MULTI-ROBOT COORDINATION AND SAFE LEARNING USING BARRIER CERTIFICATES

A Dissertation
Presented to
The Academic Faculty

By

Li Wang

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Electrical and Computer Engineering

Georgia Institute of Technology

May 2018

MULTI-ROBOT COORDINATION AND SAFE LEARNING USING BARRIER CERTIFICATES

Approved by:

Dr. Magnus Egerstedt, Advisor
School of ECE
Georgia Institute of Technology

Dr. Yorai Wardi
School of ECE
Georgia Institute of Technology

Dr. Patricio Vela
School of ECE
Georgia Institute of Technology

Dr. Evangelos Theodorou
School of AE
Georgia Institute of Technology

Dr. Sam Coogan
School of ECE
Georgia Institute of Technology

Date Approved: March 30, 2018

Stay Hungry. Stay Foolish.

Steve Jobs

To everyone who helped me along the way

ACKNOWLEDGEMENTS

As the Chinese proverb says, “when you drink water, think of its source”. I cannot finish this PhD dissertation without all the help from many individuals along the way.

Firstly, I would like to express my sincere gratitude to my research advisor, Dr. Magnus Egerstedt. As an enthusiastic and knowledgeable deep thinker, he influenced me the most on how to identify interesting research problems and how to come up with the correct methods to approach them. He led me through the confusing and scary stages of PhD study, and then gave me the freedom to explored my own research interests later on. I couldn’t be happier working on the research topics that really interest me in graduate school.

Many of the results presented in this dissertation were possible thanks to the collaboration with several great researchers. I am grateful to my collaborators Dr. Aaron Ames from the California Institute of Technology, Dr. Evangelos Theodorou from aerospace engineering, and Dr. Dongkun Han from the University of Michigan. Their expertises in nonlinear control, machine learning, and optimization have inspired me to explore these exciting research topics in this dissertation.

My journey towards the PhD degree was fun and fruitful thanks to my labmates, Siddharth Mayya, Sebastian Ruf, Paul Glotfelter, Maria Santos, and Gennaro Notomista. I have benefited a lot from the discussions with those senior lab members, Yancy Diaz-Mercado, Matt Hale, Daniel Pickem, and Thiagarajan Ramachandran. It has also been rewarding working with those newer lab members, Ian Buckley, Motoya Ohnishi, Anqi Li, Kyle Slovak, Christopher Banks, Pietro Pierpaoli, and Sean Wilson. They are smart, young, and energetic. I wish them all success in their future career.

Last but not the least, I am grateful to my parents and family, who never stopped believing in me throughout my life. I met my wife Congshan Wan at Georgia Tech, and went through the happiness and hardships in life with her hand in hand. I want to thank her for the love and constant support.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

The objective of this research is to develop a formal safety framework for collision-free and connectivity sustained motion in multi-robot coordination and learning based control. This safety framework is designed with barrier certificates, which provably guarantee the safety of dynamical systems based on the set invariance principle. The barrier certificates are enforced on the system using an online optimization-based controller such that minimal changes to the existing control strategies are required to guarantee safety.

To ensure the safety of multi-robot coordination in a provably correct manner, the barrier certificates are synthesized to explicitly enforce collision free and connectivity sustained motion of multi-robot systems. As the team of robots often needs to perform multiple tasks, Boolean logical compositions of multiple barriers are developed to address multiple objectives. Furthermore, permissive barrier certificates are computed using Sum-of-Squares programming to deal with multiple objectives, which might not be compatible with each other. Experimental implementations of the barrier certificates on the Robotarium, which is a remotely accessible multi-robot testbed at Georgia Institute of Technology, are highlighted.

In the context of safe learning based control, the barrier certificates regulate the way that the learned information is used in the actual controller. The unknown dynamics of the system is learned with Gaussian Processes, which provides a high confidence interval of the system dynamics. The barrier certificates define a high confidence safe region based on the learned system dynamics. As more data of the system dynamics is collected, the barrier certified safe region gradually grows, which means more aggressive system maneuvers are permitted. The safe learning strategy is demonstrated on a 3D nonlinear quadrotor system subject to unknown wind disturbances.

CHAPTER 1
INTRODUCTION

Safety is crucial to many physical dynamical systems, such as autonomous vehicles, industrial robots, and air-traffic control systems [1, 2, 3]. A formal safety framework is developed in this dissertation to ensure collision free and connectivity sustained motion in the context of multi-robot coordination and learning based control. We mainly focus on two problems: in the first we address how to establish provable safety guarantees with minimal impact on the existing controller and ensure simultaneous satisfaction of multiple objectives for teams of robots; In the second problem, we study how to get high confidence safety guarantee for learning based control such that it can be adopted by safety critical dynamical systems.

In order to gain a thorough understanding of these problems, an extensive literature review of the existing methods is conducted in Chapter 2. In particular, both local and global algorithms for multi-robot collision avoidance and connectivity maintenance are discussed. Multiple representative techniques for composing multiple objectives for teams of robots are analyzed. To investigate the application of learning based control methods in safety-critical systems, a number of existing safe learning approaches are discussed and compared in the literature review.

Teams of mobile robots have applications in many areas such as automated warehouse, precision agriculture, autonomous transportation, and environment monitoring. The design of multi-robot coordination strategies is typically concerned with realizing primary, global behaviors, e.g., achieving and maintaining formations, covering areas of interest, environmental exploration, and boundary tracking. Due to the complexity of the higher level controller designs, avoidance behaviors that ensure safety and connectivity of the team are then added as secondary objectives, resulting in a hierarchical composition of multiple objectives. Thus, what is ultimately deployed on the system is a combination of a "formally"

designed nominal controller together with a "wrapped-around" avoidance algorithm. This type of avoidance behavior is often not optimal and sometimes too conservative. As the "robot density" increases, the avoidance behavior might dominate the higher level controller, with the robot spending most of the time avoiding each other and as a result, they do not progress towards achieving the primary objectives, e.g., [4].

In Chapter 3, we explore how to enforce safe motion in teams of wheeled mobile robots with minimal impact to the existing higher level controller using barrier certificates. The key to being able to ensure that the robots avoid collisions is that all potential, pairwise robot-to-robot collisions are accounted for. As such, a centralized version of barrier certificates is constructed first to keep track of all robot pairs and then dictate how the nominal controllers should be modified in order to avoid collisions. Subsequently, the decentralization of the computation is presented to allow the robots themselves to make decisions in real-time. As the safety barrier constraints are designed to be decentralized and use only local sensing information, the lack of a central coordination signal might lead to deadlock among multiple robots with conflicting primary objectives. A novel deadlock-detection scheme in combination with a consistent perturbation method inspired by symmetry-breaking traffic rules are developed. Experimental implementations of the barrier certificates on teams of wheeled mobile robots are presented with highlights on the remotely accessible multi-robot testbed Robotarium.

Due to recent advances in the design, control, and sensing technology, teams of quadrotors have become widely used in aerial robotic platforms, e.g., [5, 6]. Their ability to hover and fly agilely in three dimensional space makes quadrotors effective tools for surveillance, delivery, precision agriculture, search and rescue tasks, see e.g., [7, 8, 9]. When teams of quadrotors are deployed to collaboratively fulfil these higher level tasks, it is crucial to make sure that they do not collide with each other. The focus of Chapter 4 is to rectify the nominal flight trajectory, which is generated with existing control and planning algorithms for teams of quadrotors, in a minimally invasive way to avoid collisions. The differential

flatness [10, 11] property of quadrotor dynamics is leveraged to simplify the motion planning process while still exploiting the nonlinear dynamics (allowing significant deviation from hovering state and large Euler angles) of teams of quadrotors. All collision-free states of the quadrotors are encoded in a safe set. Then, *Safety Barrier Certificates* are synthesized based on the differential flatness property. The feasibility analysis of the certificates are conducted to ensure that a safe control always exists. The developed safety barrier certificates are validated on a team of micro quadrotors with aggressive coordinated flights.

Multi-robot coordination strategies are often designed to achieve team level collective goals, such as covering areas, forming specified shapes, search and surveillance, see e.g. [12, 13, 14]. As the number of robots and the complexity of the task increases, it becomes increasingly difficult to design one single controller that simultaneously achieves multiple objectives, e.g., forming shapes, collision avoidance and connectivity maintenance. Therefore, there is a need to devise a formal approach that can provably compose multiple objectives for the teams of robots.

The goal of Chapter 5 is to introduce compositional barrier functions to enable general compositions of multiple non-negotiable objectives. Methods to compose multiple objectives through AND and OR logical operators are developed, and conditions on which objectives are composable are provided. Secondly, composite safety and connectivity barrier certificates are synthesized with compositional barrier functions, which ensure collision free and connected motion in teams of mobile robots for general coordination tasks. The compositional barrier certificates are implemented experimentally on a multi-robot testbed. Motivated by the need to simultaneously guarantee safety and stability of safety-critical dynamical systems, permissive barrier certificates are constructed to explicitly maximize the region where the system can be stabilized without violating safety constraints.

Safety is crucial to many physical control dynamical systems, such as autonomous vehicles, industrial robots, and air-traffic control systems [1, 2, 3]. To effectively control complex dynamical systems, accurate nonlinear models are typically needed. However,

these models are not always known. In Chapter 6, we present a data-driven approach based on Gaussian processes that learns models of quadrotors operating in partially unknown environments. What makes this challenging is that if the learning process is not carefully controlled, the system will go unstable, i.e., the quadcopter will crash. To this end, barrier certificates are employed for safe learning. The barrier certificates establish a non-conservative forward invariant safe region, in which high probability safety guarantees are provided based on the statistics of the Gaussian Process. A learning controller is designed to efficiently explore those uncertain states and expand the barrier certified safe region based on an adaptive sampling scheme. This safe learning strategy is implemented on a quadrotor flying in an environment with unknown wind disturbance.

A series of Control Barrier Functions (CBF) are introduced or developed in this dissertation to address the safety enforcement problems in different scenarios. A flowchart visualizing their theoretical development and experimental implementations are provided in the following page. We will briefly summarize their technical strength as follows:

- Monotone CBF [15]: to construct barrier certificates and verify safety of systems
- Reciprocal CBF [16]: to ensure forward invariance of the safe set
- Zeroing CBF [17]: to ensure both set forward invariance and asymptotic stability
- Exponential CBF [18]: used for system with high relative degrees
- Piecewise Smooth CBF [19]: to deal with piecewise smooth function definitions
- Compositional CBF [19]: to compose multiple objectives with boolean logic
- Permissive CBF [20]: to maximize the safe region subject to multiple constraints
- Learning CBF [21]: to regulate learning controller for partially modeled system

Other useful classes of CBFs, e.g., discrete CBF [22], nonsmooth CBF [23], are not discussed in this dissertation.

Family of Control Barrier Functions

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

Reciprocal CBF
$\dot{B}(x) \leq \frac{\gamma}{B(x)}$

Zeroing CBF
$\dot{h}(x) \geq -\alpha(h(x))$

Exponential CBF
$L_f^4 h(q) + L_g L_f^3 h(q) v \geq -K\eta$

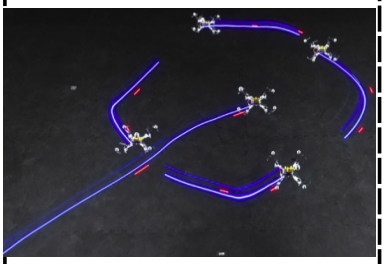
Piecewise Smooth CBF
$-B'(x; -f(x) - g(x)u) \geq -\alpha(B(x))$

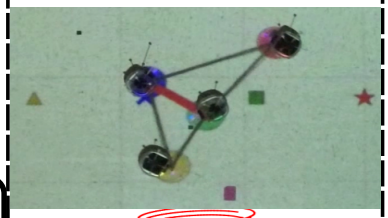
Compositional CBF
$\tilde{B}(x) = \left(\prod_{\substack{i,j \in \mathcal{M} \\ j>i}} B_{ij}(x)\right) \cdot \left(\sum_{k \in \mathcal{P}} \prod_{(i,j) \in E_k} \bar{B}_{ij}(x)\right)$

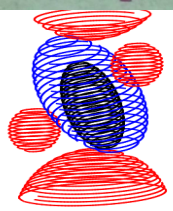
Permissive CBF
$h^*(x) = \underset{h(x) \in \mathcal{P}}{\operatorname{argmax}} \quad \mu(\mathcal{C})$

Learning CBF
$L_f h(x) + L_g h(x) u + \min_{d \in \mathcal{D}(x)} \{\frac{\partial h}{\partial x} d\} \geq -\alpha(h(x))$

Implementations of Control Barrier Functions







CHAPTER 2
LITERATURE REVIEW

Barrier certificates are presented in this dissertation as a formal safety enforcement framework for multi-robot coordination and learning based control. There were extensive work done in related areas of multi-robot collision avoidance, multi-objective compositions, and safety of learning based control. In this chapter, a comprehensive literature survey is conducted on these relevant existing methods.

2.1 Collision Avoidance for Teams of Robots

The methods to address multi-robot collision avoidance are similar to the methods used for single robot case, while the complexity of the problem brings extra challenges. The design of multi-robot coordination strategy is often very complicated, which leads to a divide and conquer scheme in many cases, i.e., the high level coordination controller and low level avoidance controller need to be designed separately and then wrapped together [13, 14, 24]. This type of method includes for example the artificial potential field, the dynamical window approach, the reciprocal velocity obstacle, and the mixed-integer programming approach.

One common problem with this type of wrapped-around avoidance controller is the existence of possible local minima. As a consequence, robots might get trapped at the local minima and make no progress towards their goals. This problem can be overcome by injecting more global information using methods like harmonic potential field and convergent dynamical window. However, these methods are introduced with a price of extra computation burden. Next, we will have a brief review of both the local and global methods.

2.1.1 Artificial Potential Field Approach

The artificial potential field approach defines repulsive potential fields around the obstacles and attractive potential field around the goal [25, 26]. A generalized force was then calculated based on the gradient of the overall potential field. Thus, the robot would be pushed away from the obstacles and dragged towards its goal. The artificial potential field approach is easy to implement and computationally efficient. However, it is often too conservative and might lead to unbounded control effort close to the boundary of the obstacles.

