, u \in \mathcal{X} \times \mathcal{U}.$$

From Eq. (4.14) the policy selects action u_h for all time steps $h \in [H]$ such that:

$$h^s(\hat{f}(x_h,u_h) + \widetilde{W}\phi(x_h,u_h)) - L\bar{\sigma}\sqrt{2n\ln\left(\frac{Hn}{\delta}\right)} \geq (1-\eta)h^s(x_h)$$

This means that for $W^\star$, we have:

$$\begin{aligned}
h^s(x_{h+1}) &= h^s(\hat{f}(x_h,u_h) + W^\star\phi(x_h,u_h) + \epsilon_h) \\
&\geq h^s(\hat{f}(x_h,u_h) + \widetilde{W}\phi(x_h,u_h)) - L\left\|(\widetilde{W} - W^\star)\phi(x_h,u_h)\right\|_2 - L\|\epsilon_h\|_2 \\
&\geq (1-\eta)h^s(x_h) + L\bar{\sigma}\sqrt{2n\ln\left(\frac{Hn}{\delta}\right)} - L\epsilon - L\|\epsilon_h\|_2 \\
&\geq (1-\eta)h^s(x_h) - L\epsilon \geq (1-\eta)^2 h^s(x_{h-1}) - L\left(\epsilon + (1-\eta)\epsilon\right) \\
&\geq (1-\eta)^{h+1}h^s(x_0) - \frac{L}{\eta}\epsilon
\end{aligned}$$

Using the initial condition that $h^s(x_0) \geq 0$, we conclude the proof. □

Thus we narrow down the search region for $W^\star$ and have $W^\star \in$ Ball$_0$ with probability at least $1-\delta$. Later on, when we improve our model during iterative learning, as long as we restrict our model $\widetilde{W}$ to Ball$_0$, we ensure that any policy that satisfies the CBF constraint under $\widetilde{W}$ (Eq. 4.14) is guaranteed to be approximately

safe in the sense of Theorem 11. Next we move to iterative learning where we aim to search for a policy using an optimism-based algorithm that performs as good as the best benchmark $\pi^\star$ in the sense of minimizing regret defined in Eq. (4.9) and subject to Eq. (4.14).

4.5 Optimism-based Safe Learning for Control with Regret Analysis

To achieve no-regret performance for efficient safe learning for control, we leverage the LC^3 algorithm developed in [37] for strategic exploration. However, the original LC^3 is designed for unconstrained optimization and hence not suitable for safety-critical applications. Here we modify the policy selection step in LC^3 to take our CBF constraint Eq. (4.14) into consideration and thus ensures approximate safety (i.e., Theorem 11). Meanwhile, similar to LC^3, we also need to leverage the principle of optimism in the face of uncertainty to achieve small regret and with safety guarantee. We propose the following framework of Optimism-based Safe Learning for Control (Algorithm 1) that seeks to minimize the cumulative regret for optimal online control performance with safety guarantee.

Algorithm 1 Optimism-based Safe Learning for Control

Input: CBF h^s, cost function c, initial data $(x_i, u_i, x_i')_{i=1}^N$, initial confidence region Ball$_0$ with $\overline{W}_0, \Sigma_0$, number of training episodes T, horizon H, regularizer λ, initial state x_0

Output: a sequence of policies for $t = 0, ..., T$

1: **for** $t = 0, \ldots, T$ **do**
2: $\quad x_0^t \leftarrow x_0$
3: $\quad$ Sample $\widetilde{W}_t \sim \mathcal{N}(\overline{W}_t, \Sigma_t^{-1})$ # `Thompson Sampling for Exploration`
4: $\quad \pi_s^t \leftarrow \arg\min_{\pi \in \Pi_{\widetilde{W}}} J^\pi(x_0^t; c, \widetilde{W}_t)$ # `Safe MPC Planning`
5: $\quad$ Execute π_s^t to sample a trajectory $\tau^t := \{x_h^t, u_h^t, c_h^t, x_{h+1}^t\}_{h=0}^{H-1}$ # `Execution and Data Collection`
6: $\quad \overline{W}_{t+1}, \Sigma_{t+1} \leftarrow$ Update Ball$_{t+1}$ # `Model Update`
7: **end for**
8: Return a sequence of policies for $t = 0, ..., T$

At the beginning of each episode t, given all previous trajectories, $\tau^i = \{x_0^i, u_0^i, \ldots, x_{H-1}^i, u_{H-1}^i, x_H^i\}$ from $i = 0$ to $t-1$, we perform ridge linear re-

gression to find $\overline{W}_t$ in Line 6, i.e.,

$$\overline{W}_t = \arg\min_{W} \sum_{i=1}^{N} \Big(W\phi(x_i, u_i) - (x_i' - \hat{f}(x_i, u_i)) \Big)^2 + \sum_{i=0}^{t-1} \sum_{h=0}^{H-1} \Big(W\phi(x_h^i, u_h^i) - (x_{h+1}^i - \hat{f}(x_h^i, u_h^i)) \Big)^2 + \lambda \|W\|_F^2 \quad (4.15)$$

and we have the shape of the estimate region as

$$\Sigma_t = V_0 + \sum_{i=0}^{t-1} \sum_{h=0}^{H-1} \phi(x_h^i, u_h^i)\phi(x_h^i, u_h^i)^\top \quad (4.16)$$

where we use the data from trajectories $\{\tau^i\}_{i=0}^{t-1}$ and also the initial data $(x_i, u_i, x_i')_{i=1}^N$ for computing Ball_0. Then the confidence region of W is defined as:

$$\text{Ball}_t = \text{Ball}_0 \cap \Big\{ W : \|(W - \overline{W}_t)\Sigma_t^{1/2}\|_2 \leq \beta' \Big\}$$

and the confidence radius parameter β' is defined as:

$$\beta' := \sqrt{\lambda}\|W^\star\|_2 + \bar{\sigma}\sqrt{8n\ln(5) + 8r\ln\left(1 + (TH + N)/\lambda\right) + 8\ln(1/\delta)} \quad (4.17)$$

LC^3 [37] shows that with probability $1 - \delta$, for all t, $W^\star \in \{W : \|(W - \overline{W}_t)\Sigma_t^{1/2}\|_2 \leq \beta_t\}$:

Lemma 12. *(Confidence Ball [37, Proof of Lemma B.5]) For any $0 < \delta < 1$, it holds with probability at least $1 - \delta$ that for all t,*

$$W^\star \in \{W : \|(W - \overline{W}_t)\Sigma_t^{1/2}\|_2 \leq \beta_t\} \quad (4.18)$$

and the confidence radius parameter β_t is defined as:

$$\beta_t := \sqrt{\lambda}\|W^\star\|_2 + \bar{\sigma}\sqrt{8n\ln(5) + 8r\ln\left(1 + (tH + N)/\lambda\right) + 8\ln(1/\delta)} \quad (4.19)$$

Under proved Assumption 9 we show that with probability at least $1-\delta$, $W^\star \in \text{Ball}_0$. Hence it is not hard to see that with probability at least $1 - 2\delta$, we have $W^\star \in \text{Ball}_0 \cap \Big\{ W : \|(W - \overline{W}_t)\Sigma_t^{1/2}\|_2 \leq \beta_t \Big\} := \text{Ball}_t$ regarding the intersection of the confidence intervals:

Proposition 13. *[High Probability of Intersection of the Confidence Intervals] Given the uncertainty regions $W^\star \in \{W : \|(W - \overline{W}_t)\Sigma_t^{1/2}\|_2 \leq \beta_t\}$ (Lemma 12) and $Ball_0$ (Eq. 4.13) with the probability of $Pr(W^\star \in \{W : \|(W - \overline{W}_t)\Sigma_t^{1/2}\|_2 \leq \beta_t\}) \geq 1 - \delta$ and $Pr(W^\star \in Ball_0) \geq 1 - \delta$, then for all t we have*

$$Pr\left(W^\star \in Ball_t := Ball_0 \cap \left\{W : \|(W - \overline{W}_t)\Sigma_t^{1/2}\|_2 \leq \beta_t\right\}\right) \geq 1 - 2\delta \quad (4.20)$$

Proof. By definition,

$$\Pr\left(W^\star \notin \{W : \|(W - \overline{W}_t)\Sigma_t^{1/2}\|_2 \leq \beta_t\}\right) \leq \delta$$
$$\Pr\left(W^\star \notin \text{Ball}_0\right) \leq \delta$$

Thus, we have

$$\begin{aligned}
&\Pr\left(W^\star \in \text{Ball}_t := \text{Ball}_0 \cap \left\{W : \|(W - \overline{W}_t)\Sigma_t^{1/2}\|_2 \leq \beta_t\right\}\right) \\
=&1 - \Pr\left(W^\star \notin \{W : \|(W - \overline{W}_t)\Sigma_t^{1/2}\|_2 \leq \beta_t\} \text{ Or } W^\star \notin \text{Ball}_0\right) \\
\geq&1 - 2\delta
\end{aligned}$$

which concludes the proof. □

Hence for all episode t with the updated uncertain region Ball_t where $W^\star$ lives with high probability (Proposition 13) and any model $\widetilde{W} \in \text{Ball}_t$, we are able to construct the safe policy class Π based on the CBF constraint under $\widetilde{W}$ (Eq. 4.14), i.e., we denote $\Pi_{\widetilde{W}}$ as follows:

$$\begin{aligned}
\Pi_{\widetilde{W}} = \Bigg\{\pi_s \in \Pi : \forall x \in \mathcal{X}, \pi_s(x) \in \Big\{u : h^s\left(\hat{f}(x,u) + \widetilde{W}\phi(x,u)\right) \\
- L\bar{\sigma}\sqrt{2n\ln\left(\frac{Hn}{\delta}\right)} \geq (1-\eta)h^s(x)\Big\}\Bigg\}
\end{aligned} \quad (4.21)$$

With this now we select our model and policy optimistically at each episode t, i.e.,

$$\left(W_t, \pi^t\right) := \underset{\widetilde{W} \in \text{Ball}_t}{\arg\min} \underset{\pi \in \Pi_{\widetilde{W}}}{\arg\min} J^\pi(x_0^t; c, \widetilde{W}). \quad (4.22)$$

Similar to [37], in face of uncertainty this optimization problem could be solved by using reasonable approximation algorithms, e.g. Thompson Sampling [107] for a $\widetilde{W}$ and then computing the safe optimal policy with MPC planning oracle such as MPPI [60] under $\widetilde{W}$ subject to the safety constraint Eq. (4.21). In this way, we build upon the achievable Bayesian regret bound for the Thompson sampling and the

optimal control performance from MPC algorithms under the sampled dynamics by $\widetilde{W}$, with approximate safety guarantee.

Then given Eq. (4.22) and conditioned on the high probability event that $W^\star \in \text{Ball}_t$, and $\pi^\star \in \Pi_{W^\star}$ by definition of $\pi^\star$, we can easily show optimism in the sense that:

$$J^{\pi^t}\left(x_0^t; c, W_t\right) \leq J^{\pi^\star}(x_0; c, W^\star).$$

Below we briefly summarize the theorem of LC^3 regret from [37] as follows.

Theorem 14. *(LC^3 Regret for finite dimensional, bounded features [37, Theorem 1.1]) Consider the finite dimension of ϕ as d_ϕ and that ϕ is uniformly bounded with $\|\phi(x,u)\|_2 \leq B$. The LC^3 algorithm [37, Algorithm 1] enjoys the following expected regret bound:*

$$\mathbb{E}_{LC^3}[\textit{Regret}_T] \leq \widetilde{\mathcal{O}}\left(\sqrt{d_\phi(d_\phi + d_{\mathcal{X}} + H)H^3T} \cdot \log\left(1 + \frac{B^2\|W^\star\|_2^2}{\sigma^2}\right)\right) \tag{4.23}$$

where $\widetilde{\mathcal{O}}(\cdot)$ notation drops logarithmic factors in T and H.

By revisiting this result, we provide our main statement as follows.

Theorem 15. *[Main Result] Set $\lambda = \bar{\sigma}^2/\|W^\star\|_2^2$. Our algorithm learns a sequence of policies $\pi^0, \ldots, \pi^{T-1}$ in T episodes, such that in expectation, we have:*

$$\mathbb{E}\left[\textit{Regret}_T\right] \leq \widetilde{\mathcal{O}}\left(H\sqrt{Hr(r+n+H)T}\right).$$

Also with probability at least $1 - O(\delta)$, we have that for all $t \in [T], h \in [H]$, $h(x_h^t) \geq -\mathcal{O}(L\epsilon/\eta)$.

Proof. For safety consideration, we proved that the sequence of policies learned from our Algorithm 1 satisfying Eq. (4.21) are all approximately safe, i.e. $h^s(x_h^t) \geq -\mathcal{O}(L\epsilon/\eta)$, with probability at least $1 - O(\delta)$ for all $t \in [T], h \in [H]$ (See Theorem 11 and Proposition 13).

For the regret analysis, our proof mainly follows Theorem 14 for LC^3 algorithm and its proofs in [37]. Readers are encouraged to refer to [37] for more details. One key assumption that allows for regret bound in Eq. (4.23) lies in the setting of optimism in the face of uncertainty that computes the optimal policy from unconstrained policy class Π

$$\pi^t := \arg\min_{\pi \in \Pi} \min_{W \in \text{Ball}_t} J^\pi(x_0^t; c, W) \tag{4.24}$$

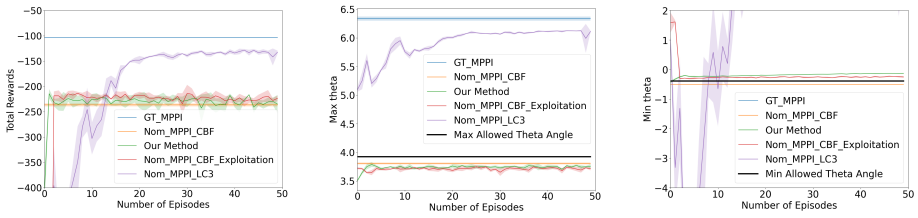
Similarly, in our analysis, by considering the constrained policy class $\Pi_{\widetilde{W}}$ defined in Eq. (4.21) and our optimism setup in Eq. (4.22) analogous to Eq. (4.24), our regret analysis naturally follows LC^3 regret in Eq. (4.23) and enjoys the regret bound with safety guarantee as follows

$$\mathbb{E}\left[\text{Regret}_T\right] \leq \widetilde{\mathcal{O}}\left(H\sqrt{Hr(r+n+H)T}\right)$$

where $\widetilde{\mathcal{O}}(\cdot)$ notation drops logarithmic factors. Thus we conclude the proof of Theorem 15. □

4.6 Results

We evaluate our Optimism-based Safe Learning framework on two simulation platforms: inverted pendulum and mobile robot navigation. In the implementation, we use Random Fourier Features (RFF) [108] to represent the known feature mapping $\phi(x, u)$, model predictive path integral control (MPPI) [60] to obtain control sequence under a learned system dynamics, and posterior reshaping with scaling of posterior covariance for Thompson Sampling during exploration process. The cumulative rewards (negative cost) are used for evaluations, i.e. the higher the better.

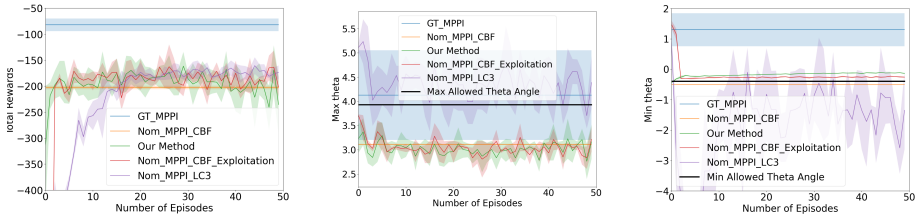


(a) (b) (c)

Figure 4.1: Performance curves of (a) cumulative rewards, (b) maximum theta angle, and (c) minimum theta angle in Inverted Pendulum environment testing under the same initial condition.

Inverted Pendulum

First we use the inverted pendulum modified from the OpenAI gym environment [109] with additive disturbance of $0.05\cos(\theta_t - 3)$ on state update to demonstrate the learning performance for control task. The pendulum has ground truth mass $m = 1$ and length $l = 1$, and is controlled by the limited torque input $u \in [-15, 15]$. The standard cost function $c = \theta^2 + 0.1\dot{\theta}^2 + 0.001$ is used to learn the optimal policy keeping the pendulum upright (i.e. $\theta = 0$). The control barrier functions $h_1^s = \theta + 1/8\pi \geq 0$ and $h_2^s = 5/4\pi - \theta \geq 0$ are designed to describe the



(a) (b) (c)

Figure 4.2: Performance curves of (a) cumulative rewards, (b) maximum theta angle, and (c) minimum theta angle in Inverted Pendulum environment with different initial conditions.

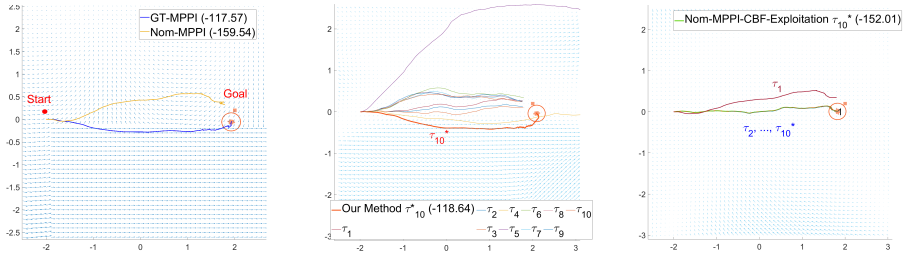
safety constraint $\theta \in [-1/8\pi, +5/4\pi]$ radians. We define the true system dynamics as $\theta_{t+1} = \theta_t + \dot{\theta}_{t+1}\Delta t + 0.05\cos(\theta_t - 3)$ and $\dot{\theta}_{t+1} = \dot{\theta}_t + \frac{3g}{2l}\sin\theta_t \Delta t + \frac{3}{ml^2}u\Delta t$.

To describe the partially known system dynamics, we assume a nominal model as $\theta_{t+1} = \theta_t + \dot{\theta}_{t+1}\Delta t$ and $\dot{\theta}_{t+1} = \dot{\theta}_t + \frac{3g}{2l'}\sin\theta_t\Delta t + \frac{3}{m'l'^2}u\Delta t$ with incorrect model parameters $m' = 1.8, l' = 1.8$ available to the learner (hence 80% error in model parameters). Using the same and different initial conditions respectively, Figure 4.1 and Figure 4.2 compare the cumulative rewards, maximum and minimum theta angle achieved during testing after each training episode by using (1) MPPI [60] with ground-truth dynamics model (GT-MPPI), (2) MPPI with nominal dynamics model and CBF (Nom-MPPI-CBF), (3) our method of optimism-based safe learning (Algorithm 1), (4) our method with exploitation only, i.e. replace Line 3 in Algorithm 1 by $\widetilde{W}_t \leftarrow \overline{W}_t$ (Nom-MPPI-CBF-Exploitation), and (5) unconstrained Lower Confidence-based Continuous Control algorithm (LC3) [37]. The last three learning-based algorithms are trained for 50 episodes with 20 testing trials after each training episode averaged from four random seeds. It is observed that our method quickly increased cumulative reward in early stage while satisfying the safety constraints as learning process evolves, and our method using exploration behavior (our method) is able to increase reward even faster than our method using exploitation behavior (Nom-MPPI-CBF-Exploitation), empirically implying sample efficiency. In contrast, GT-MPPI and LC3 severely violate angle limitation due to lack of safety consideration, and safe MPPI using CBF with nominal model (Nom-MPPI-CBF) still violates safety constraints with lower cumulative rewards due to the inaccurate nominal model with large error.

Mobile Robot Navigation

To compare our method with exploration behavior (Algorithm 1) against exploitation (Nom-MPPI-CBF-Exploitation) in terms of sample efficiency, consider a mo-

bile robot navigation task simulated in Matlab where the unicycle robot moves in a wind field without safety concerns. Here we assume the unicycle dynamics is directly available as the nominal model for the learner but suffers from unknown wind field defined by $d^\star(x) := [\cos(x_1-4)(x_2-3),\ \sin(x_1-4)(x_2-3)]^\top \in \mathbb{R}^2$ with $x = [x_1, x_2]^\top$ as the position of the robot. In particular, there is a rectangle area $[-2, 3] \times [-2.6, -0.2]$ in Figure 4.3(a) where the wind $d^\star(x)$ has uniform directions (East pointing) with larger magnitude. We use standard quadratic normalized cost $c = (x - x_{goal})^\top Q(x - x_{goal}) + u^\top Ru$ where Q, R are positive-definite to reflect the cost to go and to learn the optimal policy driving the robot towards the goal. As shown in Figure 4.3(a), with the ground-truth wind model, GT-MPPI [60] is able to plan the trajectory that takes advantage of the wind field to enjoy the lowest cost (highest reward). It is observed that our method with optimism-based exploration behavior (Algorithm 1) in Figure 4.3(b) is able to quickly find a near-optimal trajectory after data collection during training in 10 episodes. The predicted wind field correctly reflect the significant different wind distribution in the rectangle area due to the exploration process. This outperforms our method using only exploitation behavior in Figure 4.3(c) that quickly converges to a local minima without much exploration in the unknown wind field. Thus it fails to find the different wind field below in the rectangle area that could potentially yield improved solution, and has a large prediction error compared to the ground truth in Figure 4.3(a). This empirically validated the sample efficiency and the control performance of our method (Algorithm 1).



(a) (b) (c)

Figure 4.3: Mobile robot navigation trajectories in an unknown wind field. (a) ground-truth wind field and trajectories from GT-MPPI (rewards: -117.57) and Nom-MPPI (rewards: -159.54), (b) predicted wind field and trajectories from our method (Algorithm 1) during training and testing after 10 episodes (rewards: -118.64), (c) predicted wind field and trajectories from our method using exploitation behavior during training and testing after 10 episodes (rewards: -152.01).

4.7 Conclusion and Discussions

In this chapter, we address the problem of episodic safe learning for online nonlinear control tasks. Compared to existing safe learning and control approaches that exhaustively expanding safety region or optimizing policy performance without efficiency guarantee, we propose an optimism-based online safe learning framework that simultaneously achieve sample efficient learning for safe behaviors and nonlinear control optimization with bounded regret guarantee. We believe our presented work is an important first step to bridge provably efficient learning based methods and model based safety-critical control with formal guarantees.

To relax the limitation on the relative degree of safety constraint h^s, future work include extending our sample efficient learning for complex dynamical systems with higher relative degree safety constraints, e.g. using Exponential Control Barrier Functions [31].

Part II

Resilient Multi-Robot Networking

Resilience in terms of preserving the integrity of the multi-robot system itself through communication maintenance is another challenge when deploying robots to perform a set of tasks over long periods of time. In most cooperative robotic applications, the robotic members need to communicate or take observations from other robots for collective decision making and coordinated behaviors, which is often achieved through proximity-based information-exchange networks. One common assumption is the presence of sufficiently connected networks within the coherent multi-robot systems, which could be easily broken when the robot team spreads out over multiple widely separated task areas. Standard approaches often enforce a predefined network topology for robots to follow [40] or switching to cohesive behaviors when the network is about to disconnect [67, 70, 71, 110, 111], which could unnecessarily dominate the original robot behaviors and are not able to handle continuous robot failures. How to effectively retain the ability for the robots to 1) satisfy connectivity requirements across different levels and 2) recover network connectivity from faults while efficiently maintaining their original task-prescribed behaviors remain challenging.

To address these challenges, in this part we will focus on the following two problems.

- ***Multi-Robot Networking with Global and Subgroup Connectivity Maintenance*** *(Chapter 5)* by combining graph-theoretic and control-theoretic approaches to derive computationally-efficient control framework for multi-robot connectivity maintenance accounting for global communication as well as required local subgroup communication.

- ***Resilient Multi-Robot Networking in presence of Robot Failures*** *(Chapter 6)* by incorporating robustness and resilience graph design into the multi-robot connectivity control framework with provably correct convergence to the desired robust multi-robot network. This enables the robot team to satisfy any connectivity requirements demanded by human operators, providing the capability of recovering or enhancing system integrity through proximity-based networking in presence of continuous robot failures.

Chapter 5

Multi-Robot Networking with Global and Subgroup Connectivity Maintenance

Multi-robot systems are well known for the capabilities of accomplishing challenging tasks via cooperative behaviors or collaborations as a system. In many situations, it may be more appropriate and efficient to have the multi-robot systems *simultaneously* performing multiple behaviors in different subgroups while remaining connected. For example, having a robot team split into multiple operating subgroups flocking to multiple task areas at the same time as shown in Fig. 5.1. As subgroups may be formed based on the particular combinations of robots with heterogeneous capabilities, when the robot team spreads out over multiple widely separated task areas, robots in the same subgroup for a designated task area are expected to stay *locally connected by themselves* as one coherent component for efficient local collaboration. Global connectivity is still required to allow for global coordination among different subgroups, e.g. redistribution of robots due to dynamic task reallocation over time. Thus it is also necessary to ensure connectivity within each subgroup and across subgroups as well as global connectivity. We call this ability of multi-robot systems to accommodate different behaviors simultaneously within a single connected robot team while maintaining safety (collision avoidance with other robots and possibly obstacles) and within and across subgroup connectivity *Behavior Mixing*.

To the best of our knowledge, there is no existing work on connectivity maintenance that can ensure both global connectivity and subgroup connectivity at the same time for behavior mixing. The problem of behavior mixing with the global and subgroup connectivity constraints faces many challenges: (a) possible com-

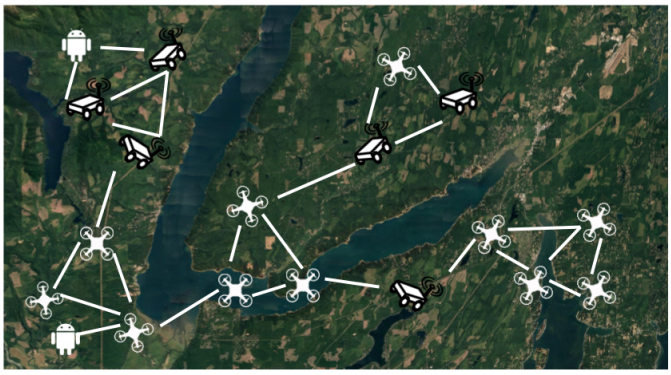


Figure 5.1: Example of multi-robot behavior mixing with global and subgroup connectivity maintenance.

munication disconnection due to performing multiple and potentially conflicting tasks at the same time; (b) the additional constraints imposed on robots with different tasks by the subgroup connectivity maintenance may lead to task failure, for example dead locks that might prevent the desired execution of behaviors; and (c) conservative robots motion incurred by the perturbation of connectivity on control outputs between different task groups.

To address these challenges, we propose a novel distributed *Minimum Connectivity Constraint Spanning Tree (MCCST)* method [15] to compute the set of communication links for the robots to maintain, which (a) has minimum cardinality, and (b) invokes the connectivity constraints for global and subgroup connectivity that would be least likely to be violated by the original controllers. With that, we then employ the control barrier functions (CBF) [40] to formulate our invoked connectivity constraints and collision avoidance constraints with respect to the controllers. Minimum connectivity maintenance is thus achieved by minimally modifying the original controllers subject to the constraints. Our work in this chapter presents the following contributions: (1) a generalized constrained behavior mixing framework to enable *simultaneous execution of different behaviors and sequences of behaviors within a single robot team,* while ensuring global and subgroup connectivity and collision avoidance; (2) a novel distributed MCCST method with quantified relationship between original task-related controllers and connectivity constraints to efficiently select *real-time minimum behavior mixing connectivity constraints with provably optimality guarantee*, (3) *computationally efficient* construction of MCCST that is *scalable* to large number of robots and suitable for real-time com-

putation to accommodate dynamic changes in the environment.

5.1 Global and Subgroup Connectivity Maintenance Problem

Consider a robotic team $\mathcal{S}$ consisting of N mobile robots in a planar space, with the position and single integrator dynamics of each robot $i \in \{1, \ldots, N\}$ denoted by $x_i \in \mathbb{R}^2$ and $\dot{x}_i = u_i \in \mathbb{R}^2$ respectively. Each robot can connect and communicate directly with other robots within its spatial proximity. The communication graph of the robotic team is defined as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where each node $v \in \mathcal{V}$ represents a robot. If the spatial distance between robots $v_i, v_j \in \mathcal{V}$ is less or equal to the communication radius R_c (i.e. $\|x_i - x_j\| \leq R_c$), then we assume the two can communicate and edge $(v_i, v_j) \in \mathcal{E}$ is undirected (i.e. $(v_i, v_j) \in \mathcal{E} \Leftrightarrow (v_j, v_i) \in \mathcal{E}$).

The joint robot states can be represented by $\mathbf{x} = \{x_1, \ldots, x_N\} \in \mathbb{R}^{2N}$ and we define the minimum inter-robot safe distance as $R_s \in \mathbb{R}$, for any pair-wise inter-robot collision avoidance constraint between robots i and j. We have the following condition defining the safe set of $\mathbf{x}$.

$$\begin{aligned} h^s_{i,j}(\mathbf{x}) &= \|x_i - x_j\|^2 - R_s^2, \qquad \forall i > j \\ \mathcal{H}^s_{i,j} &= \{\mathbf{x} \in \mathbb{R}^{2N} : h^s_{i,j}(\mathbf{x}) \geq 0\} \end{aligned} \tag{5.1}$$

The set of $\mathcal{H}^s_{i,j}$ indicates the safety set from which robot i and j will never collide. For the entire robotic team, the safety set can be composed as follows.

$$\mathcal{H}^s = \bigcap_{\{v_i, v_j \in \mathcal{V} : i > j\}} \mathcal{H}^s_{i,j} \tag{5.2}$$

[29] proposed the safety barrier certificates $\mathcal{B}^s(\mathbf{x})$ using control barrier functions $h^s_{i,j}(\cdot)$ that map the constrained safety set Eq. (5.2) of $\mathbf{x}$ to the admissible joint control space $\mathbf{u} \in \mathbb{R}^{2N}$ for ensuring $h^s_{i,j}(\cdot) \geq 0$ at all time. The result is summarized as follows.

$$\mathcal{B}^s(\mathbf{x}) = \{\mathbf{u} \in \mathbb{R}^{2N} : \dot{h}^s_{i,j}(\mathbf{x}) + \gamma h^s_{i,j}(\mathbf{x}) \geq 0, \forall i > j\} \tag{5.3}$$

where γ is a user-defined parameter to confine the available sets. It is proven in [29] that the forward invariance of the safety set $\mathcal{H}^s$ is ensured as long as the joint control input $\mathbf{u}$ stays in set $\mathcal{B}^s(\mathbf{x})$. In other words, the robots will always stay safe if they are initially inter-robot collision free and the control input lies in the set $\mathcal{B}^s(\mathbf{x})$. The constrained control space in Eq. (5.3) corresponds to a class of linear constraints over pair-wise control inputs u_i and u_j.

Likewise, if the connectivity constraint is enforced between pair-wise robots i and j to ensure inter-robot distance not larger than communication range R_c, we have

$$\begin{aligned} h^c_{i,j}(\mathbf{x}) &= R_c^2 - \|x_i - x_j\|^2 \\ \mathcal{H}^c_{i,j} &= \{\mathbf{x} \in \mathbb{R}^{2N} : h^c_{i,j}(\mathbf{x}) \geq 0\} \end{aligned} \tag{5.4}$$

The set of $\mathcal{H}^c_{i,j}$ indicates the feasible set on $\mathbf{x}$ from which robot i and j will never lose connectivity. Consider any connectivity spanning graph $\mathcal{G}^c = (\mathcal{V}, \mathcal{E}^c) \subseteq \mathcal{G}$ to enforce, the corresponding constrained set can be composed as follows.

$$\mathcal{H}^c(\mathcal{G}^c) = \bigcap_{\{v_i, v_j \in \mathcal{V} : (v_i, v_j) \in \mathcal{E}^c\}} \mathcal{H}^c_{i,j} \tag{5.5}$$

Similar to the safety barrier certificates in Eq. (5.3), the connectivity barrier certificates [40] are defined as follows indicating another class of linear constraints over pair-wise control inputs u_i and u_j for $(v_i, v_j) \in \mathcal{E}^c$ at any time point t.

$$\mathcal{B}^c(\mathbf{x}, \mathcal{G}^c) = \{\mathbf{u} \in \mathbb{R}^{2N} : \dot{h}^c_{i,j}(\mathbf{x}) + \gamma h^c_{i,j}(\mathbf{x}) \geq 0, \forall (v_i, v_j) \in \mathcal{E}^c\} \tag{5.6}$$

In behavior mixing, we assume the robotic team is tasked with M simultaneous behaviors and has been partitioned into M sub-groups $\mathcal{S} = \{\mathcal{S}_1, \ldots, \mathcal{S}_M\}$, with each robot i already assigned to a sub-group $\mathcal{S}_m$ and with original task-related controller $u_i = \hat{u}_i$. To ensure successful behavior mixing, the global connectivity graph $\mathcal{G}$ and the induced subgroup connectivity graph $\mathcal{G}_m = \mathcal{G}[\mathcal{V}_m] \subseteq \mathcal{G}$ where $\mathcal{V}_m \subseteq \mathcal{V}$ containing robots within the same sub-group $\mathcal{S}_m$ for all $m = 1, \ldots, M$ should be connected at all time. We assume these connectivity constraints are satisfied initially. With the defined forms of safety and connectivity constraints in Eq. (5.3) and Eq. (5.6), we formally define the *behavior mixing* problem as a bilevel optimization process at each time step as follows.

$$\mathbf{u}^* = \underset{\mathcal{G}^c, \mathbf{u}}{\arg\min} \sum_{i=1}^{N} \|u_i - \hat{u}_i\|^2 \tag{5.7}$$

$$\begin{aligned} \text{s.t.} \quad & \mathcal{G}^c = (\mathcal{V}^c, \mathcal{E}^c) \subseteq \mathcal{G} \quad \text{is connected} \\ & \mathcal{G}_m = \mathcal{G}^c[\mathcal{V}_m] \quad \text{is connected} \quad \forall m = 1, \ldots, M \end{aligned} \tag{5.8}$$

$$\mathbf{u} \in \mathcal{B}^s(\mathbf{x}) \bigcap \mathcal{B}^c(\mathbf{x}, \mathcal{G}^c), \quad \|u_i\| \leq \alpha_i, \forall i = 1, \ldots, N \tag{5.9}$$

This bilevel optimization problem can be solved by two-steps: find 1) the optimal connectivity spanning graph $\mathcal{G}^{c*} \subseteq \mathcal{G}$ to preserve, and 2) the one-step control inputs $\mathbf{u}^* \in \mathbb{R}^{2N}$ bounded by maximum velocities $\{\alpha_i\}$ and minimally deviated from $\hat{u}_i$ subject to constraints in Eq. (5.9) with $\mathcal{G}^c = \mathcal{G}^{c*}$.

5.2 Behavior Mixing using Minimum Connectivity Constraints

5.2.1 Minimum Connectivity Constraint Spanning Tree (MCCST)

First we consider the sub-problem of selecting optimal connectivity spanning graph $\mathcal{G}^{c*} \subseteq \mathcal{G}$ in Eq. (5.7) that introduces minimum connectivity constraints. As each edge $(v_i, v_j) \in \mathcal{E}^c$ in a candidate graph $\mathcal{G}^c$ enforces one constraint between robot i, j in Eq. (5.6), the graph $\mathcal{G}^{c*}$ whose edges define the minimum connectivity constraints must exist among the set of all spanning trees $\mathcal{T}$ of $\mathcal{G}$ that have the minimum number of edges (i.e. $N-1$) for $\mathcal{G}^{c*}$ to stay connected.

Hence, the problem boils down to find the optimal spanning tree $\mathcal{G}^{c*} = \mathcal{T}^{c*} \in \mathcal{T}$ of $\mathcal{G}$ whose edges invoke the minimum connectivity constraints in the form of Eq. (5.6) over the robots' controllers. To quantify the strength of connectivity constraint by an edge $(v_i, v_j) \in \mathcal{E}$, we introduce the weight assignment defined as follows.

$$w_{i,j} = \dot{h}^c_{i,j}(\mathbf{x}, \hat{u}_i, \hat{u}_j) + \gamma h^c_{i,j}(\mathbf{x}), \forall (v_i, v_j) \in \mathcal{E} \tag{5.10}$$

Compared to the connectivity constraint in Eq. (5.6), $w_{i,j}$ indicates the violation of the pair-wise connectivity constraint between the two robots under the original controllers $\hat{u}_i, \hat{u}_j$, with the higher value of $w_{i,j}$ the less violated the connectivity constraint is. This quantifies how likely the existing connectivity link is going to break if no revision made to the controller. It is desired to preserve those links with larger $w_{i,j}$ implying smaller revision needed for the controllers to keep the links connected. With that, each candidate spanning tree $\mathcal{T}^c \in \mathcal{G}$ is redefined as a weighted spanning tree $\mathcal{T}^c_w = (\mathcal{V}, \mathcal{E}^T, \mathcal{W}^T)$ with $\mathcal{W}^T = \{-w_{i,j}\}$. Hence the optimal connectivity graph $\mathcal{G}^{c*}$ with constraints in Eq. (5.8) can be obtained as follows.

$$\begin{aligned} \mathcal{G}^{c*} &= \underset{\mathcal{T}^c_w \in \mathcal{T}}{\arg\max} \sum_{(v_i,v_j)\in\mathcal{E}^T} w_{i,j} = \underset{\mathcal{T}^c_w \in \mathcal{T}}{\arg\min} \sum_{(v_i,v_j)\in\mathcal{E}^T} -w_{i,j} \\ \text{s.t.} \quad & \mathcal{T}_m = \mathcal{T}^c_w[\mathcal{V}_m] \quad \text{is connected} \quad \forall m = 1, \dots, M \end{aligned} \tag{5.11}$$

The optimal solution of Eq. (5.11) is the Minimal Spanning Tree (MST) weighted by $\{-w_{i,j}\}$ and constrained by sub-group connectivity requirements. We propose to define another class of spanning trees as follows and relate its unconstrained MST to the solution of the constrained MST in Eq. (5.11).

Definition 16. *Given a connectivity graph $\mathcal{G}$ and for all edges $(v_i, v_j) \in \mathcal{E}$ on $\mathcal{G}$,*

redefine their weights by the following.

$$w'_{i,j} = \begin{cases} \lambda \cdot w_{i,j}, & \textit{if} \quad v_i \textit{ and } v_j \textit{ are in the same sub-group} \\ w_{i,j}, & \textit{if} \quad v_i \textit{ and } v_j \textit{ are in different sub-groups} \end{cases} \tag{5.12}$$

where $\lambda \in \{\lambda \gg 1 : \lambda \cdot w_{i,j} \gg w_{i',j'}, \forall v_i, v'_i, v_j, v'_j \in \mathcal{V}\}$ *is a unique user-defined constant for the entire graph* $\mathcal{G}$*. The weight-modified graph is denoted as* $\mathcal{G}'$*. Then we call the redefined spanning tree* $\mathcal{T}_w^{c'} = (\mathcal{V}, \mathcal{E}^T, \mathcal{W}^{T'})$ *as the Connectivity Constraint Spanning Tree (CCST).*

The Definition 16 introduces a new class of spanning trees (CCST) $\mathcal{T}_w^{c'}$ equivalent to the original spanning trees $\mathcal{T}_w^c$ with inflated weights over the edges connecting robots in the same sub-group. In particular, the designed parameter λ in Eq. (5.12) ensures that after inflation the new weights $-w'_{i,j}$ over edges connecting different subgroups are always larger than any edges within all the subgroups for $\mathcal{T}_w^{c'}$. As we will prove by the following Lemma 18 and Theorem 19, this guarantees that the computed MST $\mathcal{T}_w^{c'}$ becomes the solution of constrained MST $\mathcal{T}_w^c$ in Eq. (5.11), namely, the MST $\mathcal{T}_w^{c'}$ contains the MST of each subgroup as well, ensuring that the subgroups are also connected in an optimal way. We review some useful definitions in graph theory [112]:

- *fragment*: a subtree of Minimum Spanning Tree;
- *outgoing edge*: a edge of a fragment if one adjacent node is in the fragment and the other is not.

The first definition describes that a connected set of nodes and edges of the MST is called a fragment. By this definition, a single node is also a fragment by itself. In the following discussion, we focus on *minimum-weight outgoing edge (MWOE)*, which is the edge with minimum weight among all outgoing edge of a fragment.

Lemma 17. *Let* e_{min} *be a minimum-weight outgoing edge (MWOE) of a fragment. Connecting* e_{min} *and its adjacent node in a different fragment yields another fragment in MST.*

The proof of Lemma 17 can be found in both [112] and [113].
With Lemma 17, the process of constructing MST is as follows [112]:

- Each node starts as a fragment by itself
- Each fragment iteratively connects with MWOE fragment

This process will result in the MST of the given graph.

Lemma 18. *By following the process above on $\mathcal{G}'$ in Definition 16, all nodes within the same sub-group will form a MST fragment before connecting to other sub-group.*

Proof. We prove by contradiction. Suppose the node v_i from sub-group graph $\mathcal{G}'_i$ connects with node v_j first, which belongs to sub-group graph $\mathcal{G}'_j$, $i \neq j$. From the MST construction process we know that, at each iteration, the edge added is the minimum-weight outgoing edge of the connecting fragment. In this case, the weight $w_{i,j}$ of the edge between v_i and v_j is the minimum of all outgoing edges of v_i. Let $v_{i'} \in \mathcal{G}'_i$ where there exists an outgoing edge between v_i and $v_{i'}$, then we know that the weight $w'_{i,j} < w'_{i,i'}$. This contradicts with the property of $\mathcal{G}'$ in Eq. 5.12. □

Theorem 19. *Given the redefined Connectivity Constraint Spanning Tree (CCST) $\mathcal{T}_w^{c'} = (\mathcal{V}, \mathcal{E}^T, \mathcal{W}^{T'})$ in Definition 16 and denote minimum weight CCST as $\bar{\mathcal{T}}_w^{c'} = \arg\min_{\mathcal{T}_w^{c'} \in \mathcal{T}} \sum_{(v_i,v_j)\in\mathcal{E}^T} -w'_{i,j}$, we have: $\bar{\mathcal{T}}_w^{c'} = \mathcal{G}^{c*}$ in Eq. (5.11). Namely, the Minimum Spanning Tree $\bar{\mathcal{T}}_w^{c'}$ of $\mathcal{G}'$ is the optimal solution of $\mathcal{G}^{c*}$ in Eq. (5.11) and we call the graph $\bar{\mathcal{T}}_w^{c'}$ as Minimum Connectivity Constraint Spanning Tree (MCCST) of the original connected graph $\mathcal{G}$.*

Proof. From Lemma 18, edges *between* sub-groups will be connected only when edges *within* each sub-groups are connected. By definition, the MST of graph $\mathcal{G}'_i$ within subgroup $\mathcal{S}_i$ is optimal with minimum total weight, which means

$$\begin{aligned}
\bar{\mathcal{T}}_w^{c'}(i) &= \underset{\mathcal{T}_w^{c'}(i)\in\mathcal{T}(i)}{\arg\min} \sum_{(v_i,v_j)\in\mathcal{E}^{T(i)}} -w'_{i,j} \\
&= \underset{\mathcal{T}_w^{c'}(i)\in\mathcal{T}(i)}{\arg\min}\ \lambda \cdot \sum_{(v_i,v_j)\in\mathcal{E}^{T(i)}} -w_{i,j} \\
&= \underset{\mathcal{T}_w^{c'}(i)\in\mathcal{T}(i)}{\arg\min} \sum_{(v_i,v_j)\in\mathcal{E}^{T(i)}} -w_{i,j}
\end{aligned} \tag{5.13}$$

The equality holds since $\lambda > 0$. Then we consider v_i and v_j in different subgroups, i.e. $\mathcal{S}(v_i) \neq \mathcal{S}(v_j)$, while (v_i, v_j) is the edge in spanning tree edges $\mathcal{E}^{T(i)}$ connecting two subgroups. Then for the next step, connecting the minimum-weighted outgoing edge between different sub-groups, yields

$$\begin{aligned}
\bar{\mathcal{T}}_w^{c'} &= \underset{\mathcal{T}_w^{c'}\in\mathcal{T}}{\arg\min} \sum_{(v_i,v_j)\in\mathcal{E}^{T(i)}} -w'_{i,j}, \quad \mathcal{S}(v_i) \neq \mathcal{S}(v_j) \\
&= \underset{\mathcal{T}_w^{c'}\in\mathcal{T}}{\arg\min} \sum_{(v_i,v_j)\in\mathcal{E}^{T}} -w_{i,j}
\end{aligned} \tag{5.14}$$

With the same form as in Eq. (5.11), this concludes the proof. □

In this way, we relax the constrained MST optimization problem in Eq. (5.11) into unconstrained MST problem with the same optimality guarantee. The connectivity constraints from the obtained MCCST $\bar{\mathcal{T}}_w^{c'}$ are thus minimally violated by the current task-related controllers, implying the least restriction due to global and subgroup connectivity requirements. Such MCCST $\bar{\mathcal{T}}_w^{c'}$ therefore specifies the optimal connectivity graph $\mathcal{G}^{c*} \subseteq \mathcal{G}$ to enforce for behavior mixing in Eq. (5.8). Next, we will present a distributed method for computing MCCST.

Algorithm 2 Distributed MCCST Construction

Input: a: adjacency edge weight list of the original weighted graph
Output: edge list of MCCST

```
1:  function ConstructDistributedMCCST(A)
2:      A ← empty adjacency matrix
3:      A ← updated from input a
4:      while msg ← getNewMessage(msg_pool) do
5:          if not initialized then
6:              A ← initialRound(msg, A)
7:          else
8:              A ← processRound(msg, A)
9:          end if
10:         if isConnected(A) then
11:             return getEdgeList(A)
12:         end if
13:     end while
14:     if isEmpty(msg_pool) then
15:         resetRound()
16:     end if
17: end function
```

5.2.2 Construction of Distributed Minimum Connectivity Constraint Spanning Tree (MCCST)

Here we propose a distributed construction of MCCST of $\mathcal{G}$. For our problem setting, the topology and weights could change over time, thus a time-optimal real-time algorithm is needed. We develop our algorithm based on the work from [112, 113, 114], but reduce the computation time while sacrificing message optimality. Different from most of the network algorithms such as [113, 114], our algorithm does not require synchronization, which also reduces the total time. Note that MST is unique for a graph with unique edge weights. Therefore the result is the same from centralized and decentralized construction.

A detailed description of the algorithm is as follows:

Overview

Given a graph $\mathcal{G}' = \{\mathcal{V}, \mathcal{E}\}$ with weights defined in Definition 16 where $|\mathcal{V}| = N$ is the number of robots, the initial state of the system is a singleton graph where each vertex is an individual isolated node without any outgoing edge, and each node is given a distinct id. This gives N fragments and each consists of one vertex. Then each fragment finds the *minimum-weight outgoing edge (MWOE)* and connect with neighboring fragments. Iteratively, the forest of fragments will join as a spanning tree connecting all vertices of the graph, resulting as the MCCST.

As shown in Algorithm 2, each robot takes an input of neighboring edge weights and connectivity information, then outputs the computed MCCST edge list. The incoming message is processed according to whether the node is being initialized or not. The process will reset when there is no new message in the message pool, which implies all the fragments finish updating within themselves and new MWOE need to be connected and a new round begins.

Initial Round

Algorithm 3 Initial Round of MST Construction

Input: msg: incoming messages, A: current adjacency matrix
Output: A: updated adjacency matrix

function INITIALROUND(msg, A)
    connect with neighbor with $MWOE$
    $leader\_id \leftarrow$ min($self\_id$, $neighbor\_id$)
    $A \leftarrow$ update with msg
    if no new information in msg **then**
        finish initial round
    end if
    send $init$ message to msg_pool
    return A
end function

In the initial round, as shown in Algorithm 3, each fragment initially only contains one vertex. Each node directly connect with the neighbor with MWOE. However, to keep information within a fragment consistent and avoid additional computation, the node with the smallest id is selected as fragment leader. Information keeps updating within the fragment until every node has the same adjacency matrix of its fragment tree.

Processing Round

Algorithm 4 Processing Round of MCCST Construction

Input: msg: incoming messages, A: current adjacency matrix
Output: A: updated adjacency matrix

function PROCESSROUND(msg, A)
  if $leader\_id$ is $self\_id$ **then**
    $MWOE\_cache\_list \leftarrow$ wait($fragment\_node$)
    $MWOE \leftarrow$ min($MWOE\_cache\_list$)
    if all $fragment\_node$ reported **then**
      inform the one with $MWOE$ to connect
    end if
  end if
  if not reported to leader **then**
    if no MWOE info **then**
      $msg\_pool[MWOE] \leftarrow get\_info$ message
    end if
    if get information from MWOE neighbor **then**
      $msg\_pool[leader\_id] \leftarrow report$ message
    end if
  end if
  $A \leftarrow$ update with msg
  return A
end function

At each processing round, the fragment leader will determine the minimum-weight outgoing edge (MWOE) in its fragment after receiving all MWOE information from each fragment node (including itself). Since each node in the fragment only has the local knowledge within its own fragment, it will ask the MWOE neighbor for their fragment information, i.e. adjacency matrix, leader id. Whenever a node receives a request to give information, it will reply accordingly. Once each node receives information from MWOE neighbor, it will report to the fragment leader. All connect requests will be accepted and this, by lemma 17, always yields a fragment. When a new connection is made, the two fragments will combine their information and update all the nodes within the fragment with the new information. Iteratively, the construction process will end when every node receives the same updated adjacency matrix representing the MST of the graph. Since only the leader of each fragment updates the adjacency matrix within the fragment, eventually when the algorithm terminates, there will be only one fragment, i.e. the MST, with one leader, marking the convergence of the distributed algorithm. The convergence speed of our distributed MCCST algorithm is dependent on the topology of the original communication graph and edge weights, ranging from one iteration

to $O(\log N)$ iterations with a worst case time complexity of $O(N \log N)$.