A local minima free version of artificial potential field is called the harmonic potential field [27]. This method is inspired by fluid dynamics and the thermodynamics. By solving the Laplace equation with appropriate boundary conditions, we can get a harmonic potential field with no local minima. But as the complexity of the obstacles increases, solving the PDE numerically become computationally expensive.

2.1.2 Dynamical Window Approach

The dynamical window approach performs collision avoidance in the velocity space [28]. To reduce computation complexity, a finite time window of the velocity space was constructed by considering the dynamical model of the robot. The obstacles in the workspace were also converted into the velocity space of the robot. A best velocity command was then selected in the admissible control space based on the target heading, obstacle clearance, and magnitude of velocity.

To overcome the local minima faced by the dynamical window approach, a convergent dynamical window was developed by adding a navigation function [29]. The collision avoidance problem was solved as a finite horizon optimal control problem. So that the robot would only move in the direction that both avoids collisions and decreases the navigation function. Again, the requirement of a navigation function is a strong assumption, and it is not clear how this method can be extended to general multi-robot coordination tasks.

2.1.3 Reciprocal Velocity Obstacle

The reciprocal velocity obstacle approach [30, 31] used a geometric interpretation to avoid collisions between the robots. The velocity obstacle was defined as the set of all safe relative velocities that will not lead to collisions within a specified amount of time. The assumption of constant velocity for the planning horizon makes it not suitable for highly dynamical collision avoidance maneuvers. Also, it is difficult to generalize the geometric velocity obstacle to the three dimensional case.

2.1.4 Mixed-Integer Programming

The mixed-integer programming method was used to plan collision-free trajectories for teams of heterogeneous quadrotors [32]. The trajectories of the quadrotors were planned as polynomials for simplicity of optimization. The vicinity of the quadrotor was partitioned into several different sides, which introduced the integer decision variables into the optimization problem. This method is easily applicable to heterogeneous team of quadrotors with nonlinear dynamics. But the requirement to discretize the trajectories at each time step makes this method not scalable to large teams of robots.

2.2 Multi-Objective Compositions

Multi-robot coordination strategies are often designed to achieve team level collective goals, such as covering areas, forming specified shapes, search and surveillance. As the number of robots and the complexity of the task increases, it becomes increasingly difficult to design one single controller that simultaneously achieves multiple objectives, e.g., forming shapes, collision avoidance, and connectivity maintenance. There are multiple techniques already developed to composes multiple objectives for the teams of robots. We will discuss about some representative multi-objective composition methods in this section.

2.2.1 Cascaded Filter Approach

The cascaded filter approach composed multi-objective of the robot by sequentially removing undesired control commands from the admissible control set [33]. Multiple objectives of the system were pre-designed as filters. These filters were then cascaded together based on the priority of the objectives; lower priority objectives were forced to choose from the actions that were allowed by the higher level objectives. This method was successfully applied to simultaneously achieve go-to-goal, collision avoidance, and line-of-sight objectives. But this method has no provable guarantee of the feasibility of the controller.

2.2.2 Hybrid Control Method

A hybrid control method was used in [34] to ensure connectivity preserving flocking, and simultaneously achieved alignment, cohesion, and separation. The neighborhood of the agent was partitioned into collision avoidance area, link addition area, and link deletion area. The network topology control schemes were then combined into a hybrid architecture. But this method is a task-specific solution to the connectivity preserving flocking problem. It is not easily applicable to other general networked control problems.

2.2.3 Control Sharing R-function

Different objectives can be encoded into different control Lyapunov functions. If these Lyapunov functions share negative gradient, then all the corresponding objectives can be achieved simultaneously [35, 36, 37]. A representative way to combine Lyapunov functions is to use the "R-functions", i.e.,

$$L_\wedge(x) = (\phi + 1 - \sqrt{\phi^2 + 1})^{-1}(\phi L_1(x) + L_2(x) - \sqrt{\phi^2 L_1(x)^2 + L_2(x)^2}), \tag{2.1}$$

where $\phi > 0$ is the mixing constant, L_1, L_2, and $L_\wedge$ are two Lyapunov functions and one composed Lyapunov function, respectively. Notice that $L_\wedge$ is positive if and only if both L_1

and L_2 are positive.

To apply the Lyapunov function composition technique, it is important to check that the control sharing property is satisfied for every admissible state of the system. But, this property is often difficult to check. Several analytic results for linear control systems and polyhedral/quadratic Lyapunov functions were derived in [35]. The other drawback of the control sharing R-function is that the composed Lyapunov function become very complicated as the number of objectives increases.

2.2.4 Recentered Barrier Function

The recentered barrier function is a set-theoretic approach [38] to compose multiple objectives. A single recentered barrier function was used to unify the go-to-goal behavior, collision avoidance, and proximity maintenance. Each objective is encoded into a constrained set $\mathscr{K}_{ij} = \{(x_i, x_j) \mid c_{ij}(x_i, x_j) \geq 0\}$, which is associated with a logarithmic barrier function $b_{ij} = -\log(c_{ij}(x_i, x_j))$. The barrier function x_{ij} is then recentered based on the goal of the robot,

$$x_{ij} = b_{ij}(x_i, x_j) - b_{ij}(x_{id}, x_j) - \nabla b_{ij}(x_{id}, x_j)^T \delta x_i, \tag{2.2}$$

where x_{id} is the goal position of robot i. By recentering the barrier function, it is non-zero everywhere in the constrained set except at the goal position. With this property, the team of robots will reach their goal positions without colliding with each other. However, the recentered barrier function was specifically constructed for go-to-goal task, and thus can not be extended to complex situations easily.

2.2.5 Barrier Lyapunov Function

Multiple objectives can also be achieved simultaneously by uniting the control barrier function with control Lyapunov function, which yields the Barrier Lyapunov function [39]. The

Barrier Lyapunov function was designed such that it grows to infinity when some safety arguments approach undesirable limits, and its derivative is always negative semi-definite [40]. With the Barrier Lyapunov function $W(x)$, the stabilization property of the CLF $V(x)$ and the safety guarantee of the CBF $B(x)$ can be ensured simultaneously,

$$W(x) = V(x) + \lambda B(x) + \kappa, \tag{2.3}$$

where λ and κ are design parameters to shift the equilibrium point and gradient of the system. The construction of the Barrier Lyapunov function requires careful design of the parameters with respect to different bounds on the unsafe region, barrier function, and Lyapunov function. It also assumes the system has small control property and appropriately lower bounded barrier function. These assumptions and design constraints make it both conservative and difficult to apply Barrier Lyapunov functions to multi-robot systems.

2.2.6 Sum-of-Squares Programming

The Lyapunov and barrier constraints are often formulated as non-negativity constraints which is difficult to verify, since checking non-negativity is often computationally intractable [41]. However, if we restrict interested systems to polynomial dynamical systems, Sum-of-Squares (SOS) programming technique can be used to greatly simply the computation. When non-negativity constraints are relaxed to SOS constraints, these complex optimization problems can be converted to numerically efficient convex problems.

Let $\mathscr{P}$ be the set of polynomials for $x \in \mathbb{R}^n$. The polynomial $l(x)$ can be written in Square Matrix Representation (SMR) [42] as $Z^T(x)QZ(x)$, where $Z(x)$ is a vector of monomials, and $Q \in \mathbb{R}^{k\times k}$ is a symmetrical coefficient matrix. A polynomial function $l(x)$ is nonnegative if $l(x) \geq 0, \forall x \in \mathbb{R}^n$. Furthermore, $p(x)$ is a SOS polynomial if

$$p(x) = \sum_{i=1}^{m} p_i^2(x)$$

for some $p_i(x) \in \mathscr{P}$. If written in SMR form, $p(x)$ has a positive semidefinite coefficient matrix $Q \succeq 0$.

SOS is used to formulate computationally efficient solutions to problems that contains multiple constraints. For instance, safe stabilization funnels based on sublevel sets of the Lyapunov function were calculated with SOS in [43]. Typical Domain of Attraction estimation [42] and barrier certificates design [15] methods all rely on SOS frameworks.

2.3 Safety of Learning Based Control

The existence of model inaccuracies and unknown disturbances create a great challenge to the design of safe controllers for safety critical systems. Tools such as robust control and adaptive control methods have been developed in classic control theory to ensure the safety and stability of the system, see [44, 45] and the references therein. Meanwhile, machine learning based control approaches are becoming increasingly popular as a way to deal with inaccurate models [46, 47], due to their abilities to infer unknown models from data and actively improve the performance of the controller with the learned model. Learning based control approaches require only limited expert knowledge and fewer assumptions about the system [48]. However, there always exists an inherent trade-off between safety and performance in these methods [49]. Data-driven learning approaches rarely provides safety guarantees, which limits their applicability to real-world safety critical control dynamical systems [48].

In order to promote the application of learning based control methods in safety-critical systems, a number of safe learning approaches [50, 51, 52, 53, 54, 55, 56] have been proposed in the literature.

2.3.1 Lyapunov Based Approach

Among these methods, the use of learning Control Lyapunov Functions (CLF) is shown to be a promising approach. A learning from demonstration method was developed in

[57] to search for a CLF from several demonstrations, and the learned CLF was used to stabilize the system. But the learned controller did not consider actuator limits and other safe operation constraints. [58] introduced a verifier to explicitly validate the learned CLF. However, when the model of the system is inaccurate, the verifier needs to check an infinite number of inequalities throughout the state space, which is computationally difficult [59]. [60] seeks to learn CLF and maximize the safe operation region for the system with GP model. High probability safety guarantees are provided based on Lyapunov stability and GP statistics.

2.3.2 Reachability Analysis

A reachability-based safe learning approach was presented in [61] to reduce the conservativeness of reachability analysis by learning the disturbance from data. The safe constraint set $\mathscr{K}$ is a compact set that the system state should not leave. With the Hamilton-Jacobi-Isaacs (HJI) reachability analysis, the discriminating kernel $Disc_{\mathbb{T}}(\mathscr{K},\mathscr{D})$ of $\mathscr{K}$ with respect to the disturbance set $\mathscr{D}$ and a time horizon $\mathbb{T}=[0,\tau]$ can be computed. The discriminating kernel $Disc_{\mathbb{T}}(\mathscr{K},\mathscr{D})$ contains all initial states which can be contained within $\mathscr{K}$ by a feasible controller for any disturbance from $\mathscr{D}$ for some time horizon $\tau>0$. The disturbance set $\mathscr{D}$ is shrunk based on the collected data over time to reduce the conservativeness of the learning algorithm. To ensure the safety of the learning process, a safety preserving action is only required near the boundary of $Disc_{\mathbb{T}}(\mathscr{K},\mathscr{D})$. However, due to the complexity of the HJI reachability analysis, the convergence of the computation of $Disc_{\mathbb{T}}(\mathscr{K},\mathscr{D})$ for a sufficiently large time horizon is assumed as a prior.

CHAPTER 3

BARRIER CERTIFICATES FOR WHEELED MOBILE ROBOTS

The safety barrier certificates are synthesized with control barrier functions (CBF) in this chapter to ensure the safety of the multi-robot team. We will first review some fundamentals behind CBFs and then retool them to ensure that teams of robots are collision-free.

3.1 Control Barrier Functions

The basic idea of CBF is to define a set of safe states and then use the CBF to formally guarantee the forward invariance of the desired set, i.e., if the system starts in the safe set, it stays in the safe set [62, 63]. There are multiple types of CBFs in the literature [40, 62, 63]. We will focus on the zeroing control barrier function (ZCBF), since it comes with both non-conservative set invariance and robustness properties [64, 65].

For the sake of generality, we first consider dynamical systems on control affine form

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

where the state $x \in \mathbb{R}^n$ and control $u \in U \subset \mathbb{R}^m$, f and g are locally Lipschitz continuous. Let the set $\mathscr{C} = \{x \in \mathbb{R}^n \mid h(x) \geq 0\}$ be the safe set, where $h : \mathbb{R}^n \to \mathbb{R}$ is a ZCBF candidate function. We note that $\frac{dh(x)}{dt} = \frac{\partial h(x)}{\partial x}\dot{x} = \frac{\partial h(x)}{\partial x}(f(x) + g(x)u)$, or, using the Lie derivative formalism $\frac{dh(x)}{dt} = L_f h(x) + L_g h(x)u$.

Definition 3.1.1 Given a dynamical system (3.1) and a set $\mathscr{C} \subset \mathbb{R}^n$ for a smooth function $h : \mathscr{D} \to \mathbb{R}$, with $\mathscr{C} \subseteq \mathscr{D} \subset \mathbb{R}^n$. h is called a Zeroing Control Barrier Function (ZCBF), if there exists an extended class-$\mathscr{K}$ function κ (strictly increasing, $\kappa(0) = 0$) such that

$$\sup_{u \in U} \{L_f h(x) + L_g h(x)u + \kappa(h(x))\} \geq 0,$$

for all $x \in \mathscr{D}$.

Given a ZCBF $h(x)$, the admissible control space $S(x)$ can be defined as

$$S(x) = \left\{u \in U \mid L_f h(x) + L_g h(x) u + \kappa(h(x)) \geq 0\right\}, \; x \in \mathscr{D},$$

To guarantee that $\mathscr{C}$ is forward invariant, we can use the following theorem.

Theorem [62]. *Given a set $\mathscr{C} \subset \mathbb{R}^n$ and a ZCBF h defined on $\mathscr{D}$, with $\mathscr{C} \subseteq \mathscr{D} \subset \mathbb{R}^n$, any Lipschitz continuous controller u: $\mathscr{D} \to \mathbb{R}$ such that $u \in S(x)$ for the system (3.1) renders the set $\mathscr{C}$ forward invariant. And $\mathscr{C}$ is asymptotically stable in $\mathscr{D}$.*

By allowing the derivative of the barrier certificate to grow within the safe set $\mathscr{C}$, this barrier certificate can ensure the forward invariance of $\mathscr{C}$ in a non-conservative manner as shown in Fig. 3.1. Since the barrier certificates synthesized with ZCBF provide a non-conservative way to ensure provable safety, they can contribute to various useful controls applications.