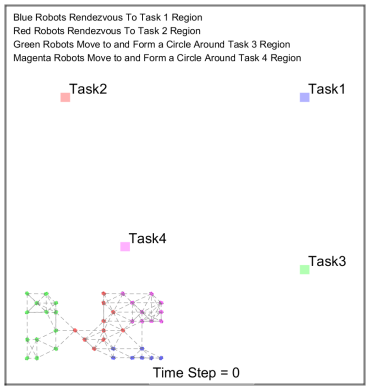
Once the final MCCST is obtained as the optimal connectivity graph $\mathcal{G}^c = \mathcal{G}^{c*}$ in Eq. (5.9), we can specify the safety and connectivity barrier certificates Eq. (5.3) and Eq. (5.6) to invoke a set of linear constraints. Thus the original quadratic programming (QP) problem in Eq. (5.7) could be efficiently solved to get optimal revised robot controllers satisfying safety and global and subgroup connectivity constraints for behavior mixing.

5.3 Results

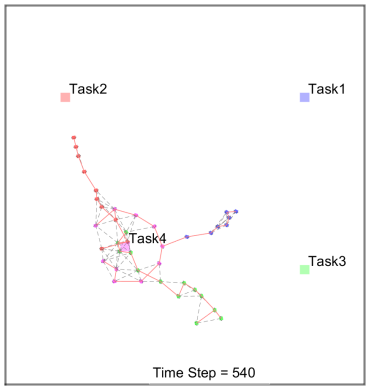
Simulation Example

The first set of experiments are performed on a team of $N = 40$ mobile robots with unicycle dynamics as shown in Figure 5.2. The robot team is divided into $M = 4$ subgroups with different colors and is tasked with 4 parallel behaviors. In the figures, robots in blue subgroup 1 and red subgroup 2 execute biased rendezvous behaviors towards the blue task site 1 and red task site 2 respectively, while robots in green subgroup 3 and magenta subgroup 4 perform circle formation behaviors around the green task site 3 and magenta task site 4 respectively. For our MCCST method, we apply the minimally revised controllers from Eq. (5.7) with single-integrator dynamics to the robots with unicycle dynamics using kinematics mapping from [29]. As shown in Figure 5.2a-c, our distributed MCCST approach is able to generate real-time minimum connectivity graph (red edges) from the present connectivity graph (grey edges) so that the invoked connectivity constraints are minimally restrictive to the original behavior controllers. Most of the target behavior configurations have been accomplished as shown in Figure 5.2c. The communication relays connecting different subgroups are implicitly formed to provide greater flexibility for the rest of the robots without the need of explicit robots roles assignment as done in [73, 75]. This is because our algorithm enforces provably minimum connectivity graph that is least restrictive to the robots.

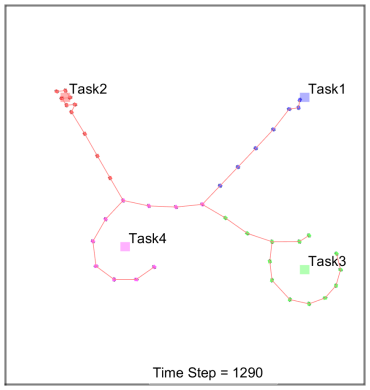
In comparison, we present converged results of other two methods with static connectivity graph in Figure 5.2e and Figure 5.2f respectively: i) always preserving communication edges in the initial MST (red) depicted in Figure 5.2d, and ii) always preserving edges in initial connectivity graph (grey) in Figure 5.2a as done in [68]. Since the invoked connectivity graph is fixed as the robots move, they can hardly achieve circle formation (Figure 5.2e) or could fall into deadlock (Figure 5.2f). Numerical results are provided in Figure 5.3 showing our method ensures safety and connectivity, while having minimal control perturbation due to connectivity and maximum task performance (very close to designated target area as shown from Figure 5.3d). Note that in Figure 5.2e the provided comparison



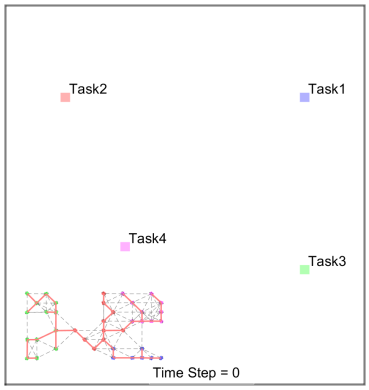
(a) Time Step $= 0$



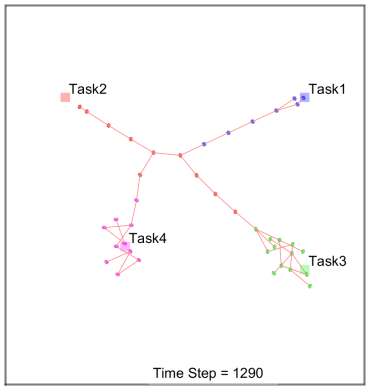
(b) Time Step $= 540$ (MCCST)



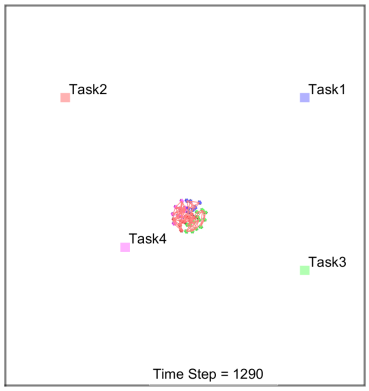
(c) Time Step $= 1290$ (MCCST, Converged)



(d) Time Step $= 0$

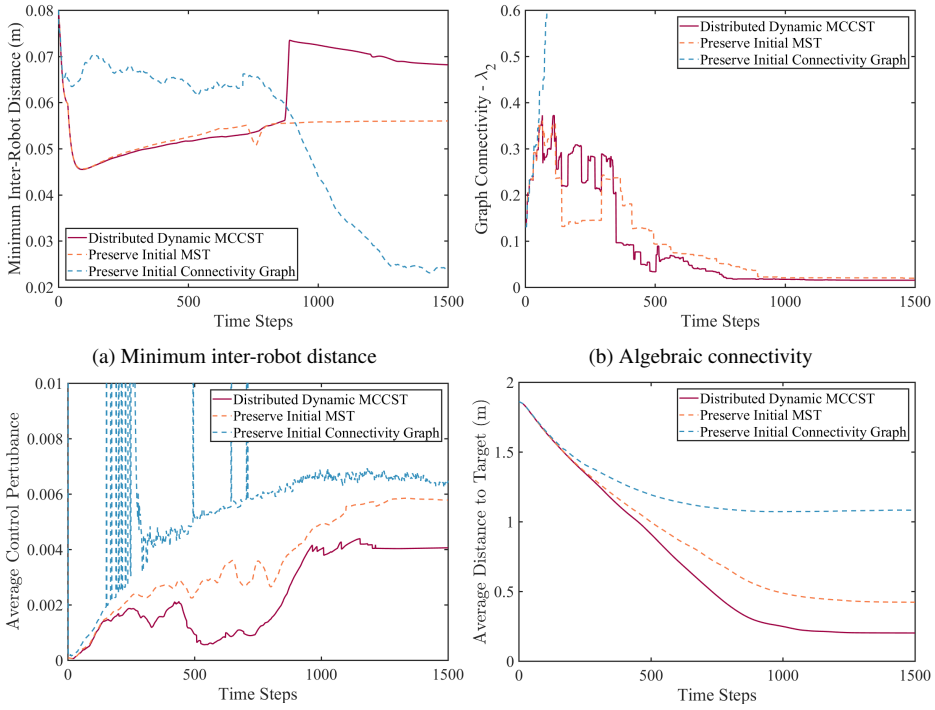


(e) Time Step $= 1290$ (Fixed Initial MST, Converged)



(f) Time Step $= 1290$ (Fixed Initial Connectivity, Converged)

Figure 5.2: Simulation example of 40 robots tasked to four different places with behaviour mixing.



(a) Minimum inter-robot distance

(b) Algebraic connectivity

(c) Average control perturbation

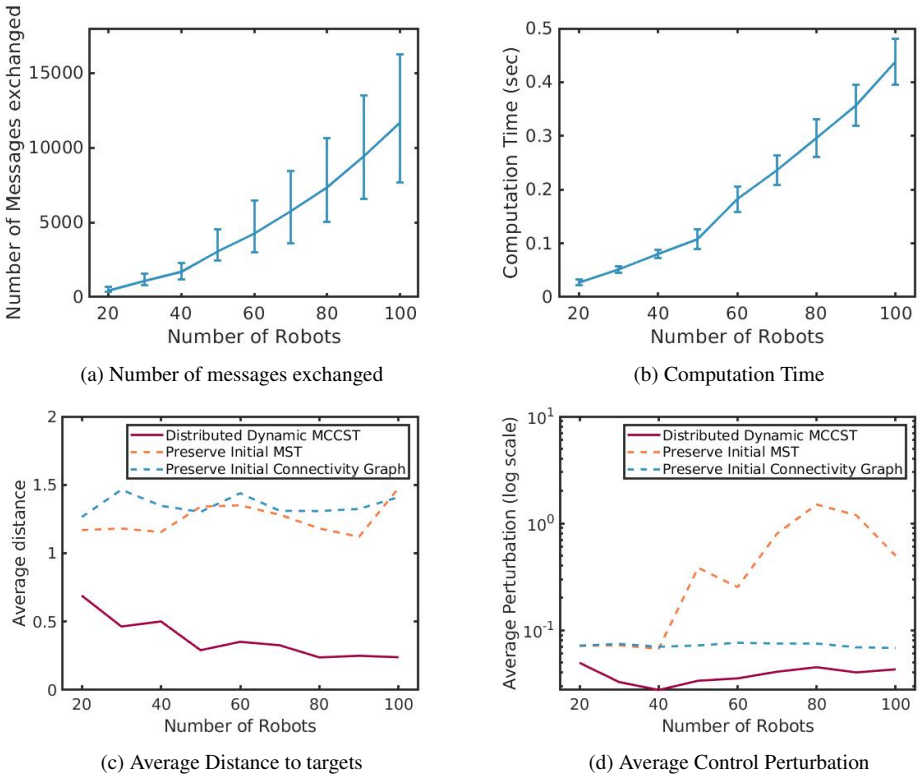
(d) Average distance to target region

Figure 5.3: Performance comparison of simulation example in Figure 5.2 w.r.t. different metrics: (a) Minimum inter-robot distance (safety distance is 0.02m), (b) Algebraic connectivity evaluated by second smallest eigenvalue of mutli-robot laplacian matrix. Positive meaning connectivity ensured, (c) Control perturbation computed by $\frac{1}{N}\sum_{i=1}^{N}\|u_i^* - \hat{u}_i\|^2$, (d) Average distance between robots to tasked region (the smaller the better).

method of preserving initial MST from our MCCST without updating in real-time is already better than other barrier certificate based connectivity controllers [29, 78] that impose predefined fixed connectivity graph not necessarily as optimal for the tasks.

Quantitative Results

For validating the computation efficiency and scalabiltiy of our algorithm, we run experiments with up to 100 robots and 4 parallel behaviors (four robots subgroups simultaneously rendezvous to four different places with safety and connectivity constraints). For Figure 5.4a and 5.4b, the experiment is done by computing the distributed MCCST 1500 to 3000 times, depending on the iterations for the system to converge, which varies with the number of robots and graph topology. The complexity of the worst case for both message and time is $O(N \log N)$. However, the average case, as shown in the figure, is better than $O(N \log N)$. Figure 5.4b



(a) Number of messages exchanged

(b) Computation Time

(c) Average Distance to targets

(d) Average Control Perturbation

Figure 5.4: Quantitative results summary. (a)-(b) are results from our proposed Distributed MCCST approach. (c)-(d) are comparison results with ours and the other two approaches with static connectivity graph but the same controller (5.7). (a) Number of messages exchanged during the distribute MCCST construction. The error bar shows the maximum and minimum number of messages exchanged. (b) Computation time of constructing the distributed MCCST. The error bar shows the standard deviation. (c) Average distance from robot to target location after converged. (d) Average control perturbation.

shows the time duration for computing the distributed MCCST, which shows that computing distributed MCCST could be done in real time with large number of robots.

The average distance to target region and perturbation after convergence is calculated from 10 runs for each batch of robots with up to 100 robots. In Figure 5.4c, the average distance to target with MCCST is significantly smaller (closer to target region) than with static connectivity graph. The distance also decreases as the number of robots increases, since only a limited number of robots are needed to maintain connectivity, which enables more robots to rendezvous to the target locations. However, for the other two methods with imposed static connectivity topology, the average distance increases with the number of robots. Figure 5.4d shows the result of average perturbation. Our method gives much smaller pertur-

bation on average. Note that the result from preserving initial MST gives much worse result than the other two, because the initial MST edges could give huge deviation from the optimal control outputs as behaviors progressed, while the full connectivity graph gives larger number of constrain edges to keep, so that some are canceled out with each other. Nevertheless, our distributed MCCST method always computes the minimum connectivity constraints, thus outperforming the other two methods significantly.

5.4 Conclusion and Discussions

In this chapter, we developed a bilevel optimization based minimum connectivity maintenance framework for behavior mixing. We proposed a distributed Minimum Connectivity Constraint Spanning Tree (MCCST) algorithm to compute provably minimum global and subgroup connectivity constraints in real-time. By formulating the invoked connectivity constraints and safety constraints using safety and connectivity barrier certificates, the robots controllers are minimally modified from the original controllers with dynamic and possibly discontinuous communication topology. Experimental results show that our method is scalable and computation efficient to large number of robots.

Our proposed framework could be combined with any nominal multi-robot controllers to provide the desired global and local connectivity guarantee while respecting their original behaviors. However, given the step-wise computation process of the optimization-based controllers, the method may not render optimal solution in terms of optimizing the primary multi-robot task over a particular mission time. In order to derive optimal multi-robot behaviors, one could integrate our connectivity control constraints into planning or MPC algorithms with the primary objective functions and then search for the optimal sequence of controllers respecting the connectivity requirements. On the other hand, while the resulting connectivity-aware multi-robot behavior could ensure the robot team stay connected as one group, the *minimum connectivity* nature could lead to vulnerable network topology of the robot team. For example, the failure of a single robot may disconnect the entire team. In Chapter 6 we will discuss resilient multi-robot connectivity maintenance designs with increased network redundancy to address potential adversaries from the environment.

Chapter 6

Resilient Multi-Robot Connectivity Maintenance

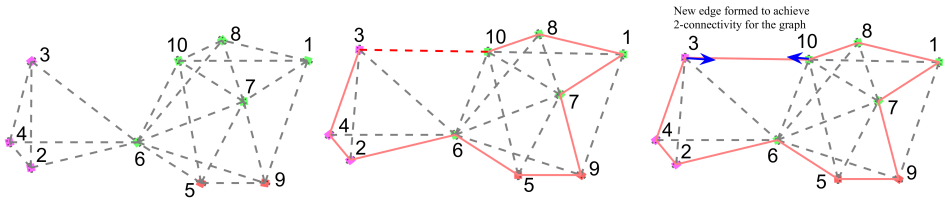
In Chapter 5 we have presented how to integrate the graph theory such as minimum spanning tree with optimization based control framework to guarantee multi-robot network connectivity during operation. In real-world applications, there are possible adversarial situations where the robotic team could suffer from unexpected robot failures, or continuously losing members under adversarial external attacks. Such adversaries could easily break the connectivity among robots and thus jeopardize the task operation of the entire team. This demands for a resilient strategy for connectivity maintenance so that the robots could be able to reconfigure their movements to maintain, recover, and increase the network connectivity in response to the faulty situations.

This is particular challenging for most existing work since (a) the resilience and robustness of the multi-robot network leads to increasing complexity over conventional connectivity control methods [67, 70, 71, 110] due to the possible discontinuity from dynamic topology changes as pointed out in [72], (b) conventional connectivity metrics such as algebraic connectivity is not suitable to explicitly model the network robustness as found by [42, 43], (c) there are often no optimality guarantee over the imposed connectivity constraints for original robot tasks [40, 78, 115] nor the perturbation from the connectivity controller to the robot original controllers [42, 43, 77], and (d) the network robustness is maintained in absence of robot failures [16, 42] and hence could be vulnerable to increasing number of robot failures over time.

Motivated by the challenges, in this chapter we aim to develop provably optimal algorithms for minimally disruptive and resilient connectivity maintenance for a team of connected robots. We assume the robots have been provided with

their original task-related controllers and seek to revise their controllers as necessary to achieve the desired resilient network connectivity and avoid collisions between robots and with obstacles. In particular, we propose a minimally disruptive resilient connectivity maintenance framework that, by inputting *any* desired value k of graph connectivity, the framework will first compute the provably optimal $k-$node connected minimum resilient graph (k-CMRG) whose edges invoke *min-size* pairwise connectivity constraints *least violated* by the robot original controllers. With the rendered optimal pairwise connectivity constraints, we employ the Finite-time Convergence Barrier Function (FCBF) from [78] to map the invoked pairwise spatial connectivity constraints to those over the robot original controllers, and minimally modify those controllers in the context of quadratic programming to respect the original tasks.

The thesis work in this chapter presents the following contributions: (1) a generalized resilient connectivity maintenance framework that jointly optimizes both the topological resilient connectivity graph and the constrained robot motions to minimally disrupt the original robot tasks, (2) a novel $k-$CMRG method to compute the optimal weighted $k-$node connected resilient graph for arbitrary initially connected multi-robot graph, imposing *least* connectivity constraints to the robots, (3) theoretical analysis and proof of the optimality of our k-CMRG with guaranteed, user-specified network resilient connectivity in presence of continuous robot failures.



(a) Initial 1-connected (k=1) graph

(b) Improved 2-connected (k=2) graph with red existing connectivity edges to preserve and one augmented edge (red dash edge) to form

(c) Robots perform their original task while ensuring the existing red edges stay connected and the augmented edge(s) are formed (i.e. robot 3 and 10 gets closer)

Figure 6.1: Simple example of resilient connectivity maintenance problem, where 10 robots are executing their original behaviors while preserving and forming the red connectivity edges defined by k-CMRG to achieve the desired connectivity (e.g. k=2 here). Connectivity edges (gray dash line) exist when two robots are within the limited communication range. Red solid edge are selected from existing gray connectivity edges and augmented with the selected red dash edge from non-exist connectivity edges to compose a k-CMRG with desired graph connectivity.

6.1 Resilient Connectivity Maintenance Problem

Consider a robotic team $\mathcal{S}$ consisting of n mobile robots in a planar space, with the position and single integrator dynamics of each robot $i \in \{1, \ldots, n\}$ denoted by $x_i \in \mathbb{R}^2$ and $\dot{x}_i = u_i \in \mathbb{R}^2$ respectively. Each robot can connect and communicate directly with other robots within its spatial proximity. The communication graph of the robotic team is defined as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where each node $v \in \mathcal{V}$ represents a robot. If the spatial distance between robot $v_i \in \mathcal{V}$ and robot $v_j \in \mathcal{V}$ is less or equal to the communication radius $R_c \in \mathbb{R}$ (i.e. $\|x_i - x_j\| \leq R_c$), then we assume the two can communicate and edge $(v_i, v_j) \in \mathcal{E}$ is undirected (i.e. $(v_i, v_j) \in \mathcal{E} \Leftrightarrow (v_j, v_i) \in \mathcal{E}$).

We assume the robotic team has been tasked with m simultaneous behaviors, partitioning the set of robots into m sub-groups. To simplify our discussion, we assume the sub-group partitions and behavior controllers are given or already derived from other multi-robot task allocation algorithms, namely, each robot i has been assigned to a sub-group with some behavior-prescribed controller $u_i = \hat{u}_i$. We also assume the current communication/connectivity graph $\mathcal{G}$ for the robots is connected as one component. Here the multi-robot network resilience is quantified by the network connectivity defined as follows [116].

Definition 20. *(k-node connected graph) A connected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is said to be k-node connected (or k-connected) if it has more than k nodes and remains connected whenever fewer than k nodes are removed.*

The objective is, given any user defined desired connectivity $k \in \mathbb{R}^+$, how to develop control laws for the robots to achieve and maintain it over time, even if the current connectivity graph is not $k-$node connected. This differentiates our work from most of the connectivity maintenance literature [15, 16, 40, 67, 78, 115] and thus provides greater freedom to enable flexible resilient connectivity maintenance. In the rest of the thesis, we will use $k-$connectivity to represent $k-$node connectivity. Then we would like to enforce such constraint as robots execute their behavior-prescribed controllers, so that the resulting time-varying connectivity graph $\mathcal{G}$ becomes and stays $k-$connected at all time. It is straightforward that $n \geq k + 1$ should be followed in order for the problem to be solvable [117, 118]. In presence of the above connectivity constraints as well as the physical constraints of the robots such as inter-robot collision avoidance and velocity limits, each robot i may have to modify their primary task-related controller $\hat{u}_i$ to accommodate the constraints. To that end, the objective is to 1) coordinately invoke active constraints to follow (particularly the connectivity constraints imposed between pair-wise robots), such that the modification to the primary controller is minimum for the robotic team, and 2) compute the modified controllers for robots

task execution. In the remaining of this section, we will discuss the formulation of the mentioned constraints in the form of Control Barrier Function (CBF) [28, 40, 55] and Finite-Time Convergence Control Barrier Function (FCBF) [78] on the controllers followed by the optimization problem formulation.

Safety Constraints using Safety Barrier Certificates

During movements of multi-robot systems, the robots should avoid collisions with each other to remain safe. Consider the joint robot states $\mathbf{x} = \{x_1, \ldots, x_n\} \in \mathbb{R}^{2n}$ and define the minimum safe distance as R_s for any pair-wise inter-robot collision avoidance constraint. We have the following condition defining the safe set of $\mathbf{x}$.

$$\begin{aligned} h^s_{i,j}(\mathbf{x}) &= \|x_i - x_j\|^2 - R_s^2, \qquad \forall i > j \\ \mathcal{H}^s_{i,j} &= \{\mathbf{x} \in \mathbb{R}^{2n} : h^s_{i,j}(\mathbf{x}) \geq 0\} \end{aligned} \tag{6.1}$$

The set of $\mathcal{H}^s_{i,j}$ indicates the safety set from which robot i and j will never collide. For the entire robotic team, the safety set can be composed as follows.

$$\mathcal{H}^s = \bigcap_{\{v_i, v_j \in \mathcal{V}: i > j\}} \mathcal{H}^s_{i,j} \tag{6.2}$$

[29, 51] proposed the safety barrier certificates $\mathcal{B}^s(\mathbf{x})$ using control barrier functions (CBF) [55] that map the constrained safety set Eq. (6.2) of $\mathbf{x}$ to the admissible joint control space $\mathbf{u} \in \mathbb{R}^{2n}$. The result is summarized as follows.

$$\mathcal{B}^s(\mathbf{x}) = \{\mathbf{u} \in \mathbb{R}^{2n} : \dot{h}^s_{i,j}(\mathbf{x}) + \gamma h^s_{i,j}(\mathbf{x}) \geq 0, \forall i > j\} \tag{6.3}$$

where γ is a user-defined parameter to confine the available sets. It is proven in [51] that the forward invariance of the safety set $\mathcal{H}^s$ is ensured as long as the joint control input $\mathbf{u}$ stays in set $\mathcal{B}^s(\mathbf{x})$. In other words, the robots will always stay safe if they are initially inter-robot collision free and the control input lies in the set $\mathcal{B}^s(\mathbf{x})$. Note that at any time point t with known current robot states $\mathbf{x}(t)$, the constrained control space in Eq. (6.3) corresponds to a class of linear constraints over pair-wise control inputs u_i and u_j for $\forall i > j$. Note that static obstacles may also be modelled in the same manner if treated as robots with zero velocity.

Connectivity Constraints using Finite-Time Control Barrier Function

Similar to safety barrier certificates for collision avoidance, pairwise connectivity constraints can also be mapped to the admissible set for control input in the same manner [40]. However, the forward invariance from CBF requires the system already in the desired set, e.g. robots are initially collision free and so to stay safe.

To enforce connectivity constraints used to form new edges, [78] proposed the Finite-Time Convergence Control Barrier Functions (FCBF) that could drive the robots from outside to the admissible set and stay inside the desired states. This has been applied to form and then preserve new connectivity edges predefined by the tasks [78]. Here we briefly introduce the mapping from a particular pairwise connectivity constraint to the admissible set for controllers using FCBF.

To enforce a connectivity constraint between pair-wise robots i and j to limit the inter-robot distance not larger than communication range R_c, we have the following condition.

$$\begin{aligned} h^c_{i,j}(\mathbf{x}) &= R_c^2 - \|x_i - x_j\|^2 \\ \mathcal{H}^c_{i,j} &= \{\mathbf{x} \in \mathbb{R}^{2n} : h^c_{i,j}(\mathbf{x}) \geq 0\} \end{aligned} \tag{6.4}$$

The set of $\mathcal{H}^c_{i,j}$ indicates the feasible set on $\mathbf{x}$ from which robot i and j will never lose connectivity. Then for any connectivity graph $\mathcal{G}^c = (\mathcal{V}, \mathcal{E}^c)$ to enforce, the corresponding constrained set can be composed as follows.

$$\mathcal{H}^c(\mathcal{G}^c) = \bigcap_{\{v_i, v_j \in \mathcal{V} : (v_i, v_j) \in \mathcal{E}^c\}} \mathcal{H}^c_{i,j} \tag{6.5}$$

The connectivity barrier certificates are hence defined as follows using FCBF [78] that indicates another class of linear constraints over pair-wise control inputs u_i and u_j for $(v_i, v_j) \in \mathcal{E}^c$ at any time point t.

$$\begin{aligned} \mathcal{B}^c(\mathbf{x}, \mathcal{G}^c) = \{\mathbf{u} \in \mathbb{R}^{2n} : &\dot{h}^c_{i,j}(\mathbf{x}) + \gamma \cdot \text{sign}(h^c_{i,j}(\mathbf{x})) \cdot |h^c_{i,j}(\mathbf{x})|^\rho \geq 0, \\ &\forall (v_i, v_j) \in \mathcal{E}^c\} \end{aligned} \tag{6.6}$$

where $\rho \in [0, 1)$ determines how fast the system is driven towards the set of $\mathcal{H}^c_{i,j}$. It has been proved in [78] that for any initial condition $\mathbf{x}_0$, any controller subject to Eq. (6.6) will drive the system to the set $\mathcal{H}^c$ in a finite time bounded by $T = \frac{|h^c_{i,j}(\mathbf{x})|^{1-\rho}}{\gamma(1-\rho)}$. This property ensures that for any pairwise connectivity constraint that is not currently satisfied, we can allocate a time period larger than T for the constraint to render the new connectivity edges. Note that the FCBF takes as inputs a given graph $\mathcal{G}^c$ which is predefined in [78]. We will use this sub-routine to enforce the construction of desired $k-$CMRG in our resilient connectivity maintenance framework in the following.

Objective Function

Consider that a task-related primary behavior control input $\hat{u}_i \in \mathbb{R}^2$ has been computed for each robot i before considering the mentioned constraints. The objective

is to minimally modify the primary controllers subject to connectivity and safety constraints. Different from other optimization-based framework with CBF [40] or FCBF [78] with predefined connectivity constraints, here we extend to the resilient connectivity maintenance framework, where the robots are optimizing both the $k-$connectivity constraints to enforce and the controllers to revise. With the defined forms of constraints in Eq. (6.3) and Eq. (6.6), we formally define the *minimally disruptive resilient $k-$connectivity maintenance* problem with any given $k \leq n-1$ at each time point t as follows.

$$\mathbf{u}^* = \underset{\mathcal{G}^c, \mathbf{u}}{\arg\min} \sum_{i=1}^{n} \|u_i - \hat{u}_i\|^2 \tag{6.7}$$

$$\text{s.t.} \quad \mathcal{G}^c = (\mathcal{V}, \mathcal{E}^c) \quad \text{is } k-\text{connected} \tag{6.8}$$

$$\mathbf{u} \in \mathcal{B}^s(\mathbf{x}) \bigcap \mathcal{B}^c(\mathbf{x}, \mathcal{G}^c), \quad \|u_i\| \leq \alpha_i, \forall i = 1, \ldots, n \tag{6.9}$$

The above Quadratic Programming (QP) optimization problem is to find the optimal active $k-$connectivity graph $\mathcal{G}^c$ to enhance and the revised control inputs $\mathbf{u}^* \in \mathbb{R}^{2n}$ bounded by maximum velocity α_i for each robot, so that $k-$connectivity, safety and velocity constraints described in Eq. (6.8) and Eq. (6.9) are always guaranteed while ensuring minimally disruption to the primary controller as shown in Eq. (6.7). While the robust connectivity maintenance problem [16] has similar formulation, it requires $\mathcal{G}^c \subseteq \mathcal{G}$ and hence can only preserve the current connectivity and the subgraph from the existing graph. Here we relax this assumption and allow connectivity enhancement with any desired connectivity k, thus making [16] a special case in our formulation when the current connectivity of $\mathcal{G}$ is larger than desired connectivity k. Note that as information regarding the primary task is not required other than $\hat{u}_i$, the objective of the original controller may not be guaranteed in form of Eq. (6.7) especially when it conflicts with connectivity or safety constraints, e.g. dispersing robots to different goal locations where robots get disconnected due to limited communication range. In this case, the objective of Eq. (6.7) first ensures constraints are satisfied at all time and then minimizes the deviation from original controller, e.g. dispersing robots towards assigned goal locations as much as possible while keeping them safe and k-connected.

The optimization problem in Eq. (6.7) can be decoupled into two dependent sub-problems: 1) compute provably optimal $k-$CMRG graph $\mathcal{G}^{c*} = \mathcal{G}_k^*$ that invokes *least violated* connectivity constraints over multi-robot behaviors, and then 2) solve the optimization problem Eq. (6.7) with the obtained optimal graph $\mathcal{G}_k^*$. In this way, it enables the robot team to form connectivity enhancement provably satisfying any demanded connectivity k while minimizing the disruption to their original tasks.

6.2 Maintaining Minimally Disruptive Resilient $k-$Connectivity

6.2.1 Min-Size $k-$Node Connected Spanning Subgraph ($k-$NCSS)

We consider the first sub-problem of computing optimal $k-$CMRG $\mathcal{G}^{c*} = \mathcal{G}^*_k(\mathcal{V}, \mathcal{E}^*_k)$ in Eq. (6.7) that introduces minimum $k-$connectivity constraints for any given connectivity demand k. Recall that each edge $(v_i, v_j) \in \mathcal{E}^c$ in a candidate graph $\mathcal{G}^c$ enforces one pair-wise linear constraint over primary control inputs $\hat{u}_i$ and $\hat{u}_j$ for robot i and j, as shown in Eq. (6.4). Thus it is straightforward that optimal graph $\mathcal{G}^{c*}$ should have minimum number of edges that satisfy k-connectivity.

Denote the connectivity of a graph by $\kappa(\cdot)$. Let us first consider the special case when $\kappa(\mathcal{G}) \geq k$, and then the $k-$CMRG boils down to finding a min-size $k-$Node Connected Spanning Subgraph ($k-$NCSS) with $\mathcal{G}^{c*} \subseteq \mathcal{G}$. This has been known as NP-hard for even $k = 2$ [118]. From graph theory, there exists a heuristic algorithmic framework, $k-$Node Connected Spanning Subgraph ($k-$NCSS) [117, 118] that finds the approximate min-size k-connected subgraph with uniform edge cost. Briefly, given an undirected connected graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ and k where $k \leq \kappa(\mathcal{G})$, the min-size $k-$connected spanning subgraph $\mathcal{G}^*_k$ can be found by the following summarized algorithm.

Algorithm 5 Minimum-size $k-$node connected spanning subgraph ($k-$NCSS)

Input: $\mathcal{G}(\mathcal{V}, \mathcal{E})$, k
Output: $\mathcal{G}^*_k$

1: find a min-size $k-1$ edge cover $M \leftarrow \arg\min\{|M| : \deg_M(v) \geq k-1, \forall v \in \mathcal{V}, M \subseteq \mathcal{E}\}$
2: find an inclusionwise minimal edge set $F \subseteq \mathcal{E} \setminus M$ such that $(\mathcal{V}, M \cup F)$ is $k-$connected
3: **return** $\mathcal{G}^*_k \leftarrow (\mathcal{V}, M \cup F)$

With the Algorithm 5, we have the following Lemma regarding its known approximation of the derived $k-$connected spanning subgraph $\mathcal{G}^*_k$.

Lemma 21. *([117, 118]) Let $\mathcal{G}(\mathcal{V}, \mathcal{E})$ be a graph of node connectivity $\geq k$. Then the Algorithm 5 finds a $k-$node connected spanning subgraph $(\mathcal{V}, M \cup F)$ such that $|M \cup F| \leq (1 + \frac{1}{k})|\mathcal{E}_{opt}|$, where $|\mathcal{E}_{opt}|$ denotes the cardinality of the optimal solution.*

Hence Algorithm 5 provides a bounded solution to find a $k-$NCSS $\mathcal{G}^*_k \subseteq \mathcal{G}$ with minimum number of edges that could be used to define active pairwise $k-$connectivity constraints when $\kappa(\mathcal{G}) \geq k$. However, such solution could not

handle the situation when $\kappa(\mathcal{G}) \leq k$ and it is more desirable to consider each edge differently due to their impact over the robots original controllers. For example, candidate connectivity constraint whose two robots are getting closer due to their original motion should be preferred, since maintaining such constraint will lead to less disruption over the original robot controllers. In the next section, we will propose a novel $k-$CMRG method to construct the optimal $k-$connected graph with any demanded connectivity $k \leq n-1$ and with consideration of the original robot controllers/motions.

6.2.2 $k-$Connected Minimum Resilient Graph ($k-$CMRG)

In general cases with arbitrary demanded connectivity k, each pairwise robots within the robotic team compose one candidate edge for determining $k-$CMRG $\mathcal{G}^{c*}$, which further increases the computation complexity of computing the optimal $k-$CMRG. Here we propose a new heuristic to evaluate any given candidate edge connecting pairwise robots $v_i, v_j \in \mathcal{V}$ as follows.

$$w_{i,j} = -\dot{h}^c_{i,j}(\mathbf{x}, \hat{u}_i, \hat{u}_j) - \gamma \cdot \text{sign}(h^c_{i,j}(\mathbf{x})) \cdot |h^c_{i,j}(\mathbf{x})|^\rho \tag{6.10}$$

This heuristic takes inspiration from the FCBF constraint in Eq. (6.6) and substitute with the original robot controllers $\hat{u}_i, \hat{u}_j$. Note that the smaller value of $w_{i,j}$ indicates forming/preserving the connectivity edges between v_i, v_j is less likely to be violated given the robot original controller. For example, $w_{i,j} < 0$ implies the FCBF constraint for preserving the corresponding edge is already satisfied by the original robot controllers without need of revision. Instead of checking for each pairwise candidate edges between any two robots in S, we augment the current connectivity graph with their weight defined by Eq. (6.10) and render a weighted connectivity graph $\hat{\mathcal{G}} = (\mathcal{V}, \mathcal{E}, \mathcal{W})$ with $w_{i,j} \in \mathcal{W}$. Next, we propose the following Algorithm 6 framework of our $k-$CMRG, a variant of Algorithm 5 with any connectivity demands k. For the rest of the chapter, we use $k-$CMRG interchangeably to refer to the optimal $k-$connectivity graph or the algorithm to compute $k-$CMRG.

In Algorithm 6, there are several modifications compared to Algorithm 5. In Line 1 of Algorithm 6, it directly augmented 2-hop edges to the existing graph so that the minimum degree $deg(v)$ of each robot node is at least k. The reason lies in that for a $k-$connected graph, each robot has at least k edges and any one node will never be isolated with the removal of at most $k-1$ neighboring nodes.

In Line 2 of Algorithm 6, we redefine the min-size $(k-1)$ edge cover from Algorithm 5 to be M' by the following.

$$M' = \underset{M' \subseteq \mathcal{E}'}{\arg\min} \quad \beta \cdot |M'| + \Sigma_{(v_i,v_j) \in M'} \{w_{i,j}\} \tag{6.11}$$

Algorithm 6 Outline of $k-$connected minimum resilient graph ($k-$CMRG)

Input: $\hat{\mathcal{G}}'(\mathcal{V}', \mathcal{E}', \mathcal{W}') \leftarrow \hat{\mathcal{G}}(\mathcal{V}, \mathcal{E}, \mathcal{W})$, k
Output: $\hat{\mathcal{G}}_k^*$

1: Expand $\mathcal{G}'$ by adding edges connecting robots with 2-hop neighbors until $\min\{\deg(v)\} \geq k$
2: find a min-size $k-1$ edge cover $M' = \arg\min_{M' \subseteq \mathcal{E}'} \quad \beta \cdot |M'| + \Sigma_{(v_i,v_j)\in M'}\{w_{i,j}\}$
3: find an inclusionwise minimal edge set $F' \subseteq \mathcal{E}' \setminus M'$ such that $(\mathcal{V}, M' \cup F')$ is $k-$connected, if not, expand $\hat{\mathcal{G}}'$ by adding edges connecting robots with 2-hop neighbors until F stay unchanged.
4: **return** $\hat{\mathcal{G}}_k^* \leftarrow (\mathcal{V}, M' \cup F')$

where β is a pre-defined parameter and we assume $\beta >> 2 \cdot \Sigma_{\forall w_{i,j} \in \mathcal{W}'} |w_{i,j}|$, so that the selected edge cover set M' has minimum number of edges. And if there are multiple solutions with same number of edges, it will break ties by comparing the total weights and then select the one with minimum total weights. This implies least constrained edges to preserve with the original robot controllers. In the end (line 3), the inclusionwise minimal edge set is found by iterative expanding graph $\hat{\mathcal{G}}'$ until $\kappa(\hat{\mathcal{G}}') \geq k$. With the new condition above for finding $(k-1)$ edge cover set M', a new weighted $k-$connected minimum resilient graph ($k-$CMRG) can be derived as $\hat{\mathcal{G}}_k^* = (\mathcal{V}, \mathcal{E}_k', \mathcal{W}_k')$ with $\mathcal{E}_k' = M' \cup F' \subseteq \mathcal{E}'$. In particular, we have the following Theorem on bounded cardinality of edge set $\mathcal{E}_k'$ of the $k-$CMRG $\hat{\mathcal{G}}_k^*$.

Theorem 22. *Given weighted undirected graph $\hat{\mathcal{G}} = (\mathcal{V}, \mathcal{E}, \mathcal{W})$ and the demanded augmented connectivity k. Then the Algorithm 6 with redefined condition Eq. (6.11) finds the $k-$CMRG $\hat{\mathcal{G}}_k^* = (\mathcal{V}, \mathcal{E}_k', \mathcal{W}_k')$ such that $|\mathcal{E}_k'| \leq (1 + \frac{1}{k})|\mathcal{E}_{opt}'|$, where $\mathcal{E}_{opt}'$ denotes the cardinality of the optimal solution required for such k, as in **Lemma** 21.*

Proof. We first prove that the solution $\hat{\mathcal{G}}_k^* = (\mathcal{V}, M' \cup F')$ from Algorithm 6 with Eq. (6.11) and $\mathcal{G}_k^* = (\mathcal{V}, M \cup F)$ from original Algorithm 5 have the same number of edges, if Algorithm 5 takes as inputs the expanded graph $\mathcal{G}'$ from Algorithm 6 (both satisfy $\kappa(\hat{\mathcal{G}}') \geq k$ after graph expansion). By contradiction, we assume they have different number of edges in M' and M, namely, the following two conditions must be true at the same time.

$$\begin{aligned} \beta \cdot |M'| + \Sigma_{(v_i,v_j)\in M'}\{w_{i,j}\} &< \beta \cdot |M| + \Sigma_{(v_i,v_j)\in M}\{w_{i,j}\} \\ |M'| &> |M| \end{aligned} \tag{6.12}$$

Recall that $\beta >> 2 \cdot \Sigma_{\forall w_{i,j} \in \mathcal{W}} |w_{i,j}|$, hence it is straightforward that the two equations contradicts to each other, proving that $|M'| = |M|$. Then since the computation of the inclusionwise minimal edge set is the same in both of the algorithms, we conclude that $|\mathcal{E}_k'| \leq (1 + \frac{1}{k})|\mathcal{E}_{opt}|$. □

With the minimum number of edges and total weights for the obtained $k-$CMRG, it thus invokes the least $k-$connectivity constraints that are minimally violated by the current behavior-prescribed robots controllers. The resulting $\hat{\mathcal{G}}_k^*$ therefore specifies the optimal $k-$connectivity graph $\mathcal{G}^{c*} = \hat{\mathcal{G}}_k^*$ for the given connectivity demand k to enforce in the optimization problem Eq. (6.8). For completeness, we provide a detailed algorithm framework of our $k-$CMRG method from Algorithm 6 in Algorithm 7.

Algorithm 7 $k-$Connected Minimum Resilient Graph ($k-$CMRG)

Input: $\hat{\mathcal{G}}'(\mathcal{V}', \mathcal{E}', \mathcal{W}') \leftarrow \hat{\mathcal{G}} = (\mathcal{V}, \mathcal{E}, \mathcal{W})$, k
Output: $\hat{\mathcal{G}}_k^*$

1: **while** $\min\{\deg(v)\} < k, \forall v \in \mathcal{V}$ **do**
2: $\quad\hat{\mathcal{G}}' \leftarrow$ExpandGraphOneHopNeighbor($\hat{\mathcal{G}}'$)
3: **end while**
4: **for** all $v \in \mathcal{V}$ **do** $b(v) \leftarrow \deg(v) + 1 - k$
5: **end for**
6: Get $b-$matching edge set: $\bar{M}' \leftarrow b-$Suitor($\hat{\mathcal{G}}', b$)
7: **while** $F' \neq \emptyset$ **do**
8: $\quad M' \leftarrow \hat{\mathcal{G}}' \setminus \bar{M}',\ F' \leftarrow \emptyset,\ \mathcal{G}_t \leftarrow \hat{\mathcal{G}}'$
9: $\quad$**for** all $e \in \bar{M}'$ **do**
10: $\quad\quad\mathcal{G}_t' \leftarrow$ CreateDigraph($\mathcal{G}_t$, unit capacities)
11: $\quad\quad$num_disjoint_path $\leftarrow$ max_flow($\mathcal{G}_t', e_{source}, e_{sink}$)
12: $\quad\quad$**if** num_disjoint_path$> k$ **then**
13: $\quad\quad\quad\mathcal{G}_t$.remove($e$)
14: $\quad\quad$**else**
15: $\quad\quad\quad F' \leftarrow F' \cup e$
16: $\quad\quad$**end if**
17: $\quad$**end for**
18: $\quad\hat{\mathcal{G}}' \leftarrow$ExpandGraphOneHopNeighbor($\hat{\mathcal{G}}'$)
19: **end while**
20: **return** $\hat{\mathcal{G}}_k^* \leftarrow (\mathcal{V}, M' \cup F')$

From Line 1-6 in Algorithm 7, the min-size $(k-1)$ edge cover M' in Eq. (6.11) is obtained by first solving for its complementary edge set $\bar{M}'$ with the following condition.

$$\begin{aligned} \bar{M}' = \underset{\bar{M}' \subseteq \mathcal{E}}{\arg\max} \quad & \beta \cdot |M'| + \Sigma_{(v_i, v_j) \in M'}\{w_{i,j}\} \\ \text{s.t.} \quad & \deg_{\bar{M}'}(v) \leq \deg(v) + 1 - k \quad \forall v \in \mathcal{V} \end{aligned} \tag{6.13}$$

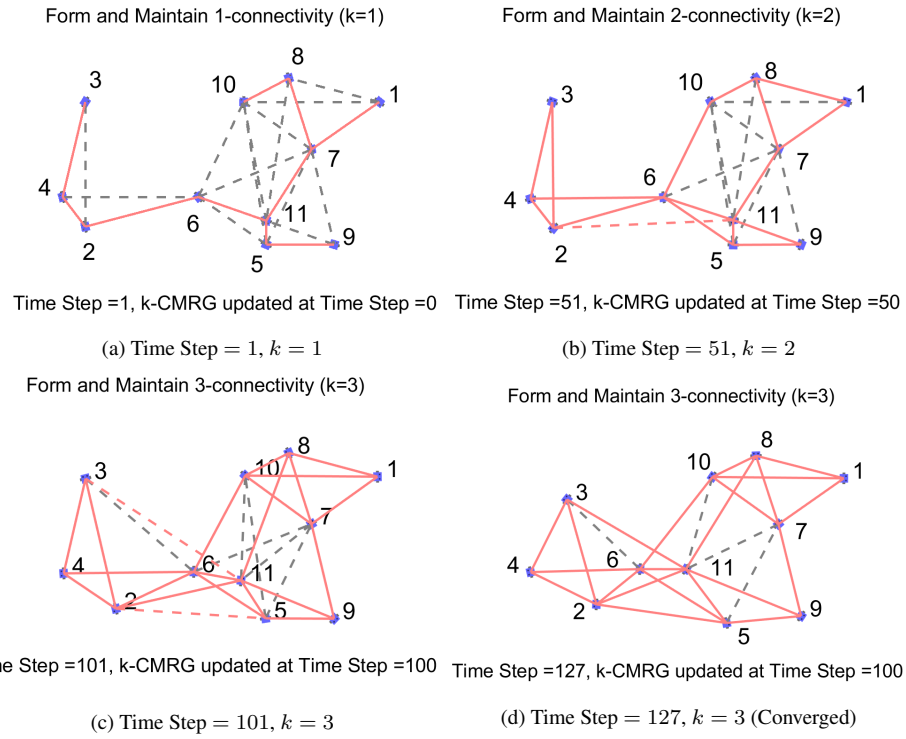
The above problem is known as a weighted $b-$matching problem [117, 118] and

we implement a subroutine $b-$Suitor [119] to solve it efficiently. When computing for the inclusionwise minimal edge set F' in Line 7-18, we start with empty set F' and initialize the current graph to be the present connectivity graph $\hat{\mathcal{G}}$. Then each candidate edge e not in the $k-1$ edge cover set M' is checked by finding if there are at least $(k+1)$-node disjoint paths in the current graph $\mathcal{G}_t$. This is done by creating a directed graph from $\mathcal{G}_t$ and run a max flow algorithm (Line 10-11) using sub-routine from [120]. If yes, then the current candidate edge e is not critical (see [118]) and hence removed from current graph. Otherwise, the edge is critical and shall be inserted into the set F' to consist of final $k-$CMRG $\hat{\mathcal{G}}_k^*$. This comes from the fact that for an optimal $k-$CMRG with least number of edges, each edge is critical and there will be no more than $k+1$ disjoint paths between the two end nodes for the edge [118]. As mentioned, in case that $\kappa(\hat{\mathcal{G}}') \leq k$, we keep looping from Line 7 to Line 18 and expanding the current graph, until no more critical edges are found.

Thus, with the final $k-$CMRG $\hat{\mathcal{G}}_k^*$ obtained from our Algorithm 7 as the optimal $k-$connectivity graph $\mathcal{G}^{c*} = \hat{\mathcal{G}}_k^*$ in Eq. (6.9), we can specify the safety and connectivity barrier certificates Eq. (6.3) and Eq. (6.6) to invoke linear constraints and efficiently solve the original quadratic programming (QP) problem in Eq. (6.7). The resultant controllers satisfy safety and $k-$connectivity constraints and minimally disrupted from the original controllers.

6.3 Results

To evaluate our proposed $k-$CMRG and the resilient connectivity maintenance framework, we designed three sets of experiments in simulation: i) $N = 11$ robots driven by uniform original task controller $\hat{u}_i = 0$ and to keep re-configuring for achieving the increased connectivity demands over time, ii) $N = 20$ robots driven by the same task controller $\hat{u}_i = 0$ with desired connectivity maintenance in presence of continuous loss of robots, and iii) $N = 20$ robots tasked to perform rendezvous and dynamic circling formation around three predefined task areas, while achieving dynamic connectivity demands and staying resilient in presence of removal of robots due to failures. In all of the experiments, we are assuming limited sensing for collision avoidance, limited communication range, and bounded velocity for the robots. We apply the resilient optimization-based controller in Eq. (6.7) with single-integrator dynamics to the unicycle mobile robots using kinematics mapping in [29].



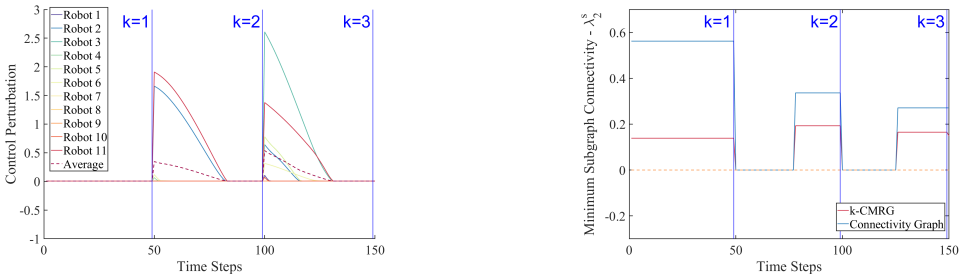
(a) Time Step $= 1$, $k = 1$

(b) Time Step $= 51$, $k = 2$

(c) Time Step $= 101$, $k = 3$

(d) Time Step $= 127$, $k = 3$ (Converged)

Figure 6.2: Simulation example of 11 robots reconfigure to achieve increasing connectivity demands. Grey dash edges are real-time connectivity edges when the connected pairwise robots stay within the limited communication range. Red solid edges are the computed $k-$CMRG edges exist in the current connectivity graph. Red dashed edges are the edges of $k-$CMRG to form (not belong to the current grey connectivity graph) and thus to reach the desired connectivity.