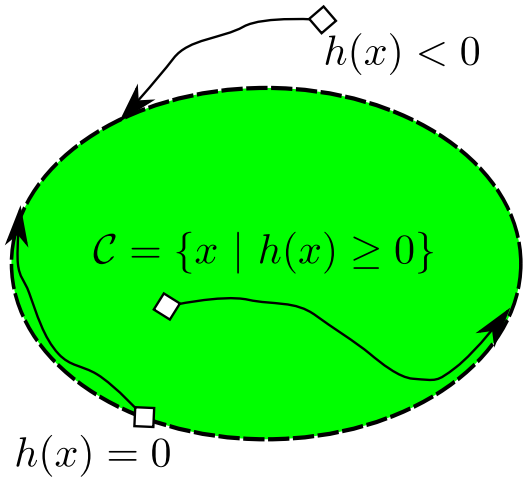


Figure 3.1: Examples of non-conservative CBFs for set invariance. The diamonds and curves are example initial states and allowed state trajectories, respectively. The state is allowed to grow or even approach the boundary from inside the safe region. Outside the safe region, the state will converge asymptotically to the safe region, due to CBF constraints.

The improvement of barrier certificates based on ZCBF can be illustrated with a simple example. Using the SOS technique described in [15], we can compute the certified safe

regions for two types of barrier certificates. The first type of barrier certificates is based on a strictly non-increasing barrier function ($\dot{h} \geq 0$), while the second type is based on ZCBF ($\dot{h} \geq -\kappa(h(x))$).

Consider a 2D autonomous dynamical system,

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ -x_1 + \frac{1}{3}x_1^3 - x_2 \end{bmatrix}.$$

The initial and unsafe sets are specified as $\mathscr{X}_0 = \{x \mid 0.25 - (x_1 - 1.5)^2 - (x_2 + 1)^2 \geq 0\}$ and $\mathscr{X}_u = \{x \mid 0.25 - (x_1 + 1.4)^2 - (x_2 + 1.6)^2 \geq 0\}$, respectively. Both types of barrier certificates can be illustrated in Fig. 3.2. The area of the barrier certified safe region generated with $\dot{h} \geq -\kappa(h(x))$ is much larger than $\dot{h} \geq 0$, which means that $\dot{h} \geq -\kappa(h(x))$ allows for a significantly more permissive safety certificate than $\dot{h} \geq 0$.

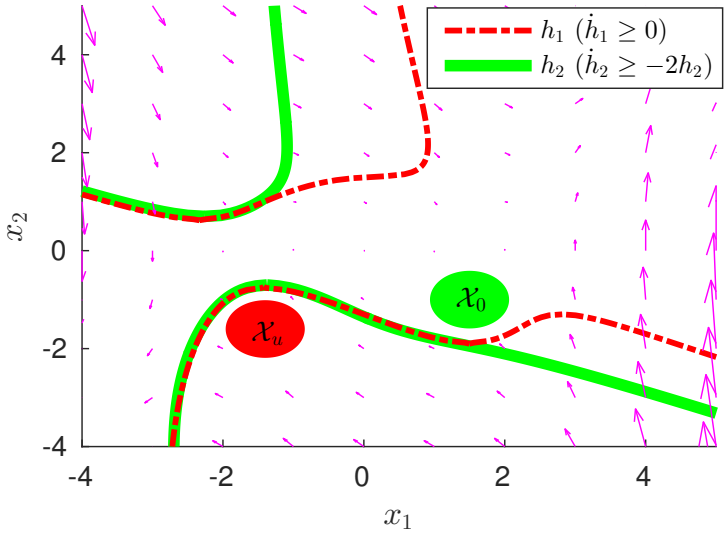


Figure 3.2: Comparison of two types of barrier certificates. The barrier certified safe region based on $\dot{h} \geq -\kappa(h(x))$ (area between the solid green lines) is significantly larger than the safe region based on $\dot{h} \geq 0$ (area between the dashed red lines). $\mathscr{X}_0$ and $\mathscr{X}_u$ are the initial and unsafe set, respectively.

3.2 Centralized Safety Barrier Certificates

Consider a multi-robot system consisting of N planar, mobile robots, indexed by $\mathscr{M} = \{i \mid i = 1, 2, \ldots, N\}$. We model the robot dynamics as double integrators

$$\begin{bmatrix} \dot{p}_i \\ \dot{v}_i \end{bmatrix} = \begin{bmatrix} 0 & I_{2\times 2} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} p_i \\ v_i \end{bmatrix} + \begin{bmatrix} 0 \\ I_{2\times 2} \end{bmatrix} u_i, \tag{3.2}$$

where $p_i \in \mathbb{R}^2$, $v_i \in \mathbb{R}^2$, and $u_i \in \mathbb{R}^2$ represent the positions, velocities, and inputs (acceleration commands) of agent i respectively. The velocity and acceleration of agent i are limited by $\|v_i\|_\infty \leq \beta_i$ and $\|u_i\|_\infty \leq \alpha_i$. The aggregate states and inputs of all N agents are denoted as $(p, v) \in \mathbb{R}^{4N}$ and $u \in \mathbb{R}^{2N}$.

Next, a pairwise robot-to-robot safety constraint is formulated to guarantee that a safety distance D_s between any two agents can be ensured. Algorithms to avoid imminent collision with static obstacles were developed in [28, 29] by decelerating the agent to zero velocity with the maximum braking force. However, to avoid imminent collision with a moving agent, the relative velocity between two agents needs to be reduced to zero instead of the absolute velocity. Consider any two agents i and j, the relative position and relative velocity between them are $\Delta\mathbf{p}_{ij} = \mathbf{p}_i - \mathbf{p}_j$ and $\Delta\mathbf{v}_{ij} = \mathbf{v}_i - \mathbf{v}_j$. As illustrated in Fig. 3.3, the normal component of the relative velocity ($\Delta\bar{v} = \|\Delta\dot{\mathbf{p}}_{ij}\| = \frac{\Delta\mathbf{p}_{ij}^T}{\|\Delta\mathbf{p}_{ij}\|}\Delta\mathbf{v}_{ij}$) is the actual component that might lead to collision between agents i and j, while the tangent component of $\Delta\mathbf{v}_{ij}$ only leads to rotation around each other. Therefore, we need to regulate $\Delta\bar{v}$ so that imminent collisions can be avoided if the maximum relative braking force is applied.

Assuming the normal component of the relative velocity between agents i and j is $\Delta\bar{v}(t_0)$ at the current time instance t_0, it takes the time $T_b = \frac{0-\Delta\bar{v}(t_0)}{\alpha_i+\alpha_j}$ to reach $\Delta\bar{v}(t_0+T_b) = 0$, while the maximum braking acceleration $(\alpha_i + \alpha_j)$ is applied to both robots. In order to remain farther away from the safety distance D_s, the following safety constraint needs to

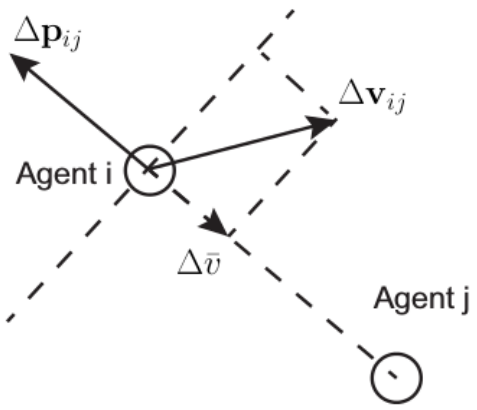


Figure 3.3: Relative position and velocity between two agents

be satisfied,

$$\|\Delta\mathbf{p}_{ij}\| + \int_{t_0}^{t_0+T_b} \Delta\bar{v}(t_0+t)\,\mathrm{d}t \geq D_s, \quad \forall\, i \neq j,$$

where $\Delta\bar{v}(t_0+t) = \Delta\bar{v}(t_0) + (\alpha_i+\alpha_j)t$, which means that

$$\|\Delta\mathbf{p}_{ij}\| - \frac{(\Delta\bar{v})^2}{2(\alpha_i+\alpha_j)} \geq D_s, \quad \forall\, i \neq j. \tag{3.3}$$

Note that this safety constraint only needs to be enforced when agents are moving closer to each other, i.e., when $\Delta\bar{v} \leq 0$. It is always considered safe when the agents are moving away from each other, i.e., when $\Delta\bar{v} > 0$. By combining this observation with the constraint in (3.3) gives

$$-\frac{\Delta\mathbf{p}_{ij}^T}{\|\Delta\mathbf{p}_{ij}\|}\Delta\mathbf{v}_{ij} \leq \sqrt{2(\alpha_i+\alpha_j)(\|\Delta\mathbf{p}_{ij}\| - D_s)}, \ \forall\, i \neq j.$$

As such, the pairwise safe set $\mathscr{C}_{ij}$ is defined as

$$\begin{aligned} \mathscr{C}_{ij} &= \{(p_i, v_i) \in \mathbb{R}^4 \mid h_{ij}(p,v) \geq 0\}, \ \forall\, i \neq j, \\ h_{ij}(p,v) &= \sqrt{2(\alpha_i+\alpha_j)(\|\Delta p_{ij}\| - D_s)} + \frac{\Delta p_{ij}^T}{\|\Delta p_{ij}\|}\Delta v_{ij}, \end{aligned} \tag{3.4}$$

where $h_{ij}(p,v)$ is the level set function of the set $\mathscr{C}_{ij}$ as well as the ZCBF candidate used

to ensure the forward invariance of $\mathscr{C}_{ij}$. To make the safe set forward invariant, the safety barrier constraint can be written as

$$-\Delta p_{ij}^T \Delta u_{ij} \leq \gamma h_{ij}^3 \|\Delta p_{ij}\| - \frac{(\Delta v_{ij}^T \Delta p_{ij})^2}{\|\Delta p_{ij}\|^2} + \|\Delta v_{ij}\|^2 + \frac{(\alpha_i + \alpha_j)\Delta v_{ij}^T \Delta p_{ij}}{\sqrt{2(\alpha_i + \alpha_j)(\|\Delta p_{ij}\| - D_s)}}.$$

This safety barrier constraint can be written as a linear constraint in u_i and u_j, which in turn can be represented as $A_{ij}u \leq b_{ij}$, with

$$A_{ij} = [0, ..., \underbrace{-\Delta p_{ij}^T}_{\text{agent } i}, ..., \underbrace{\Delta p_{ij}^T}_{\text{agent } j}, ..., 0],$$

and $b_{ij} = \gamma h_{ij}^3 \|\Delta p_{ij}\| - \frac{(\Delta v_{ij}^T \Delta p_{ij})^2}{\|\Delta p_{ij}\|^2} + \frac{(\alpha_i + \alpha_j)\Delta v_{ij}^T \Delta p_{ij}}{\sqrt{2(\alpha_i + \alpha_j)(\|\Delta p_{ij}\| - D_s)}} + \|\Delta v_{ij}\|^2$.

We denote all pairwise safety barrier constraints as the *centralized safety barrier certificates* for the multi-robot system, i.e.,

$$S_u = \{u \in \mathbb{R}^{2N} \mid A_{ij}u \leq b_{ij}, \forall\, i \neq j\}. \tag{3.5}$$

The safe set $\mathscr{C}$ for the overall system is now formally defined as,

$$\mathscr{C} = \prod_{i \in \mathscr{M}} \left\{ \bigcap_{\substack{j \in \mathscr{M} \\ j \neq i}} \mathscr{C}_{ij} \right\},$$

where the product is the Cartesian product over the state space of all agents. The following result is presented to ensure the safety of the multi-robot system.

Theorem 3.2.1. *Given a multi-robot system indexed by $\mathscr{M}$ with dynamics in (3.2), if the controller u satisfies the centralized safety barrier certificates in (3.5), and $(p(0), v(0)) \in \mathscr{C}$, then the multi-robot system is guaranteed to be safe.*

Proof. If the controller satisfies the centralized safety barrier certificates, then $\mathbf{u}(t)$ is always constrained inside the admissible control space S_u and satisfies all the pairwise safety

barrier constraints. As ensured by the ZCBFs in [62], $\mathscr{C}_{ij}$ is forward invariant for all $i \neq j$, i.e., $\mathscr{C}$ is forward invariant. Since $(\mathbf{p}(0), \mathbf{v}(0)) \in \mathscr{C}$, $(\mathbf{p}(t), \mathbf{v}(t))$ will stay in $\mathscr{C}$ for all time. Hence, the multi-robot system is guaranteed to be *safe*. □

Minimally Invasive Collision Avoidance using a QP-based Controller

The QP-based controller which minimizes the difference between the actual control command u_i and nominal control command $\hat{u}_i$, while ensuring safety using the *centralized safety barrier certificates*, is formulated as,

$$\begin{aligned} u^* = \underset{u \in \mathbb{R}^{2N}}{\operatorname{argmin}} \quad & J(u) = \sum_{i=1}^{N} \|u_i - \hat{u}_i\|^2 \\ \text{s.t.} \quad & A_{ij} u \leq b_{ij}, \qquad \forall\, i \neq j, \\ & \|u_i\|_\infty \leq \alpha_i, \qquad \forall\, i \in \mathscr{M}. \end{aligned} \tag{3.6}$$

The resulting controller u mimics the nominal controller $\hat{u}$ completely when the system is safe, and only modifies its behavior when collisions are truly imminent.