(a) Control perterbation

(b) Subgraph Algebraic Connectivity

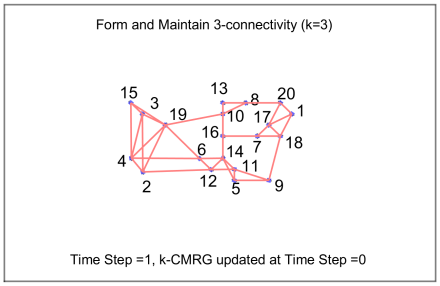
Figure 6.3: Performance of resilient connectivity maintenance with $k-$CMRG for Fig. 6.2. (a) Control perturbation computed by $\frac{1}{n}\sum_{i=1}^{N}\|u_i^* - \hat{u}_i\|^2$. (b) Minimum subgraph algebraic connectivity evaluated by second smallest eigenvalue of laplacian assuming $k-1$ robots being taken out. > 0 means graph remain connected.

Reconfiguration of static robot team with increasing connectivity demands

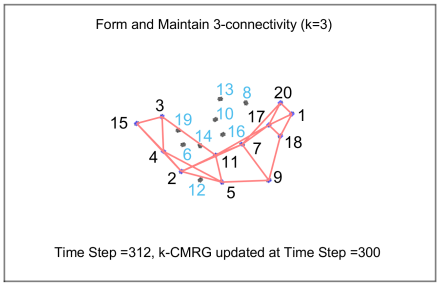
Fig. 6.2 shows the simulation example of 11 robots with zero task-related control inputs and our $k-$CMRG method for connectivity enhancement. At initial configuration Fig. 6.2a, the robots are tasked to maintain $1-$connectivity and the $k-$CMRG returns the minimum spanning tree invoking the least number of connectivity constraints with the smallest weights, reflecting the minimum efforts required to maintain the connectivity. At time step $t = 50$ in Fig. 6.2b, the connectivity demand increases to $k = 2$ that is higher than the current graph connectivity. In this case, our $k-$CMRG returns a $2-$connectivity graph with one new edges between robot 2 and 11 and with such specified constraint, our resilient connectivity control framework employs FCBF to drive the robots to form the connectivity edge as shown in Fig. 6.2c at $t = 101$. Likewise, the new demand of $3-$connectivity invokes two more edges to form, which enforces the robots to reconfigure and quickly converge to the states with satisfying connectivity (Fig. 6.2d). The performance of the maintained connectivity is plotted in Fig. 6.3, showing the convergence of the robots after reconfiguration. In absence of actual robot removal, Fig. 6.3b demonstrates the algebraic connectivity of the subgraph of current connectivity graph if randomly taking out $k - 1$ robots. As seen from the figure for $t = 50 - 60$, multi-robot network will get disconnected if removing one robot from the currently 1-connected graph. After achieving the connectivity $k = 2$ at $t = 60$, the resilience of the network is improved and hence robust to the removal of the robots. Note that the robots remain collision-free due to the employed safety barrier certificates in Eq. (6.3).

Connectivity Enhancement in presence of continuous loss of robots

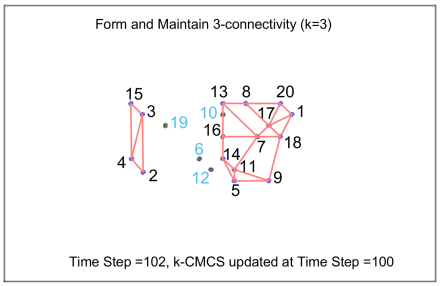
One of the advantage of the proposed $k-$CMRG is to enable increased connectivity over time for resilient robot team under faulty situation, as shown in Fig. 6.4. In this experiment, 2 randomly selected robots will stop connecting other robots every 50 time steps starting from $t = 50$ (Fig. 6.4d). The robots are tasked to maintain $k = 3$ connectivity in presence of robot losses. With the proposed $k-$CMRG, the connectivity of multi-robot network is preserved and actively recovered even if robots keep failing with a total of 8 failing robots (Fig. 6.4b). In comparison, we implemented the robust connectivity approach [16] that seek to preserve the robust connectivity of the current graph. As shown in Fig. 6.4c and Fig. 6.4d, although the robotic team could stay connected with the removal of a few robots, the robot team is not able to recover the decreased connectivity over time and hence gets disconnected eventually.



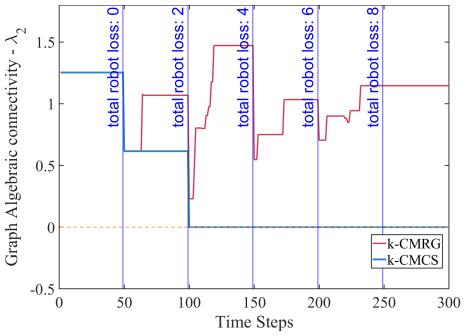
(a) Time Step $= 1$ with $k-$CMRG



(b) Time Step $= 312$ with $k-$CMRG

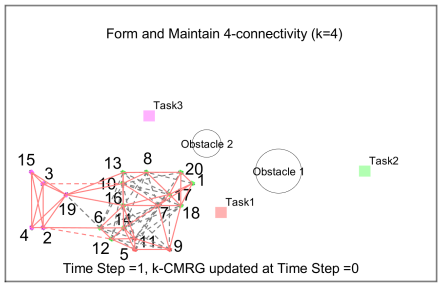


(c) Time Step $= 102$ with $k-$CMCS

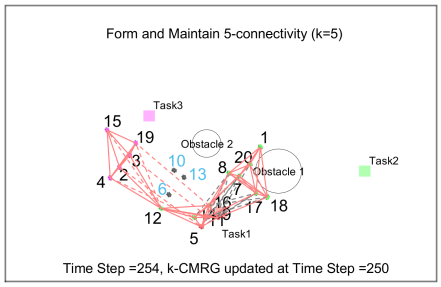


(d) Algebraic Connectivity

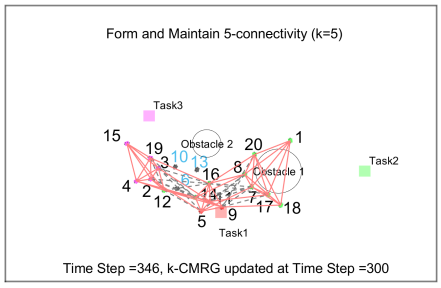
Figure 6.4: Simulation example of 20 initially static robots in presence of continuous robot failures. Red edges from (a)-(b) are defined by computed $k-$CMRG and robots with cyan index indicates the faulty robots that are no longer involved in the connectivity graph. Red edges in (c) shows the failure cases of robust connectivity maintenance method [16] due to the lack of resilience consideration. (d) plots the actual algebraic connectivity λ_2 of the real-time connectivity graph from $k-$CMRG and $k-$CMCS [16]. Connectivity preserves if $\lambda_2 > 0$. It shows $k-$CMRG (red curve) is able to keep the graph connected and recover the connectivity in presence of loss of robots).



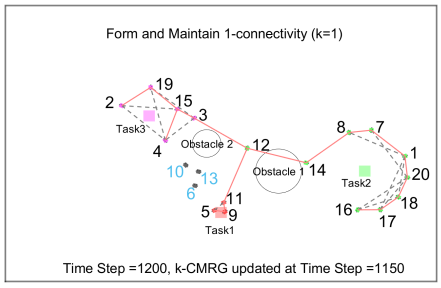
(a) Time Step=1 with $k = 4$



(b) Time Step=254 with $k = 5$

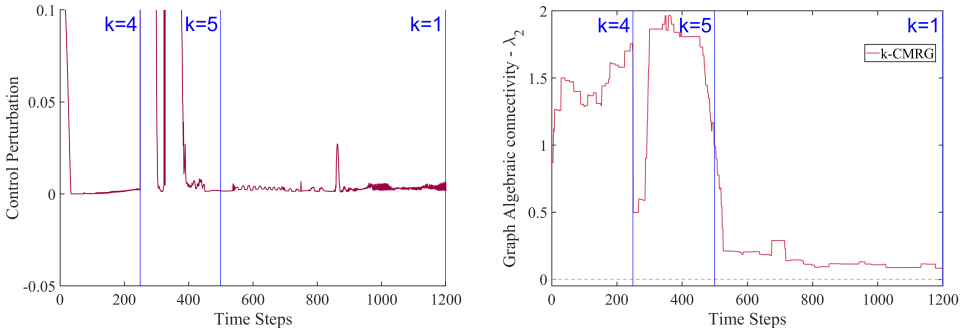


(c) Time Step=346 with $k = 5$



(d) Time Step=1200 with $k = 1$

Figure 6.5: Simulation example of 20 robots executing multiple behaviors with changing connectivity demands and robot failures: green robots 1, 6, 7, 8, 10, 12, 13, 14, 17, 18, 20 and magenta robots 2, 3, 4, 15, 19 are tasked to circle around task 2 and task 3 area respectively. Red robot 5, 9, 11 are tasked to rendezvous towards task 1 area. Grey dash edges are real-time connectivity edges. Red solid edges are the computed $k-$CMRG edges exist in the current connectivity graph. Red dashed edges are the edges of $k-$CMRG to be formed by the robots for increased connectivity.



(a) Control perturbation

(b) Algebraic connectivity

Figure 6.6: Performance of $k-$CMRG from simulation example in Fig. 6.5. Control perturbation quickly converges to zero due to minimally disruptive connectivity maintenance from $k-$CMRG and network connectivity is always satisfied and recovered despite robot failures.

Reconfiguration of moving robot team

In this task, 20 robots have been divided into 3 subgroups and each performing an individual behavior with the $k-$CMRG in presence of 2 static obstacles and robot failures. The connectivity demands are randomly chosen to be $k = 4$ for $t = 0-250$, $k = 5$ for $t = 250-500$ and $k = 1$ for $500-1200$. Robot 6, 10 and 13 are removed at $t = 250$ in Fig. 6.5b to simulate faulty situation. As shown in Fig. 6.5d, the goal for magenta robots and green robots are to circle around assigned task area 3 and 2 respectively, while red robots are to rendezvous to red task 1 area. Without connectivity maintenance, the robot team could get disconnected easily. Robots start from Fig. 6.5a to reconfigure and achieve the higher demand of connectivity $k = 4$ while executing their original behaviors (the control error reduced to almost zero quick after $t = 0$ as shown in Fig. 6.6a). At $t = 250$, the connectivity demand further increases to $k = 5$ and three robots are lost, resulting in largely reduced network connectivity as observed in Fig. 6.6b. The $k-$CMRG is able to quickly reconfigure the robots to reach the desired connectivity as shown in Fig. 6.5c, where all $k-$ CMRG edges are established as solid red edges. Meanwhile, the original behaviors of the robots are preserved as the control perturbation to the original controller reduced to almost zero soon after $t = 400$. And with the decreased connectivity demand to $k = 1$ after $t = 500$, the robots are able to stay as close to their tasks while ensuring the required connectivity in Fig. 6.5d, demonstrating the flexibility of our resilient connectivity maintenance method with $k-$CMRG.

6.4 Conclusion and Discussions

In this chapter, we propose resilient connectivity maintenance algorithms to ensure minimally disruptive connectivity enhancement for multiple robots during the exe-

cution of their primary tasks. In particular, we propose a $k-$Connected Minimum Resilient Graph ($k-$CMRG) to allow for reconfiguration of the multi-robot system to provably achieve any connectivity demand, while ensuring the robot original behaviors are minimally modified in the context of quadratic programming with the control barrier fucntions (CBF) and Finite-Time Convergence Control Barrier Functions (FCBF). Such algorithms improve resilience of connectivity reduction for open robot team with continuous robots arrival and removal. Simulation examples are demonstrated to validate our algorithm in various challenging scenarios.

The increasing of network redundancy by keeping robots closer could indeed improve the robustness and resilience of the robot team against external attacks or internal failures. However, how to determine the proper level of redundant connectivity for the group remains challenging. In our current design, the $k-$connectivity is assumed given by the user and although it could be changed manually overtime, it may not reflect the actual situation. For example, when the risk of losing robots is low, the robots are desired to maintain a relatively low network redundancy and allow themselves to spread out wider as needed by the primary task. And if they run into more risky situation, e.g. increasing loss of robots in a short amount of time, the robots should be able to monitor and detect such changes so that they can adaptively increase the $k-$connectivity by maintaining or forming more inter-robot edges for potential threats. To develop a method for *adaptive* resilient connectivity maintenance is an interesting future direction. On the other hand, finding the $k-$connected minimum resilient graph requires computation and information sharing in a centralized manner. How to decentralize the process to make the computation scalable for distributed multi-robot systems is another important future direction to pursue.

Part III

Resilient Multi-Robot Coordination

Thus far, we have discussed how to ensure safety and resilience in terms of preserving the integrity of the multi-robot system through communication maintenance for effective multi-robot coordination. In many multi-robot applications, the design of cooperative robot control laws depends on not only the evolving state of the robotic team but also the task-related information from the environment model. For example, in environment sampling or monitoring tasks, the robots need to move to the best locations that maximize the task performance, and the optimal locations are often determined by the underlying distribution of environment phenomenon of interest in the environment. However, when robots are deployed to an unknown environment, such information is not known beforehand, and thus learning is necessary for robots to adapt their behaviors over an extended period of time as they model the environment with data collected online. This will provide robots with the capability of resilience in terms of adapting to the unknown and uncertain environment with improved performance over time.

In this part, we will discuss some data-driven multi-robot coordination algorithms in the application of multi-robot sensor coverage task as an example to demonstrate the learning-enabled resilient coordination in an unknown environment.

Chapter 7

Learning-enabled Multi-Robot Sensor Coverage in Unknown Environment

We consider the problem of online environmental sampling and modeling for multi-robot sensor coverage shown in Fig. 7.1, where a group of networked robots are deployed in an environment from given starting configurations and move to the final optimal placements such that the overall sensing performance over the environmental phenomenon (e.g. temperature) from those particular locations is maximized, also known as the *Locational Optimization* problem [121]. However, when the environment is initially unknown and the robots have no prior knowledge of the distribution of temperature, namely the density function, then they have to learn such distribution of interests first by exploring in the environment and taking samples, which could be time-consuming. To that end, it is desired for the robots to take as few samples for efficient environmental modeling while approaching to its estimated optimal sensing locations.

In this chapter, we proposes a data-driven coordination framework that enables the robots to efficiently learn a model of the unknown density function on-line using adaptive sampling and non-parametric inference such as Gaussian Process (GP). To capture significantly different components of the environmental phenomenon, we propose a new approach with mixture of locally learned GPs for collective model learning and an information-theoretic criterion for adaptive sampling in multi-robot coverage that balances between exploration (i.e. moving to locations with most informative samples) and exploitation (i.e. moving to locations with best predicted sensor coverage performance). The resulting GP mixture model provides improved prediction accuracy and reduced model uncertainty

for complex distributions with significantly distinctive components, and hence increases the multi-robot coverage performance. Moreover, we propose a distributed consensus learning algorithm for the mixture of GPs with local data only. At each round of sampling, each robot first employs the Gaussian Mixture Model (GMM) to classify its collected samples and extract the local GMM parameters. With the distributed consensus learning algorithm, the consensus on global GMM parameters is reached that best classifies the local data for each robot to locally fit a mixture of GPs and predict the density function on its own. In this way, only local GMM parameters are exchanged among robots whose size is independent from the number of collected samples, and hence we avoid the transmission of all local data from every robot. Then the locally learned density function is used to construct the decentralized adaptive coverage controller with the information-theoretic criterion for adaptive sampling that drives each robot towards its updated estimated optimal location, which could reduce the actual sensing cost (increase sensing performance) and also the model uncertainty for the learned density function with new samples collected on the location.

The main *contribution* of our novel approach is three-fold. First, we couple the adaptive sampling with information-theoretic criterion into the multi-robot coverage control framework for efficient model learning and simultaneous locational optimization with a reduced number of samples in an initially unknown environment. Second, we present a fully distributed algorithm that allows for collaboratively learning the generalized non-parametric mixture of GPs model of density function with local data only. This could also be very useful for other decentralized data-driven multi-robot adaptive sampling and informative sampling tasks as most literature still assume the transmission of global data from all the robots for learning and evaluating the environment phenomenon, which is not scalable and may not be practical. Third, extensive empirical results are provided using real-world dataset including the agricultural field data collected by agricultural robot [122] and a public dataset [123] from Intel Berkeley Research Lab to demonstrate the superior performance of our approach.

7.1 Multi-Robot Sensor Coverage Problem

Consider a set of n robots moving in a bounded environment $Q \subset \mathbb{R}^2$ and assume the environment can be discretized into a set of point $q \in Q$, with the position of each robot $i \in \{1, 2, \ldots, n\}$ denoted by $x_i \in Q$. We assume the environment is free of obstacles and can be partitioned into n Voronoi cells, as done in most multi-robot sensor coverage algorithm [4, 44, 84].

$$V_i = \{q \in Q | \|q - x_i\| \leq \|q - x_j\|, \forall j \neq i\} \tag{7.1}$$

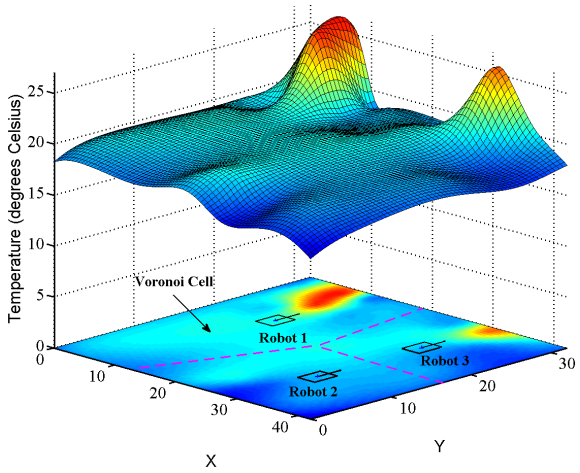


Figure 7.1: Three robots are deployed and navigate to locations that maximize the sensing/coverage performance over environmental phenomenon such as temperature in the map. The upper layer represents the actual temperature distribution interpolated from 54 deployed stationary sensors in the Intel Berkeley Lab [123]. The lower layer represents the 2D multi-robot sensor coverage scenario with projected heat map of the temperature distribution.

where $\|\cdot\|$ is the l^2-norm. Each Voronoi cell V_i corresponds to its generator robot x_i who will be responsible for sensing the points inside the cell $q \in V_i$.

Regarding the distribution of environmental phenomenon on each point of interest q, there exists an unknown density function $\phi(\cdot) : Q \to \mathbb{R}_+$ that maps the location information q to the scalar value of the phenomenon $\phi(q)$. Intuitively, in environmental monitoring task we want each robot to stay close to the area with higher phenomenon value $\phi(\cdot)$ since the sensing performance usually degrades as the distance between the robot and the point to sense increases. As (see (Eq. 7.1)) each point is assigned to one robot, the cost function of static multi-robot coverage can be formally defined as follows [4, 44].

$$\mathcal{H}(x_i, \ldots, x_n) = \sum_{i=1}^{n} \int_{q \in V_i} \|q - x_i\|^2 \phi(q) dq \tag{7.2}$$

Hence the lower $\mathcal{H}(x_i, \ldots, x_n)$ the better. Then by taking the gradient of Eq. (7.2), we have the local optimal solutions for minimizing $\mathcal{H}(\cdot)$ for all $i \in \{1, \ldots, n\}$ as follows.

$$x_i^* = \arg\min \mathcal{H}(x_i, \ldots, x_n) = \frac{\int_{V_i} q\phi(q) dq}{\int_{V_i} \phi(q) dq} = C_{V_i} \tag{7.3}$$

where $C_{V_i} \in \mathbb{R}^2$ is also referred to as the centroid of each Voronoi cell V_i. Although this critical point of $\mathcal{H}$ is a local minimum, due to the intractable solution (NP-hard) to the global optimum $\mathcal{H}$ the local optimal solution x_i^* is often considered optimal

(see [44, 84]). The decentralized gradient-based move-to-centroid controller [4] has been proven to navigate the robots to the local optimal locations.

$$\dot{x_i} = k_p(C_{V_i} - x_i) \tag{7.4}$$

where k_p is a user-defined control gain. Note that the realization of $\phi(q)$ will not be available to the robots unless $q = x_i$ and without loss of generality we ignore the intermediate visited points between consecutive waypoints by the robots. To that end, the objective is to drive the robot towards the locations with high predicted value of the phenomenon and informativeness so as to efficiently learn the distribution $\phi(\cdot)$ while simultaneously optimizing $\mathcal{H}(\cdot)$ with Eq. (7.4). In other words, we will use the optimal controller with the same form as in Eq. (7.4), but with a different specification of C_{V_i}.

7.2 Gaussian Process regression for single robot environment modeling

In this section, we introduce the modeling of density function by a single robot with its locally sampled training data set.

Gaussian Process Regression

A common approach for modeling spatial phenomena is GP regression. Such a natural non-parametric generalization of linear regression allows for modeling the hidden mapping from training data to the target phenomenon with consideration of uncertainty [98]. Assume the target phenomenon, such as temperature in our case, satisfies a multivariate joint Gaussian distribution [89, 124]. The learned GP model from training data outputs the Gaussian probability distribution of the phenomenon $\phi(q)$ specified by mean function $\mu(q) = \mathbb{E}[\phi(q)]$ and covariance function $k(q, q') = \mathbb{E}[(\phi(q) - \mu(q))^T(\phi(q') - \mu(q'))]$ for any query data.

Formally, let $\tilde{V_i} = [q_1^i, \ldots, q_{N_i}^i]^T$ be the set of N_i collected samples associated with observed noisy values of temperature $\mathbf{y_i} = [y_1^i, \ldots, y_{N_i}^i]^T$ by robot i. Each observation is noisy $y = \phi(q) + \epsilon$ with $\epsilon \sim N(0, \sigma_n^2)$ assuming the mean function to be zero without loss of generality. To that end, given a testing location $q_{test} \in Q$, we have the conditional posterior mean $\mu_{q_{test}|\tilde{V}_i,\mathbf{y_i}}$ and variance $\sigma^2_{q_{test}|\tilde{V}_i,\mathbf{y_i}}$ as follows from the learned GP model describing the Gaussian distribu-

tion of $\phi(q_{test}) \sim \mathcal{N}(\mu_{q_{test}|\tilde{V}_i,\mathbf{y_i}}, \sigma^2_{q_{test}|\tilde{V}_i,\mathbf{y_i}})$.

$$\begin{aligned}\mu_{q_{test}|\tilde{V}_i,\mathbf{y_i}} &= \mathbf{k}(q_{test})^T(\mathbf{K}_{\tilde{V}_i} + \sigma_n^2\mathbf{I})^{-1}\mathbf{y_i} \\ \sigma^2_{q_{test}|\tilde{V}_i,\mathbf{y_i}} &= k(q_{test}, q_{test}) - \mathbf{k}(q_{test})^T(\mathbf{K}_{\tilde{V}_i} + \sigma_n^2\mathbf{I})^{-1} \cdot \mathbf{k}(q_{test})\end{aligned} \tag{7.5}$$

where $\mathbf{k}(q_{test}) = [k(q_1^i, q_{test}), \ldots, k(q_{N_i}^i, q_{test})]^T$ with the covariance (kernel) function $k(q, q')$ that captures the correlation between q and q'. $\mathbf{K}_{\tilde{V}_i}$ is the positive definite symmetric kernel matrix $[k(q, q')]_{q,q' \in \tilde{V}_i \cup q_{test}}$. In particular, we use the following squared-exponential kernel function to specify the inter-sample correlation.

$$k(q, q') = \sigma_f^2 e^{-\frac{(q-q')^T(q-q')}{2l^2}} \tag{7.6}$$

where the hyper-parameters l and σ_f are length-scale and scale factor, respectively. Hence, each robot i maintains its own GP model learned from local samples $\{\tilde{V}_i, \mathbf{y_i}\}$ and the hyper-parameters of $(\sigma_n^2, \sigma_f^2, l)$ are optimized from the local training data $\{\tilde{V}_i, \mathbf{y_i}\}$, which will be introduced next.

Estimation of Hyper-Parameters

The GP model of each robot i is determined by its local training data set $\{\tilde{V}_i, \mathbf{y_i}\}$ and local hyper-parameters denoted by $\theta_i = \{\sigma_n^2, \sigma_f^2, l\}$. In particular, the hyper-parameters are desired to be the optimizer such that the kernel function can accurately describe the underlying phenomena. In order to improve computation efficiency, we assume the hyper-parameters for each robot are optimized using the local training data of the robot itself, regardless of the GP mixture process which we discuss in Section 7.3.1. One common approach for learning the hyper-parameters in a Bayesian framework is to maximize the log of the marginal likelihood as follows.

$$\begin{aligned}\theta_i^* &= \arg\max_{\theta_i} \log p(\mathbf{y_i}|\tilde{V}_i, \theta_i) \\ &= -\frac{1}{2}\mathbf{y_i}^T\tilde{\mathbf{K}}_{\tilde{V}_i}^{-1}\mathbf{y_i} - \frac{1}{2}\log|\tilde{\mathbf{K}}_{\tilde{V}_i}| - \frac{N_i}{2}\log 2\pi\end{aligned} \tag{7.7}$$

where $\tilde{\mathbf{K}}_{\tilde{V}_i} = \mathbf{K}_{\tilde{V}_i} + \sigma_n^2\mathbf{I}$. The maximizer of Eq. (7.7) can be computed by taking the partial derivatives of the marginal likelihood $p(\mathbf{y_i}|\tilde{V}_i, \theta_i)$ w.r.t. the hyper-parameters θ_i as described in [125].

7.3 Learning-enabled Coverage Control: Centralized Design

Given the local GP model learned by each robot, in this section, we introduce the centralized learning step of mixture of GPs for every robot, assuming knowledge of all the robots' data available through inter-robot communication, and then compute the corresponding decentralized control and sampling strategy built on the GP mixture model.

7.3.1 Mixture of Gaussian Process Models and Adaptive Sampling Strategy

The mixture of GP models proposed in [96] is a linear combination of multiple GP models. We have a set of locally learned GP models $\{\mathcal{GP}_1, \ldots, \mathcal{GP}_n\}$ from all n robots as aforementioned and denote $P(z(q) = i)$ as the probability of any random point $q \in Q$ being best described by the ith GP model from robot i. Then we have the GP mixture model defined by the conditional posterior mean $\mu^*_{q|\tilde{V},\mathbf{Y}}$ and variance ${\sigma^*}^2_{q|\tilde{V},\mathbf{Y}}$ for any location $q \in Q$ as follows.

$$\begin{aligned} \mu^*_{q|\tilde{V},\mathbf{Y}} &= \sum_{i=1}^{n} P(z(q) = i) \cdot \mu_{q|\tilde{V}_i,\mathbf{y}_i} \\ {\sigma^*}^2_{q|\tilde{V},\mathbf{Y}} &= \sum_{i=1}^{n} P(z(q) = i) \cdot (\sigma^2_{q|\tilde{V}_i,\mathbf{y_i}} + (\mu_{q|\tilde{V}_i,\mathbf{y}_i} - \mu^*_{q|\tilde{V},\mathbf{Y}})^2) \end{aligned} \tag{7.8}$$

where $\{\tilde{V}, \mathbf{Y}\}$ represents the set of collected samples by all the robots with $\tilde{V} = \{\tilde{V}_1, \ldots, \tilde{V}_n\}$ and $\mathbf{Y} = \{\mathbf{y}_1, \ldots, \mathbf{y}_n\}$. To that end, for any point q its actual temperature $\phi(q)$ is assumed to be sampled from the Gaussian distribution $\mathcal{N}(\mu^*_{q|\tilde{V},\mathbf{Y}}, {\sigma^*}^2_{q|\tilde{V},\mathbf{Y}})$. And the common approach for efficient sampling and modeling is to navigate the robots to the point $q^* = \arg\max \mu^*_{q^*|\tilde{V},\mathbf{Y}}$ or $q^* = \arg\max {\sigma^*}^2_{q|\tilde{V},\mathbf{Y}}$ to maximize the sampled value of phenomenon or minimize the prediction uncertainty.

In our problem, we want to simultaneously sample the area with high value of phenomenon to get closer towards the Voronoi centroid C_{V_i} while reducing the uncertainty for the learned model of the density function $\phi(\cdot)$. Here we use the Gaussian Process Upper Confidence Bound (GP-UCB) [126], a sequential stochastic optimization strategy that trades off between exploration (reduce prediction uncertainty) and exploitation (maximize sampled value). Each location q is evaluated

with the information-theoretic criterion defined as follows.

$$h(q) = \mu^*_{q|\tilde{V},\mathbf{Y}} + \beta {\sigma^*}^2_{q|\tilde{V},\mathbf{Y}} \tag{7.9}$$

where β is a parameter relates to the current sampling iteration number and regret bound [126]. When β is specified by a much higher value, then our solution becomes similar to the informative sampling [86] in which we want to reduce the model uncertainty before switching to the static coverage optimization. The GP-UCB strategy works by sequentially sampling point q that maximizes Eq. (7.9) and immediately update the GP model accordingly, such that we will be able to reach a balance by such an adaptive sampling strategy between reducing future GP model uncertainty and maximizing sampled value. However, our primary goal is to minimize the sensing cost function $\mathcal{H}(\cdot)$ in Eq. (7.2) by approaching unknown centroid of Voronoi cell C_{V_i} for each robot i. Thus, we modify the optimal solution in Eq. (7.4) by replacing unknown density function realization with the GP-UCB evaluation Eq. (7.9), which yields our adaptive sampling strategy for each robot i as follows.

$$q_i^* = \frac{\int_{V_i} q h(q) dq}{\int_{V_i} h(q) dq} = \tilde{C}_{V_i} \tag{7.10}$$

And the local coverage control law for each robot i becomes

$$\dot{x_i} = k_p(\tilde{C}_{V_i} - x_i) \tag{7.11}$$

In this case, the robots are able to simultaneously consider density function learning and sensing performance optimization. To solve for the feedback control law Eq. (7.11), it boils down to optimize the mixture of GP model by 1) finding the appropriate weight distribution $P(z(q) = \cdot)$, and 2) modifying local GP model with training data from other robots for generalizing the overall regression model. To simplify our discussion, we assume the robots are always connected as in [93] and are able to share their sampled data by communicating with its direct Voronoi neighbors [86].

7.3.2 GP Mixture Model Learning with Expectation-Maximization (EM) for Prediction

The EM algorithm [96, 127] has been widely used for estimating hidden and observable variables, such as the weight distribution of Gaussian Mixture Models for unsupervised learning. It consists of two stages such as the estimation (E) stage and the maximization (M) stage and it keeps looping until convergence under some

threshold [96]. In our problem, we initialize the probability of weight distribution for any given query data point q_j by setting

$$P(z(q_j) = i) \approx \begin{cases} 1 & \text{if} \quad q_j \in \tilde{V}_i \\ 0 & \text{Otherwise} \end{cases} \quad \forall\, i = 1, \ldots, n \tag{7.12}$$

Then in the E-stage, the algorithm updates the probability $P(z(q_j) = i)$ by computing the marginal likelihood of each data q_j for all GP models. To simplify the notation we use $\mathcal{N}_i(q_j)$ to define the probability of observation of q_j regarding the local GP model $\mathcal{GP}_i$. Then we have the $P(z(q_j) = i)$ update rule over the previous one as follows.

$$P(z(q_j) = i) := \frac{P(z(q_j) = i) \cdot \mathcal{N}_i(q_j)}{\sum_{k=1}^{n} P(z(q_j) = k) \cdot \mathcal{N}_k(q_j)} \tag{7.13}$$

Then in the M-stage, the local GP models will be modified by embedding the updated probability of each query point q_j to the GP model updates steps Eq. (7.5). Here we present the main result for updating model $\mathcal{GP}_i$ from [96, 127] as follows.

$$\begin{aligned} \mu_{q_{test}|\tilde{V}_i,\mathbf{y_i}} &= \mathbf{k}(q_{test})^T (\mathbf{K}_{\tilde{V}_i} + \Psi^i \mathbf{I})^{-1} \mathbf{y_i} \\ \sigma^2_{q_{test}|\tilde{V}_i,\mathbf{y_i}} &= k(q_{test}, q_{test}) - \mathbf{k}(q_{test})^T (\mathbf{K}_{\tilde{V}_i} + \Psi^i \mathbf{I})^{-1} \cdot \mathbf{k}(q_{test}) \end{aligned} \tag{7.14}$$

where

$$\Psi^i_{jj} = \frac{\sigma_n^2}{P(z(q_j) = i)} \tag{7.15}$$

It is noted that by modifying the value of diagonal hyper-parameter Ψ^i_{jj} from local value of σ_n^2 the effects of each training data to the local GP models are adjusted so as to account for the observations for the points outside the local traning data set. Once the EM algorithm converges, we will have the new training data set consisting of $\{q_j\}$ and the corresponding weight distribution $P(z(q_j) = i)$ for each updated GP model i. With such training data set and the updated GP model, for any new query data q_j^*, we can predict its corresponding weight distribution $P(z(q_j^*) = i)$ as well as the expected value from local GP models Eq. (7.14), and then feed into the GP mixture model Eq. (7.8) to further yield the updated control law Eq. (7.10)-(7.11) to govern the motion of the robots.

7.4 Learning-enabled Coverage Control: Decentralized Design

In this section, we introduce the distributed consensus learning of mixture of GPs for every robot i and the overall learning and adaptive coverage control algorithm that runs on each robot with the knowledge of only local data. Different from the centralized design in Section 7.3 that requires robots share all of their collected data to do the computation in a centralized manner, the decentralized design will allow the robots to share only model-related parameters in fixed size and perform all computation in a fully distributed manner.

7.4.1 Local Training Data Classification with Distributed Expectation-Maximization (EM)

Here we assume the environmental phenomenon can be described by a set of GP models $\{\mathcal{GP}_1, \ldots, \mathcal{GP}_m\}$ with m as the number of Gaussian components, and denote $P(z(q) = i_g)$ as the probability of any random point $q \in Q$ being best described by the i_gth GP model. $\tilde{V}_i^{i_g} \subset \tilde{V}_i$ is the subset of collected data by robot i that can be best described by the i_gth GP model. Recalling the mixture of GPs in Eq. (7.8) for each robot i to predict $\mu^*_{q|\tilde{V}_i,\mathbf{y}_i}$ and $\sigma^{*2}_{q|\tilde{V}_i,\mathbf{y}_i}$ at any query point q, similarly here it requires the knowledge of 1) classified local data set $\tilde{V}_i = \{\tilde{V}_i^1, \ldots, \tilde{V}_i^m\}$ and 2) *predicted* weight distribution $P(z(q) = i_g)$ for $i_g = 1, \ldots, m$ where $q \notin \tilde{V}_i$, also known as the gating function. To get such information, the EM algorithm consists of two stages such as the estimation (E) stage for computing the weight distribution $P(z(q_j) = i_g)$ where $q_j \in \tilde{V}_i$ and the maximization (M) stage for updating the GP models, and it keeps looping until convergence under some threshold [96]. In order to relax the assumption of global data needed for EM computation in Section 7.3, we propose to first employ Gaussian Mixture Models (GMM) with distributed EM algorithm [97] via peer-to-peer inter-robot communication for local training data classification and computation of weight distribution $P(z(q_j) = i_g)$ for *collected* points $q_j \in \tilde{V}_i$. Then we use the corresponding labeled local data $\tilde{V}_i$ with weight distribution for training the mixture of GPs, which will be described in our distributed mixture of GPs algorithm in Section 7.4.2.

Although a GP model is an infinite dimension object, the real-world phenomenon can often be characterized by a finite number of Gaussian components [86, 127]. Here we assume the value of environmental phenomenon $\phi(q)$ is drawn from m Gaussian components (corresponding to m GP models) with each component described by a set of unknown model parameters $\Theta_{i_g} = \{\alpha_{i_g}, \mu_{i_g}, \Sigma_{i_g}, i_g =$

$1, \ldots, m\}$ where α_{i_g} is the probability of $\phi(q)$ drawn from the i_gth Gaussian component $\mathcal{N}_{i_g}(\mu_{i_g}, \Sigma_{i_g})$. Then we can rewrite the global summary quantities for each Gaussian component $\mathcal{N}_{i_g}$ as follows.

$$\begin{aligned} \alpha_{i_g} &= \frac{1}{n}\sum_{i=1}^{n} |\mathbf{y}_i| \alpha_{i,i_g}, \quad \lambda_{i_g} = \frac{1}{n}\sum_{i=1}^{n} |\mathbf{y}_i| \lambda_{i,i_g} \\ \gamma_{i_g} &= \frac{1}{n}\sum_{i=1}^{n} |\mathbf{y}_i| \gamma_{i,i_g}, \quad \mu_{i_g} = \frac{\lambda_{i_g}}{\alpha_{i_g}}, \quad \Sigma_{i_g} = \frac{\gamma_{i_g}}{\alpha_{i_g}} \end{aligned} \tag{7.16}$$

where $\{\alpha_{i,i_g}, \lambda_{i,i_g}, \gamma_{i,i_g}\}$ are the local summary quantities for each robot i that can be computed locally as follows, given the information of its current *estimated* global model parameter $\tilde{\Theta}_{i,i_g} = \{\tilde{\alpha}_{i,i_g}, \tilde{\mu}_{i,i_g}, \tilde{\Sigma}_{i,i_g}\}$ for $i_g = 1, \ldots, m$ (obtained via dynamic consensus in Eq. (7.18)) and the observed value $\mathbf{y}_i = \{y_j\}, j = 1, \ldots, |\mathbf{y}_i|$ of local data set $\tilde{V}_i$.

$$\begin{aligned} p(z(q_j) = i_g | y_j, \tilde{\Theta}_{i,i_g}) &= \frac{\tilde{\alpha}_{i,i_g} \cdot p(y_j | \tilde{\mu}_{i,i_g}, \tilde{\Sigma}_{i,i_g})}{\sum_{h=1}^{m} \tilde{\alpha}_{i,h} \cdot p(y_j | \tilde{\mu}_{i,h}, \tilde{\Sigma}_{i,h})} \\ \alpha_{i,i_g} &= \frac{1}{|\mathbf{y}_i|}\sum_{j=1}^{|\mathbf{y}_i|} p(z(q_j) = i_g | y_j, \tilde{\Theta}_{i,i_g}) \\ \lambda_{i,i_g} &= \sum_{j=1}^{|\mathbf{y}_i|} y_j \cdot p(z(q_j) = i_g | y_j, \tilde{\Theta}_{i,i_g}) \\ \gamma_{i,i_g} &= \sum_{j=1}^{|\mathbf{y}_i|} p(z(q_j) = i_g | y_j, \tilde{\Theta}_{i,i_g}) \cdot (y_j - \tilde{\mu}_{i,i_g})(y_j - \tilde{\mu}_{i,i_g})^T \end{aligned} \tag{7.17}$$

in which the probability $p(y_j | \tilde{\mu}_{i,i_g}, \tilde{\Sigma}_{i,i_g})$ can be directly computed from the Gaussian distribution defined by $\tilde{\mathcal{N}}_{i_g}(\tilde{\mu}_{i,i_g}, \tilde{\Theta}_{i,i_g})$. Then for each robot i, we define $\mathbf{x}_{i,i_g} \in \mathbb{R}^3$ as its local estimate of the global summary quantities $\alpha_{i_g}, \lambda_{i_g}, \gamma_{i_g}$ and define its own statistics $\mathbf{u}_{i,i_g} = [|\mathbf{y}_i| \alpha_{i,i_g}, \lambda_{i,i_g}, \gamma_{i,i_g}]^T$ as the local summary quantities. A consensus filter can be designed with convergence and stability guarantee [97], so that each robot i will agree on the similar value of Gaussian components Θ_{i_g} via peer-to-peer communication through connected network graph.

$$\dot{\mathbf{x}}_{i,i_g} = \sum_{j \in \text{neighbor of } i} (\mathbf{x}_{j,i_g} - \mathbf{x}_{i,i_g}) + (\mathbf{u}_{i,i_g} - \mathbf{x}_{i,i_g}) \tag{7.18}$$

where the neighbors of robot i are specified by all robots located spatially within a predefined distance to the robot i. It is noted that the above computation only

relies on the robot's local data set $\mathbf{y}_i$ and their communication packets of model information $\mathbf{x}_{i,i_g}$ (related to $\tilde{\Theta}_{i,i_g}$) only. With the converged estimated Gaussian Mixture Model parameters $\Theta_{i_g}$for all $i_g = 1, \ldots, m$, we are able to compute the weight distribution $P(z(q_j) = i_g) = p(z(q_j) = i_g|y_j, \Theta_{i_g})$ w.r.t. each Gaussian component i_g for each training data $\{q_j; y_j\}$ of robot i and obtained the training data classifications as follows.

$$\tilde{V}_i^{i_g} : \qquad \{q_j \in \mathbb{R}^2 | i_g = \arg\max P(z(q_j) = i_g)\}, \forall i_g = 1, \ldots, m \tag{7.19}$$

Intuitively, Eq. (7.16)-(7.19) indicate the process that the robots exchange model-related parameters to dynamically form an estimated and converged global statistics Θ_{i_g} of GMM, and hence to gradually readjust the classification of its local data in Eq. (7.19) as more samples are collected. As noted in [97], the standard EM with centralized computing has the complexity of communication in bytes as $O(n^{3/2})$ and $O(n^2)$ for the worst case, while with the distributed EM in Eq. (7.16)-(7.19), the complexity is $O(n)$ that is linear to the number of robots n. In the following Section 7.4.2, we will provide a complete distributed algorithm for computing the mixture of GPs with the classified local data.

7.4.2 Distributed Mixture of GPs in Adaptive Coverage Control

With the classified training dataset $\tilde{V}_i^{i_g}, \forall i_g = 1, \ldots, m$ from Eq. (7.19) for each robot i and weight distribution $P(z(q_j) = i_g)$ of each collected data therein, then the robots only need the knowledge of the predicted weight distribution (gating function) $P(z(q) = i_g)$ for $i_g = 1, \ldots, m$ that defines the likelihood each query data q belongs to the i_gth GP, so as to complete the modeling of mixture of GPs in form of Eq. (7.8) but with m GP models. Such gating function mapping from q to $P(z(q) = i_g)$ can be learned using another GP for each robot i by considering the already obtained training data $\{q_j; P(z(q_j) = i_g)\}, \forall q_j \in \tilde{V}_i$ in the similar form as Eq. (7.5). To that end, for any new data q sensed by robot i (inside robot i's Voronoi cell), it can compute the prediction Eq. (7.8) using Eq. (7.19) and the learned gating function $P(z(q) = i_g)$, which further yields the updated control law Eq. (7.10)-(7.11) to govern the motion of the robots.

Finally, our algorithm of distributed mixture of GPs in adaptive coverage control running on each robot i is summarized as follows.

Step 1: Take one sample from robot' current location and update its local data set $\{\tilde{V}_i; \mathbf{y_i}\}$. Recompute its Voronoi region.

Step 2: Compute local GMM parameter $\mathbf{u}_{i,i_g} = [|\mathbf{y}_i|\alpha_{i,i_g}, \lambda_{i,i_g}, \gamma_{i,i_g}]^T$ using Eq. (7.17) and current belief of global GMM parameters $\tilde{\Theta}_{i,i_g}$ on current local data set $\{\tilde{V}_i; \mathbf{y_i}\}$.

Step 3: Exchange local GMM model information of $\mathbf{x}_{i,i_g}$ with neighbors and compute the consensus using Eq. (7.18) till convergence and get updated GMM parameters $\tilde{\Theta}_{i,i_g}$ from converged $\mathbf{x}_{i,i_g}$ for all $i_g = 1, \ldots, m$.

Step 4: Classify local data set $\tilde{V}_i^{i_g}$ and weight distribution $P(z(q_j) = i_g)$ of each training data using Eq. (7.19) with $\tilde{\Theta}_{i,i_g}$ for all $i_g = 1, \ldots, m$.

Step 5: Train gating function $P(z(q) = i_g)$ with training data $\{q_j; P(z(q_j) = i_g)\}$ and together with the classified local data set $\tilde{V}_i^{i_g}, \forall i_g = 1, \ldots, m$, locally fit mixture of GPs using Eq. (7.8) with m GP models.

Step 6: With the learned mixture of GPs, predict the posterior mean and variance of density function $\phi(q)$ over its Voronoi region and evaluate each $h(q)$ in the region using Eq. (7.9).

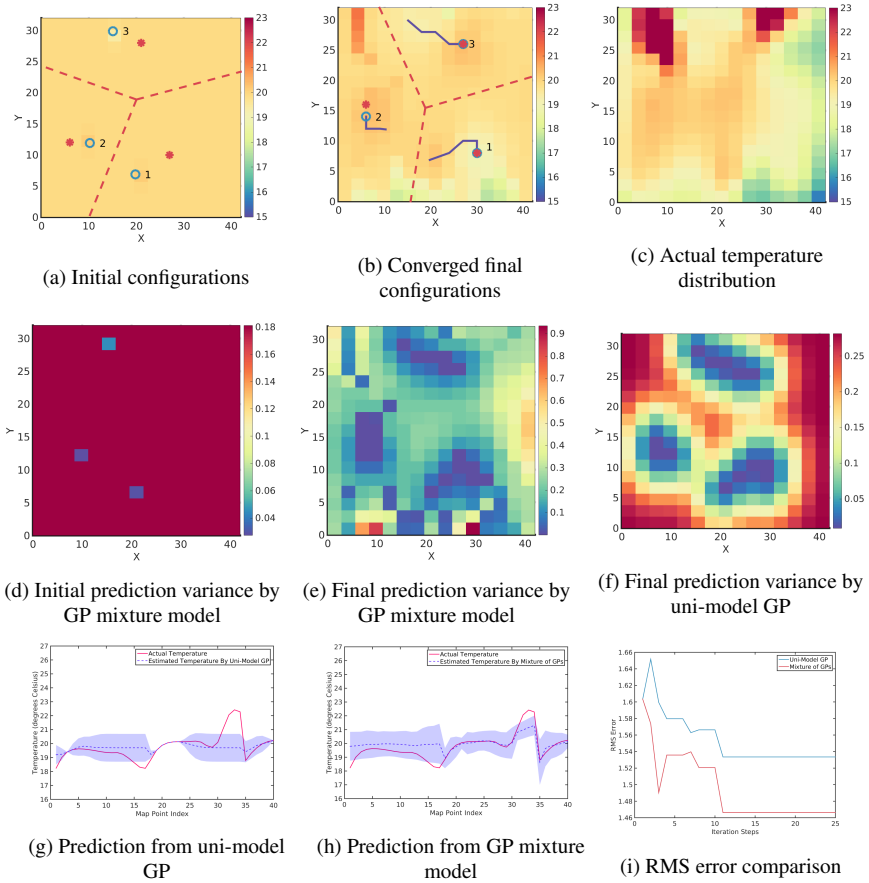
Step 7: Compute local adaptive coverage control law with Eq. (7.11), execute, and go back to *Step 1*.

7.5 Results

Centralized Learning

In this section, we present several simulation results on the benchmark real-world dataset from Intel Berkeley Lab [123] with MATLAB toolboxes: the GPML [125]. The dataset contains sensory data collected from 54 sensors in an office area between Feb 28th and Apr 5th, 2004. The data includes the time-stamped readings such as sensor 2D locations, temperature, humidity, light, and voltage. In our particular tasks, we use the 2D location information (meters) of each sensor with the temperature readings (degrees Celsius) as the ground truth of the environmental phenomenon over map and compare our algorithm performance to other approaches.

First we consider an example where we have 3 robots deployed from random starting points $(19.78, 6.84)$, $(10.22, 11.89)$ and $(15.09, 29.90)$ to find the optimal final configurations for stationary sensing as shown earlier in Fig. 7.1, where the temperature distribution has two peaks around the top corners. Once deployed the robots are governed by our adaptive coverage controller (7.11) with mixture of GPs ($k_p = 0.5, \beta = 10$) to simultaneously learn the environmental model and try to approach the actual centroid of each Voronoi cell based on its own model inference. As shown in Fig. 7.2(a) and (d) the robots first assume a uniform distribution of temperature over the map based on the data collected from the initial configurations with little uncertainties. Note that the temperature from each discrete point can only be acquired when the robot chooses it as the next point to visit (sample) except for the initial configuration. The converged results are shown in Fig. 7.2(b)



(a) Initial configurations

(b) Converged final configurations

(c) Actual temperature distribution

(d) Initial prediction variance by GP mixture model

(e) Final prediction variance by GP mixture model

(f) Final prediction variance by uni-model GP

(g) Prediction from uni-model GP

(h) Prediction from GP mixture model

(i) RMS error comparison

Figure 7.2: An example of the multi-robot sensor coverage and environment modeling results by using GP mixture model with comparison to uni-model GP. (a) Initial and (b) final configurations of the robots (marked by blue circles) with history footprints by controller using GP mixture model. The background heatmap indicates the predicted temperature distribution based on the sampled data. Edge of Voronoi cells and the optimal locations (centroids of Voronoi cells from actual temperature distribution) are represented by red dashed lines and red stars, respectively. (c) Actual temperature distribution over map. (d) Initial and (e) final predicted variance distribution by GP mixture model. (f) Converged prediction variance from uni-model GP. (g)-(h) Temperature prediction comparison with standard deviation from (g) uni-model GP and (h) GP mixture model. (i) RMS error comparison.