Reduced Neighborhoods

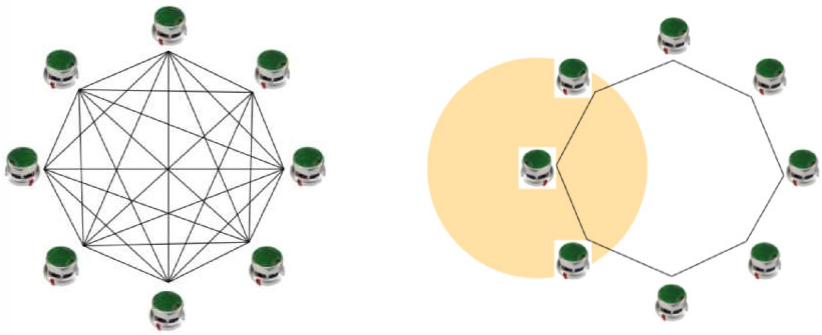
The *centralized safety barrier certificates* in the previous section considers *all* pairs of robots, which is potentially a very large number. Topologically speaking, that requires all-to-all interactions, i.e., a complete graph. As a result, the associated computation and sensing requirements will increase significantly as the number of robots increases. Motivated by the fact that agents sufficiently far apart will not collide within a finite time horizon, a neighborhood notion should be developed that reduces the required information structure to a disk graph, i.e., only pairs of nearby (within a certain distance) robots are needed, as shown in Fig. 3.4. The neighborhood set of agent i is thus defined as

$$\mathscr{N}_i = \{ j \in \mathscr{M} \mid \|\Delta \mathbf{p}_{ij}\| \leq D^i_{\mathscr{N}}, j \neq i \}, \tag{3.7}$$

where

$$D^i_{\mathscr{N}} = D_s + \frac{1}{2(\alpha_i + \alpha_{\min})}\left(\sqrt[3]{\frac{2(\alpha_i + \alpha_{\max})}{\gamma}} + \beta_i + \beta_{\max}\right)^2$$

is the choice of the radius of the neighborhood which will be elucidated later, $\alpha_{\min} = \min_{j\in\mathscr{M}}\{\alpha_j\}$ and $\alpha_{\max} = \max_{j\in\mathscr{M}}\{\alpha_j\}$ are lower and upper bounds of all agents' acceleration limits, and $\beta_{\max} = \max_{j\in\mathscr{M}}\{\beta_j\}$ is the upper bound of all agents' speed limits.



(a) Complete graph (b) Disk graph

Figure 3.4: Reduced information requirement graph

With this notion of neighborhood, we will say that each agent only needs to consider its nearby agents to avoid collision, even though the computation, so far, is done by a centralized unit. This, however, will be related in subsequent sections. A similar notion for agents with identical acceleration limits was derived in [66], and here safety barrier certificates are synthesized with ZCBFs for agents with different acceleration limits.

Theorem 3.2.2. *Agent $i \in \mathscr{M}$ only needs to form ZCBFs with its neighbors, as defined in (3.7), to guarantee safety.*

Proof. See Appendix □

With **Theorem** 3.2.2, the QP-based controller (6.13) can be simplified by only checking

the safety of a multi-robot system with disk information graph.

$$\begin{aligned} \mathbf{u}^* = \underset{\mathbf{u}\in\mathbb{R}^{2N}}{\operatorname{argmin}} \quad & J(\mathbf{u}) = \sum_{i=1}^{N} \|\mathbf{u}_i - \hat{\mathbf{u}}_i\|^2 \\ \text{s.t.} \quad & A_{ij}\mathbf{u} \leq b_{ij}, \qquad \forall\, i \in \mathscr{M},\ \forall\, j \in \mathscr{N}_i, \\ & \|\mathbf{u}_i\|_\infty \leq \alpha_i, \qquad \forall\, i \in \mathscr{M}. \end{aligned} \tag{3.8}$$

The radius $D^i_{\mathscr{N}}$ of the disk information graph can be designed by choosing appropriate γ, such that $D^i_{\mathscr{N}}$ is no larger than the sensing range of agents. Note that this notion of neighborhood is still valid when the *safety barrier certificates* are distributed to individual agent.

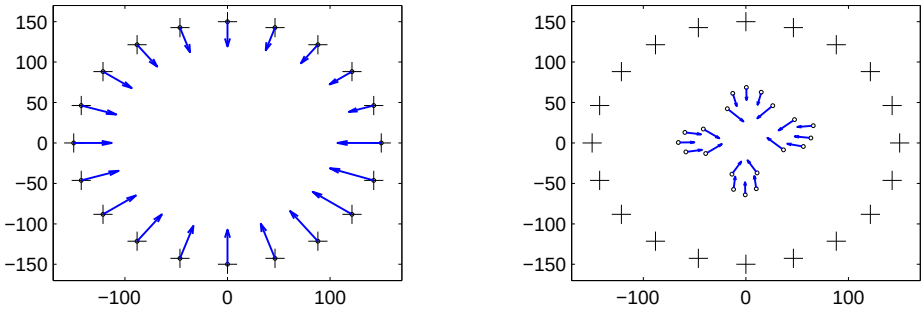
Simulated Centralized Safety Barrier Certificates

The centralized safety barrier certificates are validated on a simulated multi-robot system consisting of 20 agents modelled with double integrator dynamics. The nominal controller is designed to make all agents swap their positions with the agents on the opposite side. With the safety barrier certificates, all robots successfully navigated through the "crowded" region and swapped positions without colliding into each other as shown in Fig. 3.5.

3.3 Decentralized Safety Barrier Certificates

The centralized safety barrier certificates ensure provably safe multi-robot coordination, while the reliance on a central coordination unit potentially compromises the multi-robot system's scalability, reactiveness, and robustness. To address those issues, the safety barrier certificates can be distributed to individual agents without losing the safety guarantee.

$$\begin{aligned} -\Delta p_{ij}^T u_i &\leq \frac{\alpha_i}{\alpha_i + \alpha_j} b_{ij}, \\ \Delta p_{ij}^T u_j &\leq \frac{\alpha_j}{\alpha_i + \alpha_j} b_{ij}. \end{aligned}$$



(a) Time step 3 (b) Time step 325

(c) Time step 770 (d) Time step 1000

Figure 3.5: Simulation results of a multi-robot position swapping task regulated by the centralized safety barrier certificates. The circles and arrows represent the current positions and velocities of the agents. The safety distance $D_s = 10$.

For more details about decentralized safety barrier certificates for heterogeneous multi-robot systems, we refer the reader to [64]. With the decentralized safety barrier constraints and the notion of neighborhood, the sensing and computation tasks are completely distributed to each individual agent. Each agent $i \in \mathscr{M}$ runs their own QP-based controller,

$$\begin{aligned} u_i^* = \underset{u_i \in \mathbb{R}^2}{\text{argmin}} \quad & J(u_i) = \|u_i - \hat{u}_i\| \\ \text{s.t.} \quad & \bar{A}_{ij} u_i \leq \bar{b}_{ij}, \qquad \forall\, j \in \mathscr{N}_i, \\ & \|u_i\|_\infty \leq \alpha_i, \end{aligned} \tag{3.9}$$

where $\bar{A}_{ij} = -\Delta p_{ij}^T$, $\bar{b}_{ij} = \frac{\alpha_i}{\alpha_i + \alpha_j} b_{ij}$.

When the safety barrier certificates are distributed to each individual agents, the safety of the multi-robot system is still guaranteed by the following result.

Theorem 3.3.1. *Given a multi-robot system indexed by $\mathscr{M}$ with dynamics in (3.2), if the*

controller u_i satisfies the decentralized safety barrier certificates in (3.9) for all agent $i \in \mathscr{M}$, and $(p(0), v(0)) \in \mathscr{C}$, then the multi-robot system is guaranteed to be safe.

Proof. If all agents' controllers satisfy the decentralized safety barrier certificates, then $\mathscr{C}_{ij}$ is forward invariant $\forall i \in \mathscr{M}$, $j \in \mathscr{N}_i$, as ensured by the ZCBFs. When $j \notin \mathscr{N}_i$, $(\mathbf{p}_i, \mathbf{v}_i)$ still stays in $\mathscr{C}_{ij}$ due to **Theorem** 3.2.2. Therefore, $\mathscr{C}$ is forward invariant, and this completes the proof. □

Although safety is still ensured, the collision avoidance interventions enforcing the decentralized safety certificates have to happen earlier than the centralized case due to the lack of central coordination.

Computational Complexity and Solver Details

The computational complexity of the QP for enforcing the safety barrier certificates is analyzed in this section. The number of decision variables M_d and the number of linear constraints M_c are two important factors that determine the computation complexity of the QP. As shown in Table 3.1, the decentralized safety barrier certificates are scalable to arbitrarily large groups of robots.

Table 3.1: Computational Complexity of the Certificates

Type of Certificate	M_d	M_c (worst case)
Centralized	$2N$	$\frac{N(N-1)}{2}$
Centralized with Neighborhood	$2N$	$\frac{N}{2}\min\{N-1, ceiling(\frac{D_{\mathscr{M}}^2}{D_s^2})\}$
Decentralized	2	$N-1$
Decentralized with Neighborhood	2	$\min\{N-1, ceiling(\frac{D_{\mathscr{M}}^2}{D_s^2})\}$

The actual computation times of the certificates per iteration are listed in Table 3.2. The decentralized barrier certificates can handle more than 100 robots with an update rate

of more than 100 Hz. All the computations are performed on an Ubuntu laptop with a 2.60 GHz Intel Core i5 processor using the MATLAB *quadprog* solver.

Table 3.2: Computation Time of the Certificates

Type of Certificate	Computation Time per Iteration (ms)		
	$N = 20$	$N = 60$	$N = 100$
Centralized with Neighborhood	11.8	28.8	238.3
Decentralized with Neighborhood	6.00	5.99	8.05

3.4 Consistent Perturbation for Deadlock Resolution

When the objectives of multiple agents conflict with the safety barrier certificates, the agents might get stuck into a *deadlock*. In the deadlock scenario, the agents are safe but their tasks can not be completed. Deadlock occurs because the safety barrier certificates are designed to take the local information only. In order to detect and resolve the deadlock issue, we first come up with a definition of the deadlock.

Definition VII.1. A robot agent i is said to be stuck in a deadlock, if it remains stationary ($u_i = 0$ and $v_i = 0$) and the nominal control command $\hat{u}_i \neq 0$.

With this definition, the deadlock scenarios can be further classified into three types based on the solution to the QP problem in (3.9). The admissible control space for the QP problem is a convex polygon $\mathscr{P}_i$ defined as the intersection of multiple half spaces, i.e.,

$$\mathscr{P}_i = \{\mathbf{u}_i \in \mathbb{R}^2 \mid \bar{A}_{ij}\mathbf{u}_i \leq \bar{b}_{ij}, \forall\, i \neq j\},$$

where $\mathscr{P}_i$ is a decentralized counterpart of the centralized admissible control space S_u in 3.5. The size of the feasible control space, termed the *width of the feasible set* [67], can be

evaluated with a Linear Program (LP),

$$\begin{aligned} \min_{\mathbf{u}_i \in \mathbb{R}^2, \delta_{LP} \in \mathbb{R}} \quad & \delta_{LP} \\ \text{s.t.} \quad & \bar{A}_{ij}\mathbf{u}_i \leq \bar{b}_{ij} + \delta_{LP}, \qquad \forall\, i \neq j, \\ & \|\mathbf{u}_i\|_\infty \leq \alpha_i. \end{aligned}$$

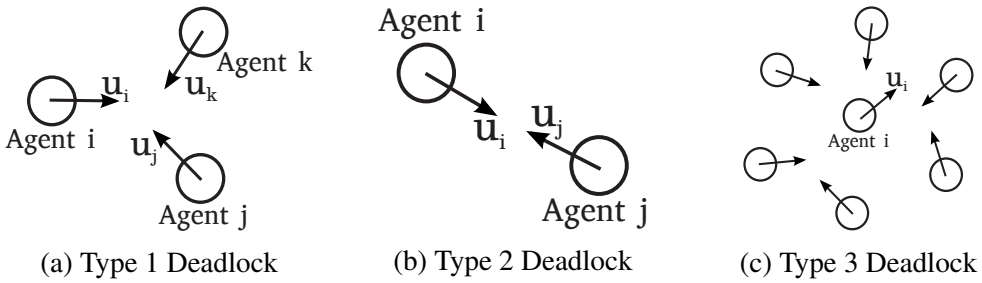
The solution of the LP characterizes how much control margin is left for the strictest safety barrier constraint. If $\delta_{LP} \leq 0$, the corresponding QP is solvable. A more negative δ_{LP} indicates larger admissible control space. Otherwise, no feasible control option is available, and the admissible control space is empty.

The deadlock scenarios are categorized into the following three cases based on the relation between u_i and $\mathscr{P}_i$

1. Type 1 Deadlock: $\delta_{LP} < 0, u_i = 0 \in \text{vertex}(\mathscr{P}_i)$;
2. Type 2 Deadlock: $\delta_{LP} < 0, u_i = 0 \in \text{edge}(\mathscr{P}_i)$;
3. Type 3 Deadlock: $\delta_{LP} \geq 0$.

It should be noted that these three types of deadlock comprise all possible types of deadlocks. This is because u_i is either on the edge or the vertex of $\mathscr{P}_i$, when the optimal solution of the constrained QP controller ($u_i = 0$) is different from the unconstrained optimal solution ($\hat{u}_i \neq 0$), due to Karush-Kuhn-Tucker (KKT) conditions [68].

More intuitively, these three deadlock scenarios are illustrated in Fig. 3.6.



(a) Type 1 Deadlock (b) Type 2 Deadlock (c) Type 3 Deadlock

Figure 3.6: Three types of Deadlocks for robot agent i in a multi-robot system.

One way to resolve the deadlock scenarios is to perturb the QP controller so that the robot agents can move around each other when they get stuck. The QP controller needs to be perturbed consistently, because multiple robot agents might still be acting against each other with random perturbations. Inspired by the traffic rule used in transportation to resolve conflicts [69, 70], the following consistent perturbation method is proposed to resolve different deadlocks

1. Type 1 Deadlock: As illustrated in Fig. 3.7a, the left barrier constraint is relaxed ($k_{\gamma(\text{left})} > 1$) and the right barrier constraint is compressed ($k_{\gamma(\text{right})} < 1$).

2. Type 2 Deadlock: As illustrated in Fig. 3.7b, $\hat{\mathbf{u}}_i$ is perturbed with $\delta^{\perp} = k_{\delta} \left[\begin{smallmatrix} 0 & -1 \\ 1 & 0 \end{smallmatrix}\right] \hat{u}_i$, which is a normal perturbation to the left of $\hat{u}_i$.

3. Type 3 Deadlock: $\mathbf{u}_i = 0$, no perturbation is performed since no admissible perturbation is available.

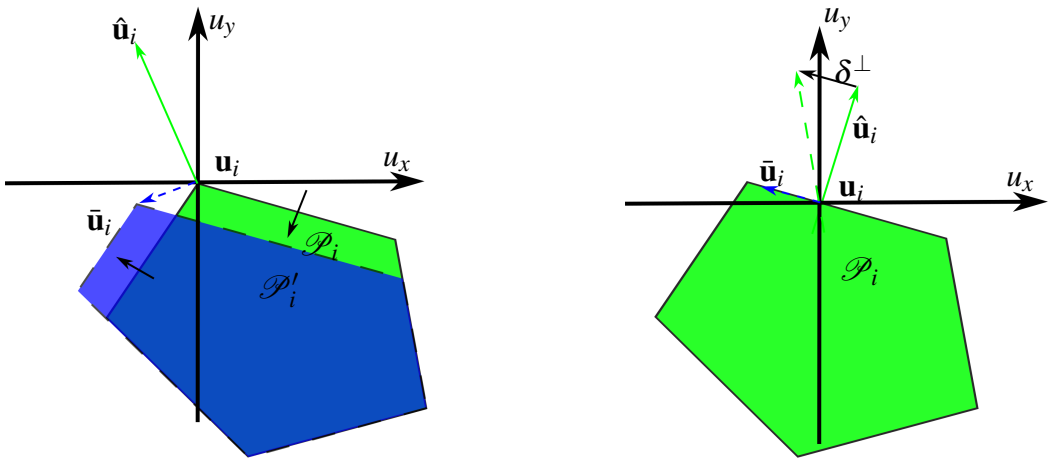
Note that the online relaxation of the left and right barrier constraints is enabled by the relaxed ZCBF introduced in section **V** of [71].

The deadlock resolution strategies are consistent for different types of deadlocks in that a clockwise motion will emerge for all agents involved in the deadlock. Therefore when multiple agents get into a deadlock, they will be perturbed to give way to the agent on the right side as if traffic rules are enforced.

Proposition 3.4.1. *Type 1 and Type 2 deadlocks are resolved with the Decentralized Deadlock Detection Resolution Algorithm 1 in the Appendix.*

Proof. For Type 1 Deadlock, $u_i = 0 \in \text{vertex}(\mathscr{P}_i)$. When the left barrier constraint is relaxed and the right barrier constraint is compressed, $u_i = 0 \notin \text{vertex}(\mathscr{P}_i')$ as shown in Fig. 3.7a. Therefore, the optimal control command to the perturbed QP-based controller is $\bar{\mathbf{u}}_i \neq 0$.

For Type 2 Deadlock, $u_i = 0 \in \text{edge}(\mathscr{P}_i)$. Due to Karush-Kuhn-Tucker (KKT) conditions, $\hat{\mathbf{u}}_i$ is perpendicular to the edge of $\mathscr{P}_i$, otherwise $u_i = 0$ is not the optimal solution



(a) Type 1 Deadlock, green($\mathscr{P}_i$)/blue($\mathscr{P}'_i$) polygons represent original/perturbed feasible control space.

(b) Type 2 Deadlock, the green polygon ($\mathscr{P}_i$) is the feasible control space.

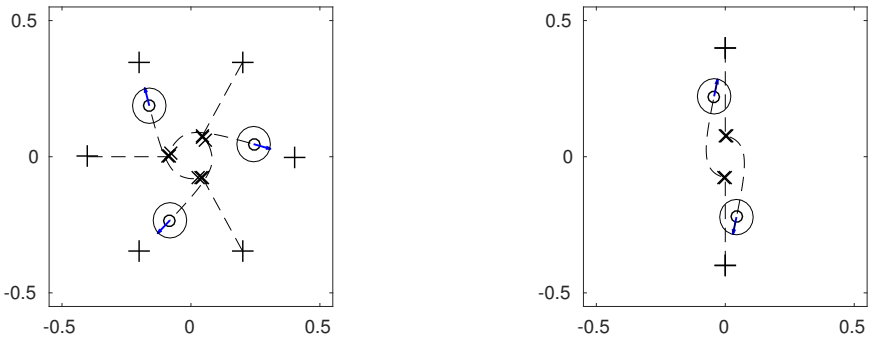
Figure 3.7: Deadlock resolution methods, where $\hat{\mathbf{u}}_i$, $\mathbf{u}_i = 0$ and $\bar{\mathbf{u}}_i$ are the nominal, original and adjusted control commands respectively.

for the QP. When $\delta^\perp$, a perturbation normal to $\hat{\mathbf{u}}_i$, is applied, $u_i = 0$ is no longer the optimal control command to the perturbed QP-based controller. Therefore, the actual control command $\bar{\mathbf{u}}_i \neq 0$ as shown in Fig. 3.7b.

Combining the two cases, the adjusted control command $\bar{\mathbf{u}}_i$ is non-zero without compromising the safety guarantee. Therefore, Type 1 and Type 2 Deadlocks are resolved. □

In Fig. 3.13, the Decentralized Deadlock Detection Resolution Algorithm 1 in the Appendix is validated against different deadlock scenarios. The algorithm successfully perturbed agents away from deadlock scenarios in a consistent way (clockwise rotation around each other emerges in both cases).

The consistent perturbation approach provides solutions to resolve all types of deadlocks except Type 3 deadlocks. In addition, livelock might still exist even if deadlock is resolved. This is because the safety barrier certificates in this dissertation is proposed to provide safety guarantee regardless of the purpose of the nominal controller. As the safety controller is not informed about what the nominal controller is ultimately trying to achieve, livelock becomes a somewhat diffuse concept. One possible idea could be to combine the safety certificates with navigation functions, such that the robots only move in the direc-



(a) Type 1 Deadlock (b) Type 2 Deadlock

Figure 3.8: Simulated deadlock resolution. The circles, arrows and dashed lines represent the current positions, velocities and trajectories of different agents respectively. The cross markers represent the places where the deadlock occurs and the deadlock resolution algorithm is active.

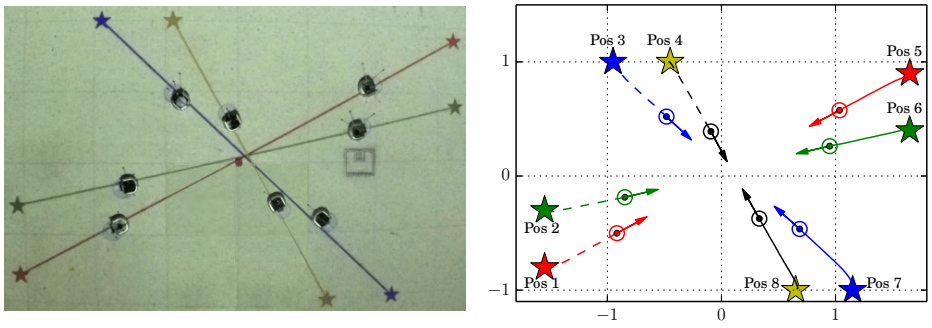
tions where the navigation function decreases.

3.5 Experimental Implementation

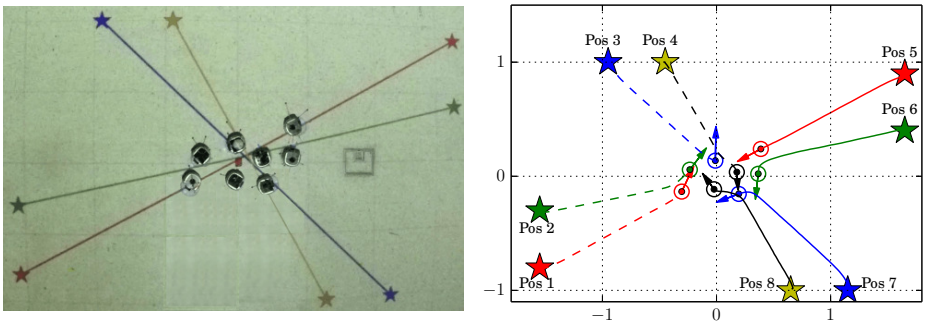
The decentralized safety barrier certificates works for both homogeneous and heterogeneous teams of robots. In this section, experimental results are presented to validate the barrier certificates on a multi-robot testbed.

3.5.1 Decentralized Barrier Certificates on a Homogeneous Team of Robots

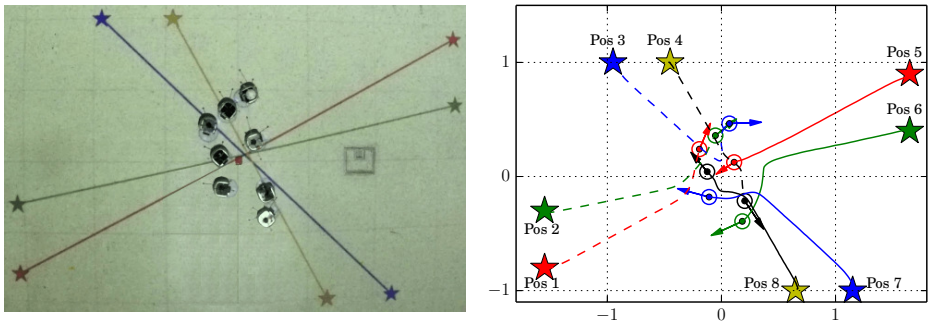
The decentralized safety barrier certificates were implemented on a multi-robot system consisting of multiple Khepera III robots. The higher level goal of the experiment was to make all robots in the multi-robot system swap positions with each other in a confined workspace, where collisions were very likely to occur. With the decentralized safety barrier certificates, all the robots successfully swapped their positions with the robots on the opposite side of the workspace without collisions as shown in Fig. 3.10.



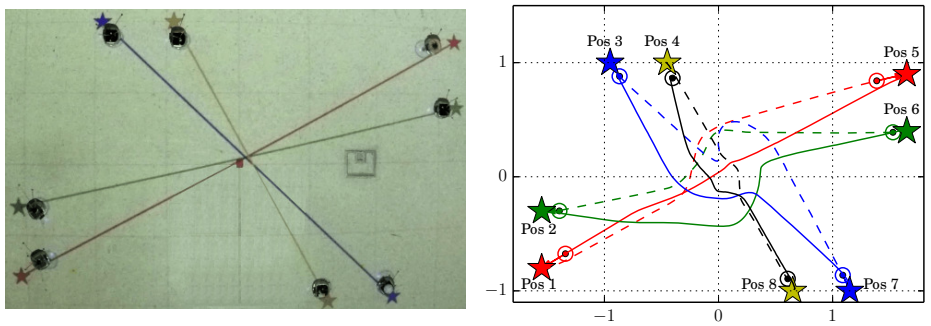
(a) Agents at 3.0s



(b) Agents at 6.0s



(c) Agents at 8.0s



(d) Agent at 15.0s

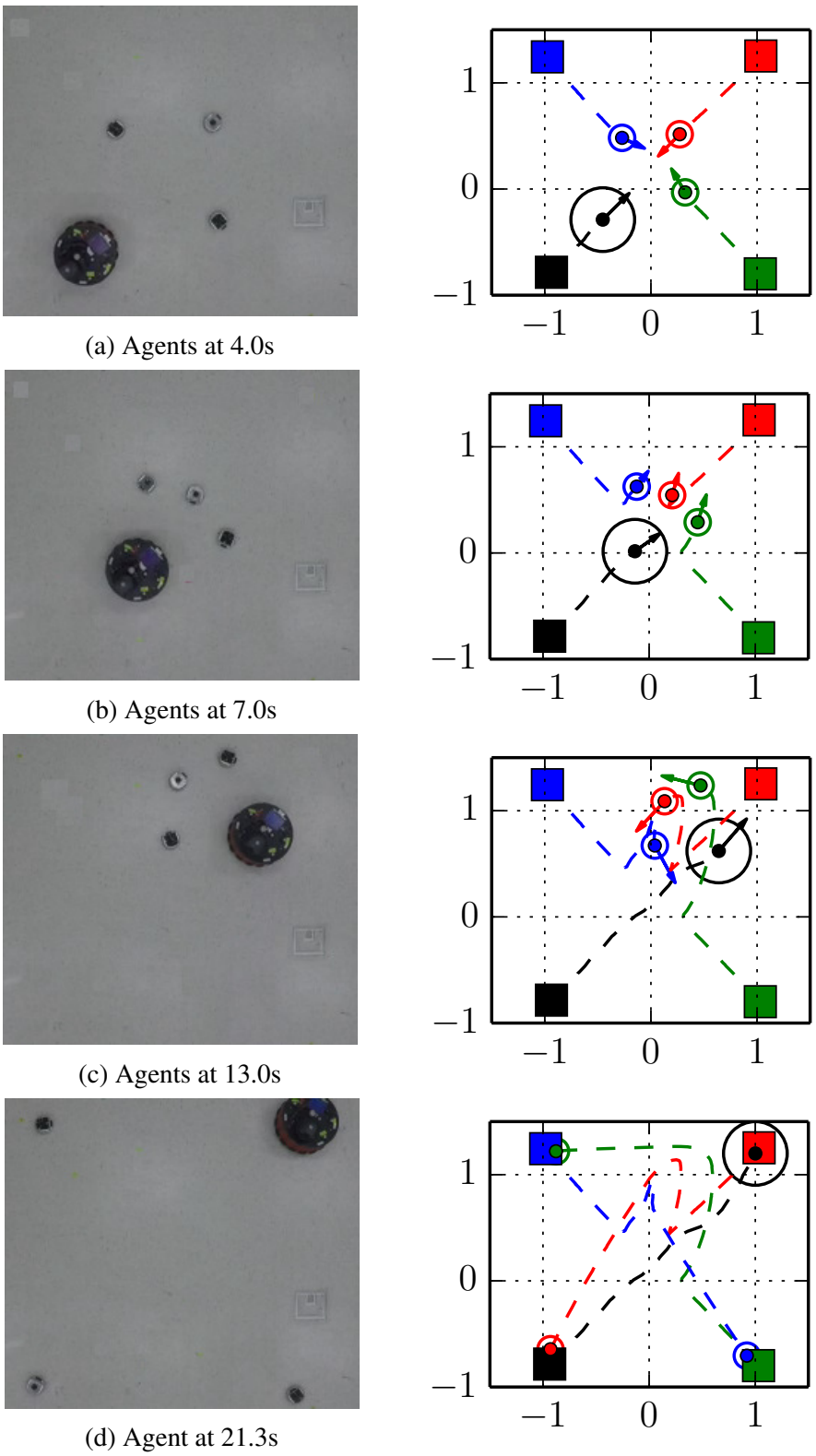
Figure 3.9: Experiment of eight Khepera robots swapping positions in a confined workspace. The pictures on the left are taken with an overhead camera. The stars and lines representing the target positions and pairs of swapped positions are projected onto the floor using a projector. The figures on the right illustrate the actual positions, velocities and trajectories of the robots. A video of the experiment can be found online [72].