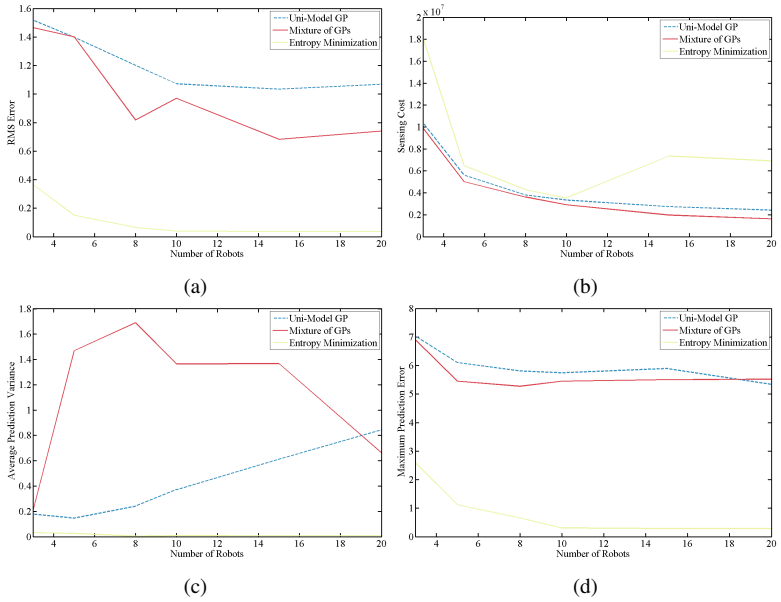
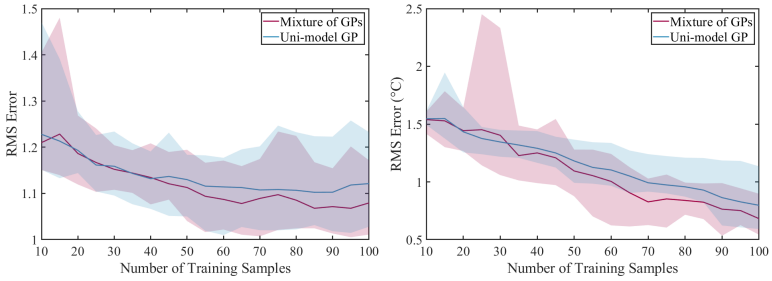


Figure 7.3: Comparisons of RMS error, sensing cost, average prediction variance and maximum prediction error occurred v.s. different number of robots.

in comparison with the actual temperature distribution in Fig. 7.2(c) and the prediction variance is also given in Fig. 7.2(e). It is noted that the converged configuration is very close to the optimal one from the actual temperature distribution due to our adaptive sampling algorithm that trades off between uncertainty reduction and centroid approaching. Moreover, although none of the robots actually visited the top corner areas with much higher temperature, the mixture of GPs is able to identify the prediction differences among the robots over similar areas and adjust the mixture accordingly to best fit the various local features, by predicting a higher temperature with higher uncertainties over those areas in Fig. 7.2(e). In contrast, using uni-model GP could ignore the local features and hence the prediction variance is almost the same over any unvisited areas as shown in Fig. 7.2(f).

To better understand the performance, we provide the comparisons on local temperature prediction as shown in Fig. 7.2(g)-(h). Note that although the uni-model GP can have accurate prediction over places close to the sampled points, it fails to recognize the peak temperature in areas surrounding the robots, which can be identified by the mixture of GPs. We also compare the root-mean-square error (RMS) in Fig. 7.2(i), where the mixture of GPs shows a better performance with lower prediction error. To further compare our algorithms with other sampling approaches, we run the simulations with different number of robots under different algorithms, including the aforementioned uni-model GP and the Entropy minimization algorithm [1] that seeks to find the point in Voronoi cell which best



(a) Agricultural Data: Stalk Count (b) Office Data: Temperature

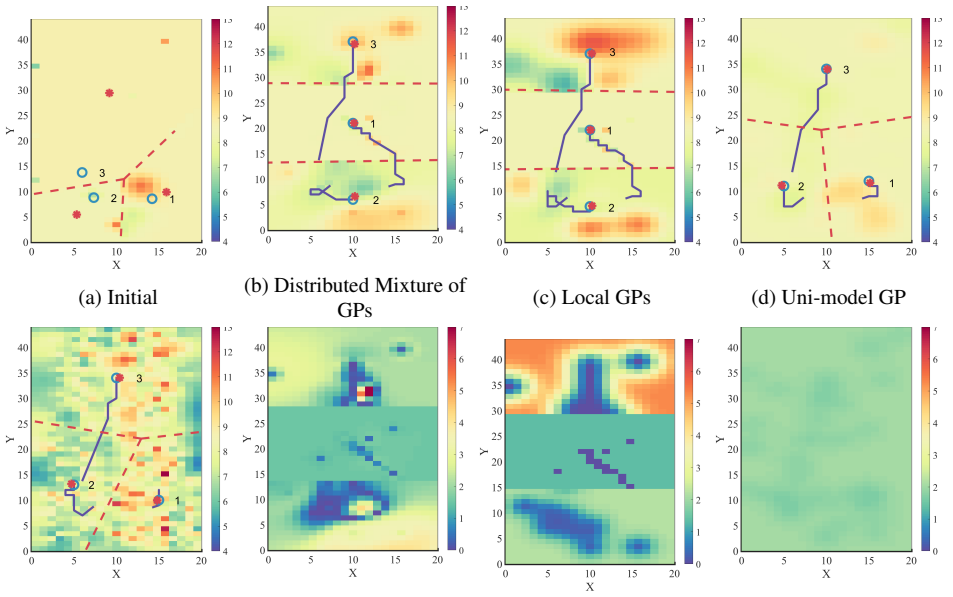
Figure 7.4: Prediction performance comparison between mixture of GPs and uni-model GP on two real-world dataset.

reduces the model prediction uncertainty. The results are shown in Fig. 7.3 and our Mixture of GPs algorithm always outperforms the uni-model GP algorithm. Although the algorithm in [1] has a better sampling performance w.r.t. model uncertainty, the resultant sensing cost is much higher in Fig. 7.3(b) and hence it cannot be directly applied to our problem. The reason lies in that it prefers areas with higher uncertainty to the ones with higher predicted value, and due to the greedy based coverage control law it could converge to the locations that are further away from the areas with peak temperature.

Decentralized Learning

In this section, we present several empirical results on two real-world datasets from the agricultural robotic sampling application [122] and Intel Berkeley Lab [123] with MATLAB toolboxes: the GPML [125]. The agricultural dataset contains data of the number of stalks counted per grid over the $21 \times 45 = 945$ distinct grids collected in August 2017 over a sorghum field. In our particular tasks, we use the 2D location information with the stalk count and temperature readings (degrees in Celsius) respectively as the ground truth of the environmental phenomenon over map and compare our algorithm performance to other approaches.

Before implementing our multi-robot sensor coverage task, we first provide the empirical results for static prediction of the two environmental phenomenon using centralized mixture of GPs ($m = 3$) and centralized uni-model GP respectively. For each dataset, we randomly select a growing number of data from 10 to 100 to serve as the training data and use the rest of the unselected data as testing data. After 10 random trials at each training sample setting, the prediction performance on the Root Mean Square (RMS) error are plotted in Fig. 7.4 with solid lines as the mean RMS error covered by the maximum-minimum error bar showing the maximum and minimum predictive RMS error in each 10 trials. The results suggest as the number of training samples grow, the mixture of GPs outperforms



(a) Initial (b) Distributed Mixture of GPs (c) Local GPs (d) Uni-model GP

(e) Control with known model (f) Variance by Distributed Mixture of GPs (g) Variance by local GPs (h) Variance by uni-model GP

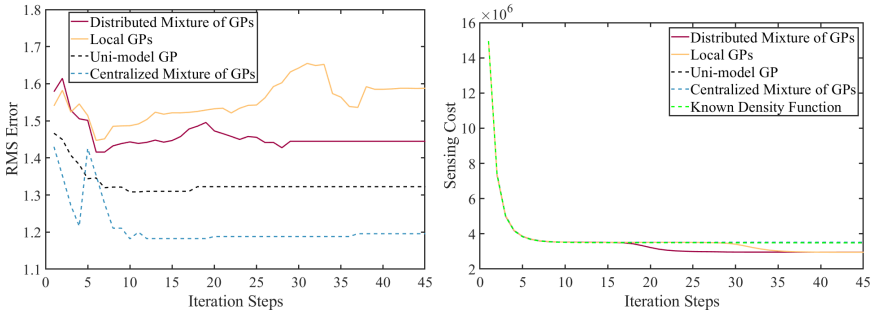
Figure 7.5: An simulation example of 3 robots covering an agricultural field and the modeling results using our distributed mixture of GP model in comparison to 1) local GPs with local data and 2) uni-model GP with global knowledge of collected data from all robots. Robots' current positions are marked by blue circles with dark blue history footprints in (a)-(e). The background heatmap in (a)-(d) indicates the predicted stalk count distribution based on the collected data and (e) the true distribution. The background heatmap in (f)-(h) represents the predicted variance over the map. Edge of Voronoi cells and the ground-truth optimal locations for that particular configurations are represented by red dashed lines and red stars, respectively.

the uni-model GP in both datasets, indicating that for the considered real-world non-smooth data the mixture of GPs can better characterize the distribution of environmental phenomenon.

Simulation Example with Agricultural Dataset

For multi-robot sensor coverage task, we consider the example in Fig. 7.5 where we have 3 robots deployed from random starting points shown in Fig. 7.5(a) to find the optimal final configurations for stationary sensing, where the distribution of stalk count has multiple peaks around the top right corners.

As shown in Fig. 7.5(a), the robots initially have little knowledge about the true distribution with only 13 randomly chosen prior training data points over the map. With our distributed mixture of GPs algorithm in Fig. 7.5(b), the robots are governed by our proposed adaptive coverage controller Eq. (7.11) with distributed learning of mixture of GPs ($k_p = 0.5, \beta = 1, m = 3$) to simultaneously learn the environmental model and try to approach the actual centroid of each Voronoi cell

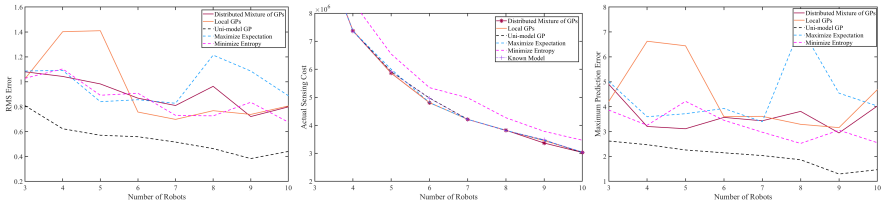


(a) RMS error (b) Sensing cost

Figure 7.6: Prediction and coverage performance comparison.

based on its own model inference. We assume any robot pairs sharing the same Voronoi edge could communicate to each other. The converged results of configurations under adaptive coverage controllers with other modeling approaches are shown in Fig. 7.5(c)-(e), where the local GPs (Fig. 7.5(c)) are trained by each robot with its own local collected data without communication and the uni-model GP (Fig. 7.5(d)) assumes global knowledge of all robots' collected data. The known model (Fig. 7.5(e)) is the ground truth controller with full knowledge of the density function as done in [4]. They have different modeling of $\phi(q)$ but use the same form of our adaptive coverage control laws Eq. (7.4). Quantitative prediction and coverage performance on the same map are also provided in Fig. 7.6 and evaluated by the metrics of 1) Root Mean Square (RMS) error between predicted stalk count and ground truth stalk count on all unvisited locations, and 2) the actual sensing cost computed by Eq. (7.2). Besides the mentioned comparison algorithms, in Fig. 7.6 we also introduce the result from centralized mixture of GPs modeling algorithm [3] with the same parameter settings ($k_p = 0.5, \beta = 1, m = 3$), but use the training data from all robots. It is noted from Fig. 7.6 that when assuming global knowledge of all robot's data, the centralized mixture of GPs has better prediction performance over uni-model GP. When the global information is not available, our proposed distributed mixture of GPs has a better prediction performance w.r.t. RMS error than local GPs and the best coverage performance level.

In particular, it is also noted from Fig. 7.5(b) that the robots with a distributed GP mixture can successfully identify the top right corner with 3 distinguished peak areas by exchanging model information, while in Fig. 7.5(c) robot 3 falsely predicts a larger peak area due to no information exchange. With the uni-model GP shown in Fig. 7.5(d), the robots fail to identify significantly different components due to its uni-model inference (although it performs better in prediction RMS error compared to the distributed algorithms due to its global knowledge of all the collected data). Results of prediction variance are shown in Fig. 7.5(f)-(h) and



(a) RMS error (b) Sensing cost (c) Maximum prediction error

Figure 7.7: Comparisons of RMS error, sensing cost, and maximum prediction error occurred v.s. different number of robots for different algorithms.

our distributed mixture of GPs has smoother prediction due to the consensus of environment model but are still able to capture the prediction uncertainty differences over the area. Local GP method in Fig. 7.5(g) has much larger prediction variance among different robots as they are sampling in different places with significantly distinct components without information exchanging. Using uni-model GP in Fig. 7.5(h) ignores the local features and hence the prediction variance is almost the same over any unvisited areas. It is noted that even with only a few samples collected from the map, the converged configurations are very close to the optimal ones from the actual temperature distribution due to our adaptive sampling criterion and algorithm that trades off between uncertainty reduction and centroid approaching.

Quantitative Results

To further compare our algorithms performance in other environmental phenomenon, we run 40 trials on 5 different sets of data from Intel Berkeley dataset [123] with 3 to 10 robots respectively in Fig. 7.7. Note that as the complexity of communication for our distributed mixture of GPs is linear to the number of robots and each robot only computes on the local data only, our algorithm is also scalable to a larger number of robots. The other modeling algorithms we are comparing includes the mentioned local GPs, uni-model GP, known model coverage control, as well as the Expectation maximization ($\beta = 0$) and Entropy minimization algorithm ($\beta > 20$) coupled with the same form of coverage controller in Eq. (7.4). Expectation maximization algorithm seeks to find the area with highest value of predicted environmental phenomenon while the Entropy minimization algorithm seeks to find the area with highest prediction variance to reduce model uncertainty. The centralized uni-model GP has the best performance w.r.t. RMS error and prediction error due to its global knowledge of data. In particular, the Entropy minimization has the best prediction performance w.r.t. the RMS error and maximum prediction error, but the worst coverage performance since it only prefers high uncertainty area to the place with higher density value, and thus lead to infe-

rior suboptimal configurations due to local minima nature of the move-to-centroid controller (finding global optimal config is NP hard). Our distributed Mixture of GPs has the best coverage performance even with only local data (considering the scale of the cost) and in general the second best performance in prediction (following Entropy minimization) due to the GP-UCB criteria in our algorithm that trades off between uncertainty reduction and prediction maximization. As the number of robots increases, our algorithm outperforms the local GPs w.r.t. RMS error and maximum prediction error as our distributed Mixture of GPs is able to approximate global statistics via the consensus algorithm.

7.6 Conclusion and Discussions

In this chapter, we present adaptive sampling algorithms for learning the density function in multi-robot sensor coverage problem using Mixture of Gaussian Processes models. By using the information-theoretic sampling criterion we are able to modify the traditional coverage control law to consider the uncertainty as well as the potential environmental phenomenon inferred from the environmental model learned on-line. Besides, considering significantly different components that may exist in the real-world environmental phenomenon, we propose to employ the mixture of GP models to capture local features for the global distribution by optimizing the linear combination of GP models locally learned by the robots. Simulation results have shown the effectiveness of our algorithm compared to other approaches.

For both centralized and decentralized learning, they rely on the communication between robots to exchange information. Although this chapter does not explicitly discuss how to enforce connectivity maintenance constraints to achieve the assumption, it is straightforward that one could use the adaptive coverage controller as the nominal controller and pass it to our connectivity and resilient connectivity maintenance framework in Chapter 5-6. An example of using Voronoi-based coverage control together with our connectivity maintenance framework is provided in [6]. On the other hand, the communication topology in the robot team could affect the convergence rate of the consensus algorithm for the distributed learning, and hence how to address such additional topology consideration to further improve the computation performance is another important future direction.

Part IV

Final Words

7.7 Conclusions

In this thesis, we discussed the fundamental problems of robot behavior design to account for safety and resilience for large-scale networked autonomous systems under uncertainty and developed computationally-efficient and provably-correct decision-making algorithms. We first present an explicit behavior design for computationally efficient safety assurance under uncertainty on large-scale autonomous systems, such as a team of drones. In the presence of unknown robot models and uncertain environments, we then proposed a sample efficient safe reinforcement learning framework that integrates control-theoretic safe design into a learning-based approach for a robot to learn to optimally perform a task with safety guarantee.

To address the resilience autonomy, we discussed the effective multi-robot coordination with desired inter-robot information exchange capability through coordinated networking behaviors against defective robots. Then with safety and communication assurance, we discussed how these results lead to reliable multi-robot behaviors design with guaranteed performance for practical applications such as data-driven environmental sampling and monitoring. Finally, I will discuss future challenges and new ideas to build long-term autonomy that is correct by design for robots to safely and reliably collaborate with humans and each other in a variety of real-world applications. We believe the presented work in this thesis will facilitate new application domains for a future of human-cyber-physical systems (h-CPS), such as collaborative human-robot manufacturing and mixed autonomy in semi-autonomous driving.

7.8 Future Research Directions

7.8.1 Safety Assurance for Human-Centered Autonomous Systems

Recent advances in AI and robotics rapidly expand the robotic technologies from factory floors to the shared space where human and robots are working much more closely with each other, e.g. in autonomous driving and massive warehouse management. While decision making algorithms such as deep RL methods have started emerging and shown success in complex robotic tasks, the optimization process is often like a "black-box" and verifiable safety-critical consideration is in desperate need, especially in the face of uncertainty from the environment and human behaviors. In my recent work [13, 14], I have studied provably correct method to ensure safety for multi-robot systems under uncertainties and address safe exploration and exploitation in learning for control. When it comes to interaction with unknown systems and humans, it requires reasoning over not only the robot's own uncer-

tainty, but also the uncertainty of other agents and humans in the shared space, and how actions impact the evolving uncertainties. In the future work, I will draw inspirations from machine learning, control theory, and cognitive science to characterize the human behavioral models and the mutual influence between robots' decisions and those of humans or other unknown agents for effective safety assurance and coordination in various applications, e.g. mixed transportation networks with human-driven and self-driving cars, and collaborative human-robot manufacturing.

7.8.2 Emergent and Resilient Teamwork for Long Term Autonomy

The current state of the art algorithms in collaborative robotic teamwork often focus on static objectives with predefined roles, structures, and constraints given by humans from domain knowledge—a setting where robots solve one or a fix sequence of explicitly defined optimization problems with constraints. However, the increasing scale of interplay among robots from different groups with diverse capabilities makes the coordination process far more complex and often unpredictable. The objectives, team configurations and constraints may change over time as mission and inter-robot relationship evolves. For example, if some robots fail to fulfill their roles in coordination, how can we let the other robots decide *on their own* which ones to take over those roles and how to modify the behaviors for the others? My research on resilient communication maintenance [15, 16, 17] and task allocation [128] are examples for emergent communication topology in robot teaming. The design goal is to keep robots connected and resilient to failures, but we allow the robots themselves to compute the optimal dynamic communication topology and improvise the connectivity constraints to enforce that are in favor of the overall task performance. Building upon this, in the future I am interested in bringing multi-agent learning, network science and control theoretic approaches to 1) model and formalize emergent inter-robot relationships with design features for diversity and heterogeneity, and 2) automate the synthesis of emerging spatial and temporal behaviors in long-term resilient collaboration.

In the near future, robotic systems could increasingly co-exist with humans in our daily lives, from household service to large-scale warehouse logistics, agriculture environment sampling, and smart city. Among these applications, robots and humans as networked heterogeneous components will frequently interact with each other in a variety of interactions in uncertain, rapidly-changing, and possibly hostile environment. With the development of technology, cooperative autonomous systems will play a more significant role in our future life that safely operate in close proximity to human presence, going beyond mere teleoperation.

Bibliography

[1] Stephanie Kemna, John G Rogers, Carlos Nieto-Granda, Stuart Young, and Gaurav S Sukhatme. "Multi-robot coordination through dynamic Voronoi partitioning for informative adaptive sampling in communication-constrained environments". In: *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2017, pp. 2124–2130.

[2] Wenhao Luo, Shehzaman S Khatib, Sasanka Nagavalli, Nilanjan Chakraborty, and Katia Sycara. "Distributed knowledge leader selection for multi-robot environmental sampling under bandwidth constraints". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2016, pp. 5751–5757.

[3] Wenhao Luo and Katia Sycara. "Adaptive Sampling and Online Learning in Multi-Robot Sensor Coverage with Mixture of Gaussian Processes". In: *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2018, pp. 6359–6364.

[4] Jorge Cortés, Sonia Martınez, Timur Karatas, and Francesco Bullo. "Coverage control for mobile sensing networks". In: 20.2 (2004).

[5] Mac Schwager, Michael P Vitus, Samantha Powers, Daniela Rus, and Claire J Tomlin. "Robust adaptive coverage control for robotic sensor networks". In: *IEEE Transactions on Control of Network Systems* 4.3 (2015), pp. 462–476.

[6] Wenhao Luo and Katia Sycara. "Voronoi-based Coverage Control with Connectivity Maintenance for Robotic Sensor Networks". In: *International Symposium on Multi-Robot and Multi-Agent Systems (MRS)*. IEEE. 2019, pp. 148–154.

[7] Illah R Nourbakhsh, Katia Sycara, Mary Koes, Mark Yong, Michael Lewis, and Steve Burion. "Human-robot teaming for search and rescue". In: *IEEE Pervasive Computing* 4.1 (2005), pp. 72–79.

[8] Felix Berkenkamp, Matteo Turchetta, Angela Schoellig, and Andreas Krause. "Safe model-based reinforcement learning with stability guarantees". In: *Advances in neural information processing systems*. 2017, pp. 908–918.

[9] Jaime F Fisac, Anayo K Akametalu, Melanie N Zeilinger, Shahab Kaynama, Jeremy Gillula, and Claire J Tomlin. "A general safety framework for learning-based control in uncertain robotic systems". In: *IEEE Transactions on Automatic Control* 64.7 (2018), pp. 2737–2752.

[10] Li Wang, Aaron D Ames, and Magnus Egerstedt. "Safe certificate-based maneuvers for teams of quadrotors using differential flatness". In: *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2017, pp. 3293–3298.

[11] Andrew Taylor, Andrew Singletary, Yisong Yue, and Aaron Ames. "Learning for safety-critical control with control barrier functions". In: *Learning for Dynamics and Control*. PMLR. 2020, pp. 708–717.

[12] Richard Cheng, Gábor Orosz, Richard M Murray, and Joel W Burdick. "End-to-end safe reinforcement learning through barrier functions for safety-critical continuous control tasks". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 33. 2019, pp. 3387–3395.

[13] Wenhao Luo, Wen Sun, and Ashish Kapoor. "Multi-Robot Collision Avoidance under Uncertainty with Probabilistic Safety Barrier Certificates". In: *Advances in neural information processing systems (NeurIPS)* (2020).

[14] Wenhao Luo, Wen Sun, and Ashish Kapoor. "No-Regret Safe Learning for Online Nonlinear Control with Control Barrier Functions". In: *ICRA 2021 Workshop on Safe Robot Control with Learned Motion and Environment Models* (2021).

[15] Wenhao Luo, Sha Yi, and Katia Sycara. "Behavior Mixing with Minimum Global and Subgroup Connectivity Maintenance for Large-Scale Multi-Robot Systems". In: *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2020.

[16] Wenhao Luo and Katia Sycara. "Minimum k-Connectivity Maintenance for Robust Multi-Robot Systems". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2019, pp. 7370–7377.

[17] Wenhao Luo, Nilanjan Chakraborty, and Katia Sycara. "Minimally disruptive connectivity enhancement for resilient multi-robot teams". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2020, pp. 11809–11816.

[18] Wenhao Luo, Changjoo Nam, George Kantor, and Katia Sycara. "Distributed Environmental Modeling and Adaptive Sampling for Multi-Robot Sensor Coverage". In: *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*. International Foundation for Autonomous Agents and Multiagent Systems. 2019, pp. 1488–1496.

[19] Dorsa Sadigh and Ashish Kapoor. "Safe Control under Uncertainty with Probabilistic Signal Temporal Logic". In: *Proceedings of Robotics: Science and Systems*. AnnArbor, Michigan, June 2016. DOI: 10.15607/RSS.2016.XII.017.