3.5.2 Decentralized Barrier Certificates on a Heterogeneous Team of Robots

The heterogeneous safety barrier certificates were implemented on a heterogeneous robotic swarm with three Khepera III robots ($\alpha_K = 2.0\ m/s^2$) and one Magellan Pro robot ($\alpha_M = 0.5\ m/s^2$). The positions of robots are tracked by Optitrack motion capture system. Those two types of robots have distinct dimensions and dynamical capabilities. The diameters of Khepera III and Magellan Pro robots are 13 *cm* and 41 *cm*. The actual dynamical model of mobile robots used in this experiment is unicycle model, which is approximated with double integrated dynamics using Lyapunov based approach. The pre-designed controller is a goal-to-goal controller ($\hat{\mathbf{u}}_i = -k_1(\mathbf{p}_i - \mathbf{r}_i) - k_2\mathbf{v}_i$), which exchanges the positions of agents on the diagonal line of a rectangle, without considering collision avoidance. The heterogeneous safety barrier certificates were executed as a lower level safety program with no knowledge about overall goal of the higher level controller.

Fig. 3.10 shows a overhead view of the robots during the experiment and plots of corresponding experimental data. All four robots started heading straightly towards the opposite side of the rectangle (Fig. 3.10a). The safety barrier was inactive because the pre-designed coordination control command is considered safe. When robots moved closer, the safety barrier interfered because collision was about to happen. As illustrated in (Fig. 3.10b), three Khepera III robots turned around to avoid collision, while the Magellan robot kept pushing forward. This is because Magellan Pro robot has more momentum and can not brake fast enough to avoid collision. Those more agile Khepera III robots carried more responsibilities in collision avoidance when Magellan Pro robot reacted slowly. When the Magellan Pro robot almost reached its goal position and became slower in motion, other Khepera III robots got the chance to pursue their goals (Fig. 3.10c). It can be observed that the safety barrier directed robots away from collision and computes the command that is closest to pre-designed control command. After robots navigated away from the ”crowded” area, the pre-designed controller took over again. At last, all robots reached desired configuration, i.e. exchange position with robots on the opposite side (Fig. 3.10d).

Compared with the experiments performed in the homogeneous case, the heterogeneity in the robots' dynamical capabilities brought more challenges to guarantee the safety of the robotic swarm. Those challenges in the experiment have been successfully addressed by the heterogeneous safety barrier certificates. Meanwhile, the safety barrier certificates are implemented in a minimal invasive manner, which provides a faster way to develop and validate multi-agent algorithm without worrying about collision avoidance.



(a) Agents at 4.0s

(b) Agents at 7.0s

(c) Agents at 13.0s

(d) Agent at 21.3s

Figure 3.10: Test run of three Khepera robots (small circles) and one Magellan robot (large circle) with heterogeneous safety barrier certificates. The arrow, circle and dashed line represent current velocity, position and trajectory of robot agents. The square markers stand for initial and goal positions.

3.6 Applications in the Robotarium

The Robotarium is a remotely accessible open-access swarm robotic research platform [65]. It is developed to allow robotics researchers to implement swarm algorithms on the real robots without significant investments of manpower and resources. The prototype of the Robotarium is a table top equipped with multiple miniature sized GRITSBot [73] and wireless charging module as shown in Fig. 3.11. Now it has grown to room-sized experimental platform with remote users from around the world as illustrated in Fig. 3.12.

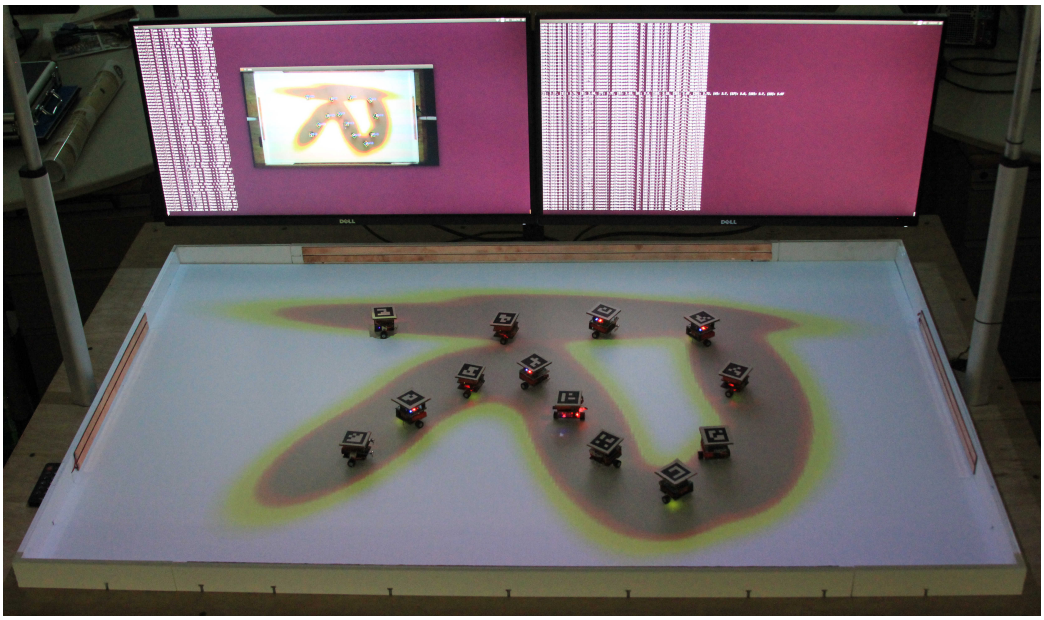


Figure 3.11: Table top version of the Robotarium

The Robotarium uses *Safety Barrier Certificates* to ensure provably collision-free behavior of all robots, which ensures the following three principles.

- All robots are provably safe in the sense that collisions are avoided.
- Users' commands are only modified when collisions are truly imminent.
- Collision avoidance is executed in real-time (in excess of 30 Hz update rate).

Safety barrier certificates are enforced through the use of control barrier functions, which are Lyapunov-like functions that can provably guarantee forward set invariance, i.e. if the

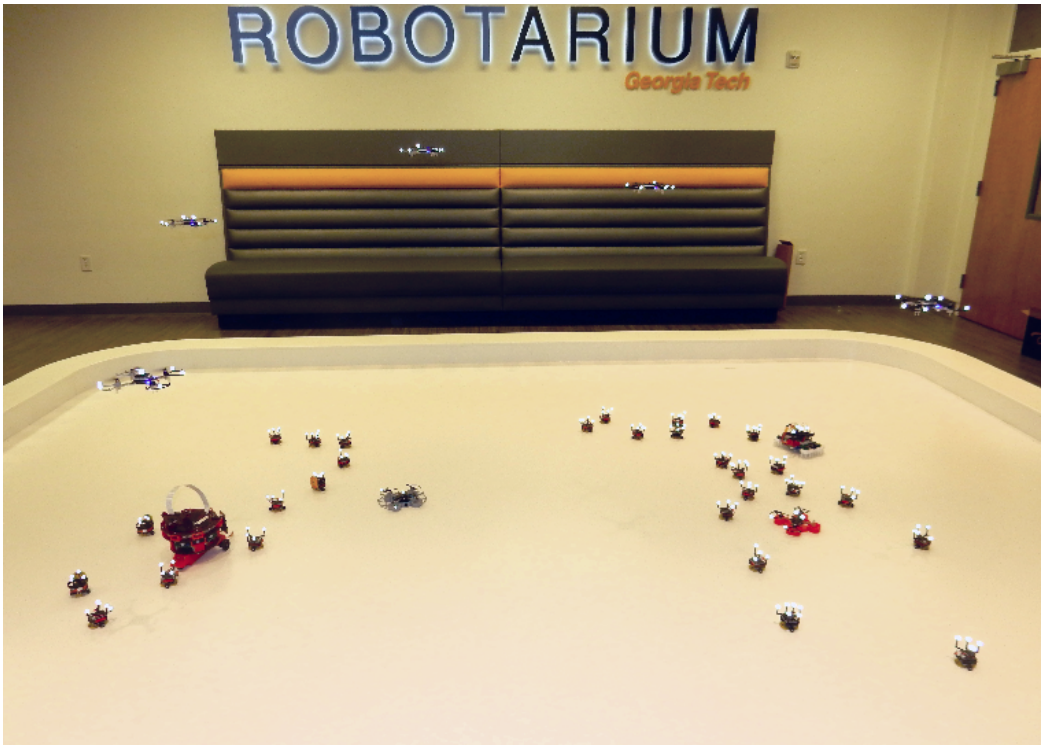


Figure 3.12: Current version of the Robotarium

system starts in the safe set, it stays in the safe set for all time. A specific class of maximally permissive control barrier functions was introduced in [16], whose construction provides the basis for the minimally invasive safety guarantees afforded by the Robotarium.

Consider a team of N mobile robots with the index set $\mathscr{M} = \{1, 2, ..., N\}$. Each robot i uses single integrator dynamics of the form $\dot{x}_i = u_i$, where $x_i \in \mathbb{R}^2$ is the planar position of robot i, and $u_i \in \mathbb{R}^2$ is its input velocity.[1] Additionally, robot i's velocity u_i is bounded by $\|u_i\| \leq \alpha, \forall i \in \mathscr{M}$. Let $x = [x_1^T, x_2^T, ..., x_N^T]^T$ and $u = [u_1^T, u_2^T, ..., u_N^T]^T$ denote the aggregate state and velocity input of the entire team of robots. To avoid inter-robot collisions, any two robots i and j need to maintain a minimum safety distance D_s between each other. This requirement is encoded into a pairwise safe set $\mathscr{C}_{ij}$, which is a super level set of a smooth

[1] Single integrator dynamics can be easily mapped to the GRITSBot's unicycle dynamics using a nonlinear inversion method. It is important to note that safety barrier certificates can be extended to more complex dynamical systems as well.

function $h_{ij}(x)$,

$$\mathscr{C}_{ij} = \{x_i \in \mathbb{R}^2 \mid h_{ij}(x) = \|x_i - x_j\|^2 - D_s^2 \geq 0\},\ \forall\, i \neq j. \tag{3.10}$$

The function $h_{ij}(x)$ is called a control barrier function, if the admissible control space

$$K_{ij}(x) = \left\{ u \in \mathbb{R}^{2N} \;\middle|\; \frac{\partial h_{ij}(x)}{\partial x} u \geq -\gamma h_{ij}(x) \right\}, \tag{3.11}$$

is non-empty for all $x_i \in \mathscr{C}_{ij}$. It was shown in [62] that if the control input u stays in $K_{ij}(x)$ for all time, then the safe set $\mathscr{C}_{ij}$ is forward invariant. In addition, the forward invariance property of $\mathscr{C}_{ij}$ is robust with respect to different perturbations on the system.

Combining (4.7) and (3.11) as well as the single integrator dynamics, the velocity input u needs to satisfy

$$-2(x_i - x_j)u_i + 2(x_i - x_j)u_j \leq \gamma h_{ij}(x),\ \forall\, i \neq j.$$

This inequality can be treated as a linear constraint on u when the state x is given, i.e., $A_{ij}u \leq b_{ij},\ \forall\, i \neq j$, where

$$\begin{aligned} A_{ij} &= [0, \ldots, \underbrace{-2(x_i - x_j)^T}_{\text{robot } i}, \ldots, \underbrace{2(x_i - x_j)^T}_{\text{robot } j}, \ldots, 0] \\ b_{ij} &= \gamma h_{ij}(x). \end{aligned}$$

Similar constraints must be established for the workspace boundary. The corresponding safety set of robot i with regards to the boundary is denoted by $\bar{\mathscr{C}}_i$, and the corresponding constraints by $\bar{A}_i u_i \leq \bar{b}_i,\ \forall\, i \in \mathscr{M}$.

Combining these constraints – *all* pairwise collisions and collisions with the workspace

boundaries – results in the safety set for the entire team as

$$\mathscr{C} = \prod_{i \in \mathscr{M}} \Big\{ \bigcap_{\substack{j \in \mathscr{M} \\ j \neq i}} \mathscr{C}_{ij} \bigcap \bar{\mathscr{C}}_i \Big\}.$$

The forward invariance of the safe set $\mathscr{C}$ is guaranteed by the safety barrier certificates, which are defined as

$$K(x) = \left\{ u \in \mathbb{R}^{2N} \;\middle|\; A_{ij} u \leq b_{ij}, \bar{A}_i u_i \leq \bar{b}_i, \; \forall \, i \neq j \right\}. \tag{3.12}$$

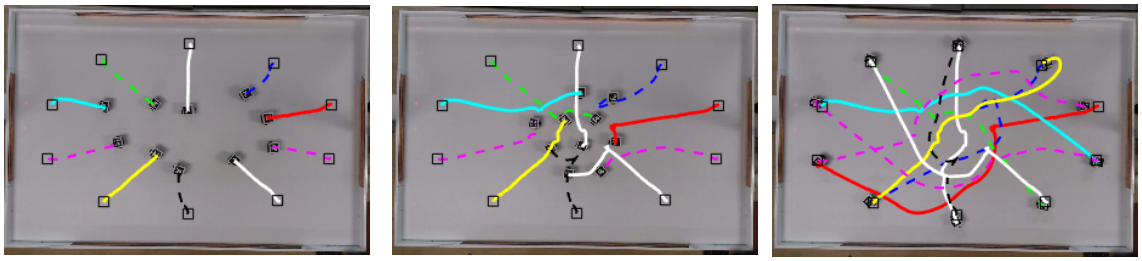
These safety barrier certificates define a convex polytope $K(x)$ in which safe control commands must stay. By constraining users' control commands to within $K(x)$, the Robotarium is guaranteed to operate in a provably collision-free manner.

The *minimally invasive* nature of barrier certificate-enabled collision avoidance stems from the fact that the deviation between the user-specified control signal and the actual, safe, executed signal is minimized, subject to the safety constraints through a Quadratic Program (QP)-based controller

$$\begin{aligned} u^* = \underset{u \in \mathbb{R}^{2n}}{\operatorname{argmin}} \quad & J(u) = \sum_{i=1}^{N} \|u_i - \hat{u}_i\|^2 & \\ \text{s.t.} \quad & A_{ij} u \leq b_{ij}, & \forall \, i \neq j, \\ & \bar{A}_i u_i \leq \bar{b}_i, & \forall \, i \in \mathscr{M}, \\ & \|u_i\|_\infty \leq \alpha, & \forall \, i \in \mathscr{M}, \end{aligned} \tag{3.13}$$

where $\hat{u}$ is the user's control command, u^* is the actual control command, and α is the bound for the control input. Note that in the absence of impending collisions (i.e. when the safety barrier certificates in (3.12) are satisfied), the user's code is executed faithfully. When violations occur, a closest possible (in a least-squares sense) safe control command is computed and executed instead. An experiment showing ten GRITSBots swapping po-

sitions with active safety barrier certificates is shown in Fig. 3.13, while the corresponding video is referenced in [74].



(a) Time 2.67s (b) Time 9.50s (c) Time 19.83s

Figure 3.13: Ten GRITSBots swap positions with active safety barrier certificates. The robots' trajectories are shown together with square markers representing their initial positions.

Scalability of Safety Barrier Certificates

Safety barrier certificates are computed in a centralized fashion on the Robotarium's back end server and therefore scalability is a concern. The following analysis shows the feasibility of barrier certificates for large swarms. As the size of the swarm increases, the number of decision variables (u) in the QP-based controller increases linearly, while the number of pairwise safety constraints grows quadratically. However, a more computationally efficient implementation similar to [66] is possible, where agent i only considers its neighbors for collision avoidance and the certificates computation can be distributed to individual agents. More specifically, the robot's finite physical dimensions limit the maximum robot density. For example, for a minimum safety distance of $D_s = 8cm$ and a neighborhood radius of $20cm$, any given neighborhood can contain at most 26 other robots, which limits the size of each individual robots QP problem to 2 decision variables and at most 26 linear constraints. More specifically, for a minimum safety distance between GRITSBots of $D_s = 8cm$ and a GRITSBot's neighborhood radius of, for example, $20cm$, there can be at most 26 other robots in the most densely packed scenario. By distributing the computation of barrier certificates, each agent only needs to solve a QP with 2 decision variables and at

Swarm Size N	**Centralized Certificates** T_c **(ms)**	**Decentralized Certificates** T_d/N **(ms)**
10	5.6	3.2
40	11.6	3.5
100	78.0	5.4

Table 3.3: Computation time of barrier certificates for each iteration.

most 26 linear constraints. Therefore, the certificates can be computed in real-time on the Robotarium, i.e., an update frequency in excess of 30 Hz.

The computation time of safety barrier certificates for each iteration is shown in Table 3.3. Note that the decentralized barrier certificates here are simulated on a single central computer.[2] Thus, the total computation time T_d is divided by N to characterize the decentralized and fully parallel implementation. As Table 3.3 shows, decentralized safety barrier certificates can handle 100 GRITSBots with an update frequency of 185Hz and therefore scale to large numbers of robots without compromising update rates.

With the embedded safety mechanism provided with the barrier certificates, the Robotarium supports remote-access multi-robot research without posing risks to physical equipment. The barrier certificates are adopted by multiple research groups for experiments conducted on the Robotarium [75, 23, 76]. The transition from theoretical development in multi-robot systems to experimental implementations is accelerated significantly with the help of the Robotarium.

[2]Barrier certificates were computed on an Intel I7 4790 3.6 GHz with 16 GB of memory.

CHAPTER 4

BARRIER CERTIFICATES FOR TEAMS OF QUADROTORS

Teams of quadrotors are widely used aerial robotic platforms. Due to the under-actuated and intrinsically unstable nature of unmanned aerial vehicles, it is often challenging to generate safe trajectories for arbitrary tasks, such as aerial delivery, convoy protection, and cooperative environment surveillance, see [77, 78, 79] and the references therein. The objective of this chapter is to develop safety certificates for efficient or even aggressive maneuvers in teams of quadrotors based on their nonlinear dynamics [80].

4.1 Quadrotor Dynamics and Differential Flatness

The quadrotor is a well-modelled dynamical system with forces and torques generated by four propellers and gravity. $Z-Y-X$ Euler angles conventions are adopted to define the roll (ϕ), pitch (θ), and yaw (ψ) angles as illustrated in Fig. 4.1.

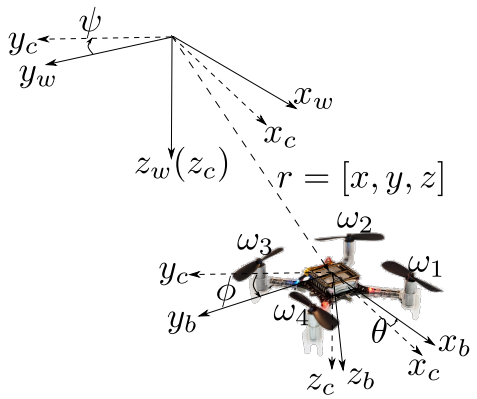


Figure 4.1: Quadrotor coordinate frames. The subscripts w denotes the world frame F_w, b for the quadrotor body frame F_b, and c for an intermediate frame F_c after yaw angle rotation. ω_1 to ω_4 are the angular velocities of the four propellers. The palm-sized quadrotor illustrated is a Crazyflie 2.0 [81] used in the experiment section.

The quadrotor dynamics has been shown to be differentially flat in [6, 82], i.e., the states and inputs of the system can be written in terms of algebraic functions of appropriately

chosen flat outputs and their derivatives. As shown in [82], the flat output for quadrotor can be chosen as $\sigma = [x, y, z, \psi]^T$. The full state $\xi = [x, y, z, v_x, v_y, v_z, \psi, \theta, \phi, p, q, r]^T$ and input $\mu = [f_z, \tau_x, \tau_y, \tau_z]^T$ can be represented algebraically using the following functions

$$\xi = \beta(\sigma, \dot{\sigma}, \ddot{\sigma}, \dddot{\sigma}), \qquad \mu = \gamma(\sigma, \dot{\sigma}, \ddot{\sigma}, \dddot{\sigma}, \ddddot{\sigma}),$$

where we refer to [82] for a derivation and formula of the endogenous transformation (β, γ).

The flight trajectory can be planned in a greatly simplified flat output space with the differential flatness property. For simplicity of planning, the yaw angle is set to $\psi(t) = 0$. Note that the yaw angle control is useful when an onboard camera is present, in which case the differential flatness based planning method still applies. Let $r = \sigma_{1:3} = [x, y, z]^T \in \mathbb{R}^3$, a virtual control input $v \in \mathbb{R}^3$ can be created for the forth order integrator dynamics $\ddddot{r} = v$, which can be equivalently written as state space form

$$\dot{q} = \underbrace{\overbrace{\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}}^{F \in \mathbb{R}^{4\times 4}} \otimes I_{3\times 3} \cdot q}_{f(q)} + \underbrace{\overbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}}^{G \in \mathbb{R}^4} \otimes I_{3\times 3} \cdot v}_{g(q)}, \tag{4.1}$$

where $q = [r^T, \dot{r}^T, \ddot{r}^T, \dddot{r}^T]^T \in \mathbb{R}^{12}$, $\otimes$ is Kronecker product. Note that since collision avoidance requires simultaneous response of three degrees of freedom, the trajectory planning problem here can not be simplified by decoupling three independent degrees of freedom, as was done in [6, 5].

With the differential flatness property, trajectory planning for quadrotors can be simplified as spline interpolations. "keyframes" representing the desired waypoints can be placed in the 3D space using existing higher level planning algorithms, e.g., [83, 84]. An optimal

(minimal-snap) trajectory can then be generated with smooth splines [6].

The optimal flight trajectory is generated segment by segment. Let the starting and ending "keyframe" of a segment of trajectory be denoted as $[\bar{r}_0, \dot{\bar{r}}_0, \ddot{\bar{r}}_0, \dddot{\bar{r}}_0]^T \in \mathbb{R}^{4\times 3}$ and $[\bar{r}_1, \dot{\bar{r}}_1, \ddot{\bar{r}}_1, \dddot{\bar{r}}_1]^T \in \mathbb{R}^{4\times 3}$. The actual flight trajectory is formulated as a 3D spline.

$$r(t) = \begin{bmatrix} \sum_{i=0}^{N} \alpha_{1i} B_i(t) \\ \sum_{i=0}^{N} \alpha_{2i} B_i(t) \\ \sum_{i=0}^{N} \alpha_{3i} B_i(t) \end{bmatrix}, \tag{4.2}$$

where α_{ij} and $B_i(t)$ are the coefficients and polynomial basis. A convex optimization problem can be formulated as

$$\begin{aligned} \min_{\alpha_{ij}} \quad & \int_{t_0}^{t_1} \left\| \frac{\mathrm{d}^4 r(t)}{\mathrm{d}t^4} \right\|^2 \mathrm{d}t \\ \text{s.t.} \quad & \left. \frac{\mathrm{d}^k r(t)}{\mathrm{d}t^k} \right|_{t=t_0} = \frac{\mathrm{d}^k \bar{r}_0}{\mathrm{d}t^k},\ k = 0,1,2,3 \\ & \left. \frac{\mathrm{d}^k r(t)}{\mathrm{d}t^k} \right|_{t=t_1} = \frac{\mathrm{d}^k \bar{r}_1}{\mathrm{d}t^k},\ k = 0,1,2,3. \end{aligned}$$

An example of the flight trajectory generation method is provided in Fig. 4.2. As the "keyframes" are placed in the space, smooth splines can be generated to execute these desired maneuvers.

In addition to planning in the free space, spline interpolations can deal with safety corridor constraints. For example, let the safety corridor constraints between "keyframes" be defined as

$$d(t) = \|(r(t) - \bar{r}_0) - ((r(t) - \bar{r}_0) \cdot n)n\|_\infty \leq \delta,$$

where $n = \frac{\bar{r}_1 - \bar{r}_0}{\|\bar{r}_1 - \bar{r}_0\|}$. When a safety corridor is added between two "keyframes", we can sample multiple points along the corridor and add them as linear constraints. To accommodate

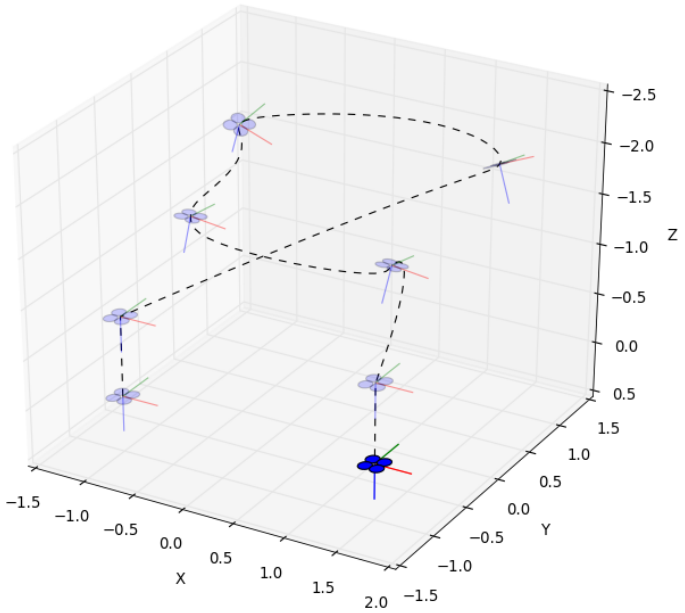


Figure 4.2: Flight trajectory generated with splines

these additional safety constraints, the order of the polynomial spline is increased.

$$
\begin{aligned}
\min_{\alpha_{ij}} \quad & \int_{t_0}^{t_1} \left\| \frac{\mathrm{d}^4 r(t)}{\mathrm{d}t^4} \right\|^2 \mathrm{d}t \\
\text{s.t.} \quad & \left. \frac{\mathrm{d}^k r(t)}{\mathrm{d}t^k} \right|_{t=t_0} = \frac{\mathrm{d}^k \bar{r}_0}{\mathrm{d}t^k}, \; k = 0,1,2,3 \\
& \left. \frac{\mathrm{d}^k r(t)}{\mathrm{d}t^k} \right|_{t=t_1} = \frac{\mathrm{d}^k \bar{r}_1}{\mathrm{d}t^k}, \; k = 0,1,2,3 \\
& d(t_m) \leq \delta, \; t_m = t_0 + \frac{m}{N}(t_1 - t_0), m = 1,2,...,N-1
\end{aligned}
$$

Considering the same planning problem with Fig. 4.2, a safety corridor is added between two "keyframes". The resulting flight trajectory is shown in Fig. 4.3. Compared with Fig. 4.2, the flight trajectory is constrained within the safety corridor.

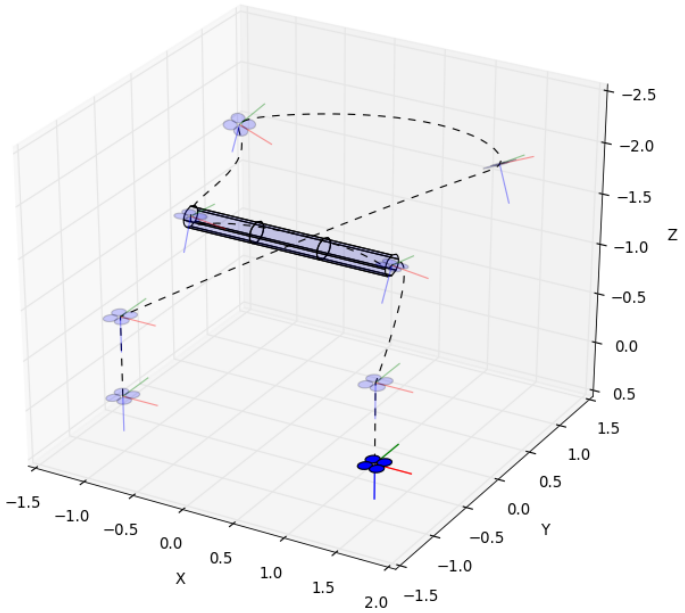


Figure 4.3: Flight trajectory generated with splines and safety corridor constraints, where the meshed tube is the safety corridor.