[20] Hai Zhu and Javier Alonso-Mora. "Chance-constrained collision avoidance for mavs in dynamic environments". In: *IEEE Robotics and Automation Letters* 4.2 (2019), pp. 776–783.

[21] Hai Zhu and Javier Alonso-Mora. "B-UAVC: Buffered Uncertainty-Aware Voronoi Cells for Probabilistic Multi-Robot Collision Avoidance". In: *International Symposium on Multi-Robot and Multi-Agent Systems (MRS)*. IEEE. 2019.

[22] Bharath Gopalakrishnan, Arun Kumar Singh, Meha Kaushik, K Madhava Krishna, and Dinesh Manocha. "Prvo: Probabilistic reciprocal velocity obstacle for multi robot navigation under uncertainty". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2017, pp. 1089–1096.

[23] Michael P Vitus and Claire J Tomlin. "A hybrid method for chance constrained control in uncertain environments". In: *IEEE 51st IEEE Conference on Decision and Control (CDC)*. IEEE. 2012, pp. 2177–2182.

[24] Daniel Claes, Daniel Hennes, Karl Tuyls, and Wim Meeussen. "Collision avoidance under bounded localization uncertainty". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2012, pp. 1192–1198.

[25] Jason Hardy and Mark Campbell. "Contingency planning over probabilistic obstacle predictions for autonomous road vehicles". In: *IEEE Transactions on Robotics* 29.4 (2013), pp. 913–929.

[26] Mina Kamel, Javier Alonso-Mora, Roland Siegwart, and Juan Nieto. "Robust collision avoidance for multiple micro aerial vehicles using nonlinear model predictive control". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2017, pp. 236–243.

[27] Chonhyon Park, Jae S Park, and Dinesh Manocha. "Fast and bounded probabilistic collision detection for high-DOF trajectory planning in dynamic environments". In: *IEEE Transactions on Automation Science and Engineering* 15.3 (2018), pp. 980–991.

[28] Aaron D Ames, Samuel Coogan, Magnus Egerstedt, Gennaro Notomista, Koushil Sreenath, and Paulo Tabuada. "Control barrier functions: Theory and applications". In: *18th European Control Conference (ECC)*. IEEE. 2019, pp. 3420–3431.

[29] Li Wang, Aaron D Ames, and Magnus Egerstedt. "Safety barrier certificates for collisions-free multirobot systems". In: *IEEE Transactions on Robotics* 33.3 (2017), pp. 661–674.

[30] Jun Zeng, Bike Zhang, and Koushil Sreenath. "Safety-critical model predictive control with discrete-time control barrier function". In: *arXiv preprint arXiv:2007.11718* (2020).

[31] Quan Nguyen and Koushil Sreenath. "Exponential control barrier functions for enforcing high relative-degree safety-critical constraints". In: *American Control Conference (ACC)*. IEEE. 2016, pp. 322–328.

[32] Javier Garcıa and Fernando Fernández. "A comprehensive survey on safe reinforcement learning". In: *Journal of Machine Learning Research* 16.1 (2015), pp. 1437–1480.

[33] Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. "Constrained policy optimization". In: *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. 2017, pp. 22–31.

[34] Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. "Concrete problems in AI safety". In: *arXiv preprint arXiv:1606.06565* (2016).

[35] Teodor Mihai Moldovan and Pieter Abbeel. "Safe exploration in markov decision processes". In: *arXiv preprint arXiv:1205.4810* (2012).

[36] Matteo Turchetta, Felix Berkenkamp, and Andreas Krause. "Safe exploration in finite markov decision processes with gaussian processes". In: *Advances in Neural Information Processing Systems* 29 (2016), pp. 4312–4320.

[37] Sham Kakade, Akshay Krishnamurthy, Kendall Lowrey, Motoya Ohnishi, and Wen Sun. "Information theoretic regret bounds for online nonlinear control". In: *Advances in Neural Information Processing Systems* 33 (2020).

[38] Li Wang, Evangelos A Theodorou, and Magnus Egerstedt. "Safe learning of quadrotor dynamics using barrier certificates". In: *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2018, pp. 2460–2465.

[39] Jason Choi, Fernando Castañeda, Claire Tomlin, and Koushil Sreenath. "Reinforcement Learning for Safety-Critical Control under Model Uncertainty, using Control Lyapunov Functions and Control Barrier Functions". In: *Proceedings of Robotics: Science and Systems*. Corvalis, Oregon, USA, July 2020.

[40] Li Wang, Aaron D Ames, and Magnus Egerstedt. "Multi-objective compositions for collision-free connectivity maintenance in teams of mobile robots". In: *IEEE 55th Conference on Decision and Control (CDC)*. IEEE. 2016, pp. 2659–2664.

[41] Daniel Zelazo, Antonio Franchi, Frank Allgöwer, Heinrich H Bülthoff, and P Robuffo Giordano. "Rigidity maintenance control for multi-robot systems". In: *Robotics: science and systems*. 2012, pp. 473–480.

[42] Cinara Ghedini, Carlos HC Ribeiro, and Lorenzo Sabattini. "A decentralized control strategy for resilient connectivity maintenance in multi-robot systems subject to failures". In: *Distributed Autonomous Robotic Systems*. Springer, 2018, pp. 89–102.

[43] Jacopo Panerati, Marco Minelli, Cinara Ghedini, Lucas Meyer, Marcel Kaufmann, Lorenzo Sabattini, and Giovanni Beltrame. "Robust connectivity maintenance for fallible robots". In: *Autonomous Robots* 43.3 (2019), pp. 769–787.

[44] Jorge Cortés. "Coverage optimization and spatial load balancing by robotic sensor networks". In: *IEEE Transactions on Automatic Control* 55.3 (2010), pp. 749–754.

[45] Jaime Fisac, Andrea Bajcsy, Sylvia Herbert, David Fridovich-Keil, Steven Wang, Claire Tomlin, and Anca Dragan. "Probabilistically Safe Robot Planning with Confidence-Based Human Predictions". In: *Proceedings of Robotics: Science and Systems*. Pittsburgh, Pennsylvania, June 2018. DOI: 10.15607/RSS.2018.XIV.069.

[46] Sylvia L Herbert, Mo Chen, SooJean Han, Somil Bansal, Jaime F Fisac, and Claire J Tomlin. "FaSTrack: a modular framework for fast and guaranteed safe motion planning". In: *IEEE 56th Annual Conference on Decision and Control (CDC)*. IEEE. 2017, pp. 1517–1522.

[47] Javier Alonso-Mora, Andreas Breitenmoser, Paul Beardsley, and Roland Siegwart. "Reciprocal collision avoidance for multiple car-like robots". In: *IEEE International Conference on Robotics and Automation*. IEEE. 2012, pp. 360–366.

[48] Javier Alonso-Mora, Andreas Breitenmoser, Martin Rufli, Paul Beardsley, and Roland Siegwart. "Optimal reciprocal collision avoidance for multiple non-holonomic robots". In: *Distributed Autonomous Robotic Systems*. Springer, 2013, pp. 203–216.

[49] Jur Van den Berg, Ming Lin, and Dinesh Manocha. "Reciprocal velocity obstacles for real-time multi-agent navigation". In: *IEEE International Conference on Robotics and Automation*. IEEE. 2008, pp. 1928–1935.

[50] Jur Van Den Berg, Jamie Snape, Stephen J Guy, and Dinesh Manocha. "Reciprocal collision avoidance with acceleration-velocity obstacles". In: *IEEE International Conference on Robotics and Automation*. IEEE. 2011, pp. 3475–3482.

[51] Urs Borrmann, Li Wang, Aaron D Ames, and Magnus Egerstedt. "Control barrier certificates for safe swarm behavior". In: *IFAC-PapersOnLine* 48.27 (2015), pp. 68–73.

[52] Dingjiang Zhou, Zijian Wang, Saptarshi Bandyopadhyay, and Mac Schwager. "Fast, on-line collision avoidance for dynamic vehicles using buffered voronoi cells". In: *IEEE Robotics and Automation Letters* 2.2 (2017), pp. 1047–1054.

[53] Mingyu Wang and Mac Schwager. "Distributed Collision Avoidance of Multiple Robots with Probabilistic Buffered Voronoi Cells". In: *International Symposium on Multi-Robot and Multi-Agent Systems (MRS)*. IEEE. 2019.

[54] PS Naga Jyotish, Yash Goel, AVS Sai Bhargav Kumar, and K Madhava Krishna. "PIVO: Probabilistic Inverse Velocity Obstacle for Navigation under Uncertainty". In: *28th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*. IEEE. 2019, pp. 1–6.

[55] Aaron D Ames, Xiangru Xu, Jessy W Grizzle, and Paulo Tabuada. "Control Barrier Function Based Quadratic Programs for Safety Critical Systems". In: *IEEE Transactions on Automatic Control* 62.8 (2017), pp. 3861–3876.

[56] Bin Xu and Koushil Sreenath. "Safe teleoperation of dynamic uavs through control barrier functions". In: *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2018, pp. 7848–7855.

[57] Yan Duan, Xi Chen, Rein Houthooft, John Schulman, and Pieter Abbeel. "Benchmarking deep reinforcement learning for continuous control". In: *International Conference on Machine Learning*. 2016, pp. 1329–1338.

[58] Benjamin Recht. "A tour of reinforcement learning: The view from continuous control". In: *Annual Review of Control, Robotics, and Autonomous Systems* 2 (2019), pp. 253–279.

[59] Francois R Hogan and Alberto Rodriguez. "Reactive planar non-prehensile manipulation with hybrid model predictive control". In: *The International Journal of Robotics Research* (2020), p. 0278364920913938.

[60] Grady Williams, Nolan Wagener, Brian Goldfain, Paul Drews, James M Rehg, Byron Boots, and Evangelos A Theodorou. "Information theoretic MPC for model-based reinforcement learning". In: *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2017, pp. 1714–1721.

[61] Andrew J Taylor and Aaron D Ames. "Adaptive safety with control barrier functions". In: *2020 American Control Conference (ACC)*. IEEE. 2020, pp. 1399–1405.

[62] Thomas Gurriet, Andrew Singletary, Jacob Reher, Laurent Ciarletta, Eric Feron, and Aaron Ames. "Towards a framework for realizable safety critical control through active set invariance". In: *2018 ACM/IEEE 9th International Conference on Cyber-Physical Systems (ICCPS)*. IEEE. 2018, pp. 98–106.

[63] Andrew Clark. "Control barrier functions for complete and incomplete information stochastic systems". In: *2019 American Control Conference (ACC)*. IEEE. 2019, pp. 2928–2935.

[64] Mohammad Javad Khojasteh, Vikas Dhiman, Massimo Franceschetti, and Nikolay Atanasov. "Probabilistic safety constraints for learned high relative degree system dynamics". In: *Learning for Dynamics and Control*. 2020, pp. 781–792.

[65] Motoya Ohnishi, Li Wang, Gennaro Notomista, and Magnus Egerstedt. "Barrier-certified adaptive reinforcement learning with applications to brushbot navigation". In: *IEEE Transactions on robotics* 35.5 (2019), pp. 1186–1205.

[66] Dimos V Dimarogonas and Karl H Johansson. "Decentralized connectivity maintenance in mobile networks with bounded inputs". In: *IEEE International Conference on Robotics and Automation*. IEEE. 2008, pp. 1507–1512.

[67] Meng Ji and Magnus Egerstedt. "Distributed coordination control of multiagent systems while preserving connectedness". In: *IEEE Transactions on Robotics* 23.4 (2007), pp. 693–703.

[68] Michael M Zavlanos, Ali Jadbabaie, and George J Pappas. "Flocking while preserving network connectivity". In: *46th IEEE Conference on Decision and Control*. IEEE. 2007, pp. 2919–2924.

[69] Paolo Robuffo Giordano, Antonio Franchi, Cristian Secchi, and Heinrich H Bülthoff. "Bilateral Teleoperation of Groups of UAVs with Decentralized Connectivity Maintenance." In: *Robotics: Science and Systems*. Citeseer. 2011.

[70] Lorenzo Sabattini, Nikhil Chopra, and Cristian Secchi. "Decentralized connectivity maintenance for cooperative control of mobile robotic systems". In: *The International Journal of Robotics Research* 32.12 (2013), pp. 1411–1423.

[71] Peng Yang, Randy A Freeman, Geoffrey J Gordon, Kevin M Lynch, Siddhartha S Srinivasa, and Rahul Sukthankar. "Decentralized estimation and control of graph connectivity for mobile sensor networks". In: *Automatica* 46.2 (2010), pp. 390–396.

[72] Ryan K Williams, Andrea Gasparri, Gaurav S Sukhatme, and Giovanni Ulivi. "Global connectivity control for spatially interacting multi-robot systems with unicycle kinematics". In: *IEEE international conference on robotics and automation (ICRA)*. IEEE. 2015, pp. 1255–1261.

[73] Jacopo Banfi, Nicola Basilico, and Stefano Carpin. "Optimal Redeployment of Multirobot Teams for Communication Maintenance". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2018, pp. 3757–3764.

[74] Meghan Chandarana, Wenhao Luo, Michael Lewis, Katia Sycara, and Sebastian Scherer. "Decentralized Method for Sub-Swarm Deployment and Rejoining". In: *IEEE International Conference on Systems, Man, and Cybernetics (SMC)*. IEEE. 2018, pp. 1209–1214.

[75] Nathalie Majcherczyk, Adhavan Jayabalan, Giovanni Beltrame, and Carlo Pinciroli. "Decentralized connectivity-preserving deployment of large-scale robot swarms". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2018, pp. 4295–4302.

[76] Luis Guerrero-Bonilla, David Saldana, and Vijay Kumar. "Design Guarantees for Resilient Robot Formations on Lattices". In: *IEEE Robotics and Automation Letters* 4.1 (2018), pp. 89–96.

[77] Rosario Aragues, Carlos Sagues, and Youcef Mezouar. "Triggered minimum spanning tree for distributed coverage with connectivity maintenance". In: *European Control Conference (ECC)*. IEEE. 2014, pp. 1881–1887.

[78] Anqi Li, Li Wang, Pietro Pierpaoli, and Magnus Egerstedt. "Formally correct composition of coordinated behaviors using control barrier certificates". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2018, pp. 3723–3729.

[79] Michael M Zavlanos and George J Pappas. "Controlling connectivity of dynamic graphs". In: *Proceedings of the 44th IEEE Conference on Decision and Control*. IEEE. 2005, pp. 6388–6393.

[80] Nuzhet Atay and Burchan Bayazit. "Mobile wireless sensor network connectivity repair with k-redundancy". In: *Algorithmic Foundation of Robotics VIII*. Springer, 2009, pp. 35–49.

[81] Alejandro Cornejo and Nancy Lynch. "Fault-tolerance through k-connectivity". In: *Workshop on network science and systems issues in multi-robot autonomy: ICRA*. Vol. 2. 2010, p. 2010.

[82] Nilanjan Chakraborty and Katia Sycara. "Reconfiguration algorithms for mobile robotic networks". In: *IEEE International Conference on Robotics and Automation*. IEEE. 2010, pp. 5484–5489.

[83] Qiang Du, Maria Emelianenko, and Lili Ju. "Convergence of the Lloyd algorithm for computing centroidal Voronoi tessellations". In: *SIAM journal on numerical analysis* 44.1 (2006), pp. 102–119.

[84] Alyssa Pierson, Lucas C Figueiredo, Luciano CA Pimenta, and Mac Schwager. "Adapting to sensing and actuation variations in multi-robot coverage". In: *The International Journal of Robotics Research* 36.3 (2017), pp. 337–354.

[85] Mac Schwager, Daniela Rus, and Jean-Jacques Slotine. "Decentralized, adaptive coverage control for networked robots". In: *The Int'l Journal of Robotics Research* 28.3 (2009), pp. 357–375.

[86] Mac Schwager, Michael P Vitus, Samantha Powers, Daniela Rus, and Claire J Tomlin. "Robust adaptive coverage control for robotic sensor networks". In: *IEEE Transactions on Control of Network Systems* 4.3 (2017), pp. 462–476.

[87] Ruofei Ouyang, Kian Hsiang Low, Jie Chen, and Patrick Jaillet. "Multi-robot active sensing of non-stationary gaussian process-based environmental phenomena". In: *Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems*. International Foundation for Autonomous Agents and Multiagent Systems. 2014, pp. 573–580.

[88] Geoffrey A Hollinger and Gaurav S Sukhatme. "Sampling-based robotic information gathering algorithms". In: *The International Journal of Robotics Research* 33.9 (2014), pp. 1271–1287.

[89] Andreas Krause, Ajit Singh, and Carlos Guestrin. "Near-Optimal Sensor Placements in Gaussian Processes: Theory, Efficient Algorithms and Empirical Studies". In: *Journal of Machine Learning Research (JMLR)* 9 (Feb. 2008), pp. 235–284.

[90] Amarjeet Singh, Andreas Krause, and William J Kaiser. "Nonmyopic Adaptive Informative Path Planning for Multiple Robots." In: *IJCAI*. Vol. 3. 2009, p. 2.

[91] Kian Hsiang Low, John M Dolan, and Pradeep Khosla. "Adaptive multi-robot wide-area exploration and mapping". In: *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems-Volume 1*. International Foundation for Autonomous Agents and Multiagent Systems. 2008, pp. 23–30.

[92] Kai-Chieh Ma, Lantao Liu, and Gaurav S Sukhatme. "Informative planning and online learning with sparse gaussian processes". In: *ICRA*. IEEE. 2017, pp. 4292–4298.

[93] Micah Corah and Nathan Michael. "Efficient Online Multi-robot Exploration via Distributed Sequential Greedy Assignment". In: *Proceedings of Robotics: Science and Systems*. Cambridge, Massachusetts, July 2017. DOI: 10.15607/RSS.2017.XIII.070.

[94] Lauren M Miller and Todd D Murphey. "Trajectory optimization for continuous ergodic exploration". In: *2013 American Control Conference*. IEEE. 2013, pp. 4196–4201.

[95] Lauren M Miller, Yonatan Silverman, Malcolm A MacIver, and Todd D Murphey. "Ergodic exploration of distributed information". In: *IEEE Transactions on Robotics* 32.1 (2016), pp. 36–52.

[96] Volker Tresp. "Mixtures of Gaussian processes". In: *Advances in neural information processing systems*. 2001, pp. 654–660.

[97] Dongbing Gu. "Distributed EM algorithm for Gaussian mixtures in sensor networks". In: *IEEE Transactions on Neural Networks* 19.7 (2008), pp. 1154–1166.

[98] Carl Edward Rasmussen and Christopher KI Williams. *Gaussian processes for machine learning*. Vol. 1. MIT press Cambridge, 2006.

[99] Daniel Pickem, Paul Glotfelter, Li Wang, Mark Mote, Aaron Ames, Eric Feron, and Magnus Egerstedt. "The robotarium: A remotely accessible swarm robotics research testbed". In: *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2017, pp. 1699–1706.

[100] Dingjiang Zhou and Mac Schwager. "Vector field following for quadrotors using differential flatness". In: *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2014, pp. 6567–6572.

[101] Christoph Schöller, Vincent Aravantinos, Florian Lay, and Alois Knoll. "What the constant velocity model can teach us about pedestrian motion prediction". In: *IEEE Robotics and Automation Letters* 5.2 (2020), pp. 1696–1703.

[102] Federico Celi, Li Wang, Lucia Pallottino, and Magnus Egerstedt. "Deconfliction of Motion Paths With Traffic Inspired Rules". In: *IEEE Robotics and Automation Letters* 4.2 (2019), pp. 2227–2234.

[103] Shital Shah, Debadeepta Dey, Chris Lovett, and Ashish Kapoor. "Airsim: High-fidelity visual and physical simulation for autonomous vehicles". In: *Field and service robotics*. Springer. 2018, pp. 621–635.

[104] Horia Mania, Michael I Jordan, and Benjamin Recht. "Active learning for nonlinear system identification with guarantees". In: *arXiv preprint arXiv:2006.10277* (2020).

[105] Ayush Agrawal and Koushil Sreenath. "Discrete Control Barrier Functions for Safety-Critical Control of Discrete Systems with Application to Bipedal Robot Navigation." In: *Robotics: Science and Systems*. 2017.

[106] Fritz John. "Extremum problems with inequalities as subsidiary conditions". In: *Traces and emergence of nonlinear programming*. Springer, 2014, pp. 197–215.

[107] Daniel Russo and Benjamin Van Roy. "An information-theoretic analysis of thompson sampling". In: *The Journal of Machine Learning Research* 17.1 (2016), pp. 2442–2471.

[108] Ali Rahimi and Benjamin Recht. "Random features for large-scale kernel machines". In: *Advances in neural information processing systems* 20 (2007), pp. 1177–1184.

[109] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. "Openai gym". In: *arXiv preprint arXiv:1606.01540* (2016).

[110] Lorenzo Sabattini, Cristian Secchi, Nikhil Chopra, and Andrea Gasparri. "Distributed control of multirobot systems with global connectivity maintenance". In: *IEEE Transactions on Robotics* 29.5 (2013), pp. 1326–1332.

[111] Marco Minelli, Jacopo Panerati, Marcel Kaufmann, Cinara Ghedini, Giovanni Beltrame, and Lorenzo Sabattini. "Self-optimization of resilient topologies for fallible multi-robots". In: *Robotics and Autonomous Systems* 124 (2020), p. 103384.

[112] Robert G. Gallager, Pierre A. Humblet, and Philip M. Spira. "A distributed algorithm for minimum-weight spanning trees". In: *ACM Transactions on Programming Languages and systems (TOPLAS)* 5.1 (1983), pp. 66–77.

[113] David Peleg. "Distributed computing". In: *SIAM Monographs on discrete mathematics and applications* 5 (2000).

[114] Gopal Pandurangan, Peter Robinson, and Michele Scquizzato. "A time- and message-optimal distributed algorithm for minimum spanning trees". In: *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*. ACM. 2017, pp. 743–756.

[115] Michael M Zavlanos, Magnus B Egerstedt, and George J Pappas. "Graph-theoretic connectivity control of mobile robot networks". In: *Proceedings of the IEEE* 99.9 (2011), pp. 1525–1540.

[116] Reinhard Diestel. "Graph Theory". In: *Graduate texts in mathematics* 173 (2012), p. 7.

[117] Joseph Cheriyan and Ramakrishna Thurimella. "Approximating minimum-size k-connected spanning subgraphs via matching". In: *SIAM Journal on Computing* 30.2 (2000), pp. 528–560.

[118] Zeev Nutov. "The k-connected subgraph problem". In: *Handbook of Approximation Algorithms and Metaheuristics: Contemporary and Emerging Applications, Volume 2* (2018).

[119] Arif Khan, Alex Pothen, Md Mostofa Ali Patwary, Nadathur Rajagopalan Satish, Narayanan Sundaram, Fredrik Manne, Mahantesh Halappanavar, and Pradeep Dubey. "Efficient approximation algorithms for weighted b-matching". In: *SIAM Journal on Scientific Computing* 38.5 (2016), S593–S619.

[120] Yuri Boykov and Vladimir Kolmogorov. "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision". In: *IEEE Transactions on Pattern Analysis & Machine Intelligence* 9 (2004), pp. 1124–1137.

[121] Atsuyuki Okabe and Atsuo Suzuki. "Locational optimization problems solved through Voronoi diagrams". In: *European Journal of Operational Research* 98.3 (1997), pp. 445–456.

[122] Tim Mueller-Sim, Merritt Jenkins, Justin Abel, and George Kantor. "The Robotanist: A ground-based agricultural robot for high-throughput crop phenotyping". In: *Robotics and Automation (ICRA), 2017 IEEE International Conference on*. IEEE. 2017, pp. 3634–3639.

[123] *Intel Lab Data*. 2004. URL: `http://db.csail.mit.edu/labdata/labdata.html` (visited on 06/02/2004).

[124] W. Luo, C. Nam, and K. Sycara. "Online decision making for stream-based robotic sampling via submodular optimization". In: *2017 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*. Nov. 2017, pp. 118–123. DOI: 10.1109/MFI.2017.8170416.

[125] Carl Edward Rasmussen and Hannes Nickisch. "Gaussian processes for machine learning (GPML) toolbox". In: *Journal of Machine Learning Research* 11.Nov (2010), pp. 3011–3015.

[126] Niranjan Srinivas, Andreas Krause, Sham M Kakade, and Matthias W Seeger. "Information-theoretic regret bounds for gaussian process optimization in the bandit setting". In: *IEEE Transactions on Information Theory* 58.5 (2012), pp. 3250–3265.

[127] Cyrill Stachniss, Christian Plagemann, and Achim J Lilienthal. "Learning gas distribution models using sparse Gaussian process mixtures". In: *Autonomous Robots* 26.2 (2009), pp. 187–202.

[128] Chendi Lin, Wenhao Luo, and Katia Sycara. "Online Connectivity-aware Dynamic Deployment for Heterogeneous Multi-Robot Systems". In: *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2021.