4.2 Exponential Control Barrier Functions

With the simplified forth-order integrator model for quadrotors, Control Barrier Functions (CBF) can be used to ensure collision-free flight maneuvers. Let the safe set be defined as

$$\mathscr{C}_0 = \{q \in \mathbb{R}^{12} \mid h(q) \geq 0\}, \tag{4.3}$$

where $h : \mathbb{R}^{12} \to \mathbb{R}$ is a smooth function.

Because $h(q)$ only contains the position variable r, we denote $y_0(r) = h(q)$. The relative degree of $y_0(r)$ is 4, which means that $y_0^{(4)}(r) = L_f^4 h(q) + L_g L_f^3 h(q) v$. With the high relative degree of $y(r)$, CBFs in [16, 62] can not be directly applied. A variation of the CBF, i.e., "Exponential Control Barrier Function" (ECBF) [18], can however ensure the forward invariance of $\mathscr{C}_0$.

Definition 3.2.2: Given the dynamical system (4.1) and a set $\mathscr{C}_0$ defined in (4.3), the smooth function $h : \mathscr{C}_0 \to \mathbb{R}$ with relative degree of 4 is an Exponential Control Barrier

Function (ECBF) if there exists a vector $K \in \mathbb{R}^{1\times 4}$ such that $\forall x \in \mathscr{C}_0$,

$$\sup_{u\in U}[L_f^4 h(q) + L_g L_f^3 h(q) v + K\eta] \geq 0, \tag{4.4}$$

and $h(q(t)) \geq Ce^{F-GK}\eta(q_0) \geq 0$ when $h(q_0) \geq 0$, where $\eta = [h(q), L_f h(q), L_f^2 h(q), L_f^3 h(q)]^T$, $C = [1,0,0,0]$. K can be obtained by placing the poles of the closed-loop matrix $(F-GK)$ at $p = -[p_1, p_2, ..., p_4]^T$, where $p_i > 0$ for $i = 1,2,3,4$. With these pole locations, a family of outputs $y_i, i = 1,2,3,4$ can be defined as

$$y_i = (\frac{\mathrm{d}}{\mathrm{d}t} + p_1) \circ (\frac{\mathrm{d}}{\mathrm{d}t} + p_2) \circ ... \circ (\frac{\mathrm{d}}{\mathrm{d}t} + p_i) \circ h(q),$$

with $y_0 = h(q)$, and the associated family of super level sets

$$\mathscr{C}_i = \{q \in \mathbb{R}^{12} \mid y_i(q) \geq 0\}. \tag{4.5}$$

Theorem 4.2.1. *Given a safe set $\mathscr{C}_0$ in (4.3) and associated ECBF $h(q) : \mathscr{C}_0 \to \mathbb{R}$, with initially $q_0 \in \mathscr{C}_i, i = 0,1,2,3$ for system (4.1), any Lipschitz continuous controller $v(q) \in K_v(q)$ renders $\mathscr{C}_0$ forward invariant, where*

$$K_v(q) = \{v \in V \mid L_f^4 h(q) + L_g L_f^3 h(q) v + K_v \eta \geq 0\},$$

and $\eta = [h(q), L_f h(q), L_f^2 h(q), L_f^3 h(q)]^T$.

We refer to [18] for the proof of general cases of this theorem.

4.3 Safety Barrier Certificates for Teams of Quadrotors

With the differential flatness property of quadrotor dynamics and ECBF for a single quadrotor, we can construct *Safety Barrier Certificates* for teams of quadrotors.

Safety Region Modelled With Super-ellipsoids

Consider a team of m quadrotors ($\mathscr{M} = \{1,2,\ldots,m\}$), each quadrotor is modelled as

$$\dddot{r}_i = v_i,\ i \in \mathscr{M} \tag{4.6}$$

where $r_i = [x_i, y_i, z_i]^T$ is the center of mass of quadrotor i. The full state of quadrotor i is represented by $q_i = [r_i^T, \dot{r}_i^T, \ddot{r}_i^T, \dddot{r}_i^T]^T \in \mathbb{R}^{12}$. Let $r = [r_1^T, r_2^T, ..., r_m^T]^T \in \mathbb{R}^{3m}$ and $v = [v_1^T, v_2^T, ..., v_m^T]^T \in \mathbb{R}^{3m}$ denote the aggregate position and virtual control. Each quadrotor is encapsulated with a "rectangle shape" super-ellipsoid[1],

$$\begin{aligned} \mathscr{C}_{ij} &= \{(q_i, q_j) \mid h_{ij}(q_i, q_j) \geq 0\}, \\ h_{ij}(q_i, q_j) &= (x_i - x_j)^4 + (y_i - y_j)^4 + (\frac{z_i - z_j}{c})^4 - D_s^4, \end{aligned} \tag{4.7}$$

where D_s is the safety distance, c is the scaling factor along the Z axis caused by air flow disturbance.

Since the pairwise safe set $\mathscr{C}_{ij}$ is defined in terms of position variables r_i, r_j, the ECBF candidate $h_{ij}(q_i, q_j)$ has a relative degree of 4. To ensure the forward invariance of $\mathscr{C}_{ij}$, virtual controls of quadrotor i and j need to satisfy

$$\ddddot{h}_{ij} + K \cdot [h_{ij}, \dot{h}_{ij}, \ddot{h}_{ij}, \dddot{h}_{ij}]^T \geq 0, \tag{4.8}$$

where $\ddddot{h}_{ij}$ is affine in v_i, v_j. Thus, it can be rearranged into a linear constraint on the virtual control when q_i, q_j are given, i.e., $A_{ij}(q_i, q_j) \cdot v \leq b_{ij}(q_i, q_j)$. The *Safety Barrier Certificates* are then formed by assembling all the pairwise safety barrier constraints

$$K_{\text{safe}} = \{v \in \mathbb{R}^{3m} \mid A_{ij}(q_i, q_j) \cdot v \leq b_{ij}(q_i, q_j), \forall i < j, i, j \in \mathscr{M}\}. \tag{4.9}$$

[1] A super-ellipsoid is a solid geometry generally defined with the implicit function $[(\frac{x}{a})^r + (\frac{y}{b})^r]^{\frac{n}{r}} + (\frac{z}{c})^n \leq 1$ with $r, n \in \mathbb{R}^+$[85]. $r = n = 4$ is selected to approximate a "rectangle shape" here.

As long as the virtual control v satisfies the *Safety Barrier Certificates* K_{safe} and corresponding initial conditions, the team is guaranteed to be safe by *Theorem 4.2.1*.

Modifying the Nomimal Trajectory with Safety Barrier Certificates

It is often difficult to generate provably collision-free trajectories when planning the nominal trajectory for teams of quadrotors. Instead, we can first plan the flight trajectory without considering collisions, and then modify it using *Safety Barrier Certificates* in a minimally invasive way to avoid collisions. Here we consider the case when a nominal trajectory $\hat{r}(t) = [\hat{r}_1^T(t), \hat{r}_2^T(t), ..., \hat{r}_m^T(t)] \in C^4$ is provided. This smooth reference trajectory $\hat{r}_i(t)$ is then tracked by a simulated integrator model using a pole placement controller with a simulated time step of $0.02s$ (simulates the 50Hz flight controller),

$$\hat{v}_i = \ddddot{\hat{r}}_i - K \cdot [\hat{r}_i \; \dot{\hat{r}}_i \; \ddot{\hat{r}}_i \; \dddot{\hat{r}}_i]^T, \tag{4.10}$$

where K is picked to be the same as used for ECBFs in (4.8) to trade-off tracking performance and safety enforcement.

Similar to Section 3.2, a QP is used to minimize the difference between the actual and nominal control,

$$\begin{aligned} v^* = \underset{v}{\text{argmin}} \quad & J(v) = \sum_{i=1}^{N} \|v_i - \hat{v}_i\|^2 \\ \text{s.t.} \quad & A_{ij}(q_i, q_j)v \leq b_{ij}(q_i, q_j), \qquad \forall i < j, \\ & \|v_i\|_\infty \leq \alpha_i, \qquad \forall i \in \mathscr{M}, \end{aligned} \tag{4.11}$$

It can be observed that the actual controller v_i will be the same as $\hat{v}_i$, if it is safe. The controller will only be rectified if it violates the Certificates, i.e., if it leads to collisions. The dynamics of the simulated forth-order integrator system is integrated forward using forward Euler method. According to [67], the controller v_i generated by the QP (6.13) will be Lipschitz continuous as well. Thus, the rectified collision-free trajectory $r(t)$ will still

be four times differentiable. Differential flatness property of quadrotors can still be used to execute the rectified collision-free trajectory $r(t)$.

4.4 Feasibility of the Certificates

The safety barrier certificates consist of multiple inequality constraints. The following result provides theoretical guarantees for feasibility.

Theorem 4.4.1. *Given a team of quadrotors indexed by $\mathscr{M}$ with dynamics given in (4.6), the aggregate admissible safe control space K_{safe} in (4.9) allowed by Safety Barrier Certificates is guaranteed to be non-empty.*

Proof. See Appendix. □

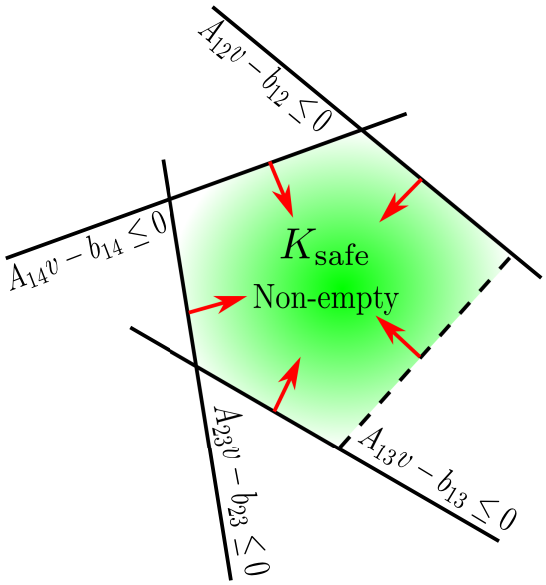


Figure 4.4: Visualization of K_{safe} as the intersection of multiple half spaces

The idea of control sharing barrier function used in the proof is similar to control-sharing and merging control Lyapunov functions introduced in [35].

4.4.1 Virtual Vehicle Parameterization

Collision avoidance maneuvers of quadrotors might sometimes lead to significant deviations from reference trajectories. After the collision hazard disappears, excessive control effort might be required for the quadrotors to return to the reference point along the nominal trajectory. To address this issue, a virtual vehicle parameterization method similar to [86] is developed.

The basic idea of virtual vehicle paramterization is to slow down or speed up the virtual vehicle (reference point $\hat{r}(t)$ on the nominal trajectory) as the tracking error $e_r = \|r - \hat{r}\|$ increases or decreases. In this particular application, we use the following virtual time variable to parameterize the reference point on the nominal trajectory

$$\dot{s} = \mathrm{e}^{-k_s \|e_r\|^2}, \tag{4.12}$$

where k_s is the virtual parameterization gain. Instead of $\hat{r}(t)$, $\hat{r}(s(t))$ is fed into the *Safety Barrier Certificates* rectifier shown in Fig. 4.7. Intuitively, the virtual vehicle will slow down ($\dot{s} < 1$) when the tracking error is large; it will travel exactly at the desire speed ($\dot{s} = 1$) when the tracking error is zero. This parameterization mechanism is intended to reduce the amount of control effort when the quadrotor has to deviate away from the virtual vehicle to avoid collisions.

To demonstrate the effectiveness of virtual vehicle parameterization, a simulation of two quadrotors flying pass each other is presented. The trajectories of two quadrotors are illustrated in Fig. 4.5.

In this example, the collision avoidance manuever requires a maximum pitch angle of 70° and a maximum thrust of 2.8 times hovering thrust without parameterization ($k_s = 0$) as shown in Fig. 4.6. In contrast, a maximum pitch angle of 25° and a maximum thrust of 1.2 times hovering thrust are needed with parameterization ($k_s = 100$). In both cases, the desired task is accomplished within 6s.

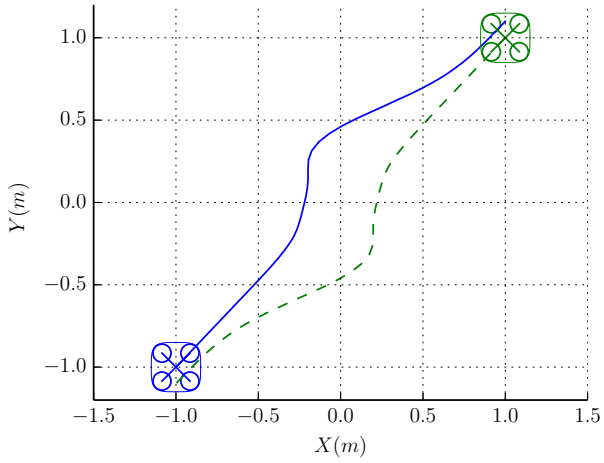


Figure 4.5: Trajectories of two quadrotors flying pass each other plotted in X-Y plane. Control efforts for performing this task are illustrated in Fig. 4.6.

As proved in section 4.4, the QP in (4.11) will always generate a feasible solution. However, the required control effort to avoid collisions might be excessive. In this circumstance, virtual vehicle parameterization method can be used to reduces instantaneous control efforts. By increasing the virtual parameterization gain k_s significantly, the quadrotors will be granted considerably more time to perform collision avoidance manuevers. Thus, the parameterization mechanism will generate a feasible trajectory that satisfies given actuator constraints.

4.4.2 Overview of Safe Trajectory Generation Strategy

An overview of the safe trajectory generation strategy is summarized in Fig. 4.7. A smooth reference trajectory $\hat{r}(t) \in C^4$ is first fed into the safety barrier certificates rectifier, where the QP controller (6.13) is used to enforce collision-free flight maneuvers. The rectified smooth safe trajectory $r(t) \in C^4$ is then transformed into quadrotor states and controls using the differential flatness property. The full states and controls are checked to ensure that actuator limits are not violated. Otherwise, the reference trajectory is parameterized $\hat{r}(s(t)) \in C^4$ and fed into the safety barrier certificates rectifier again. This process can be repeated until the virtual vehicle parameterization strategy yields appropriate flight trajec-

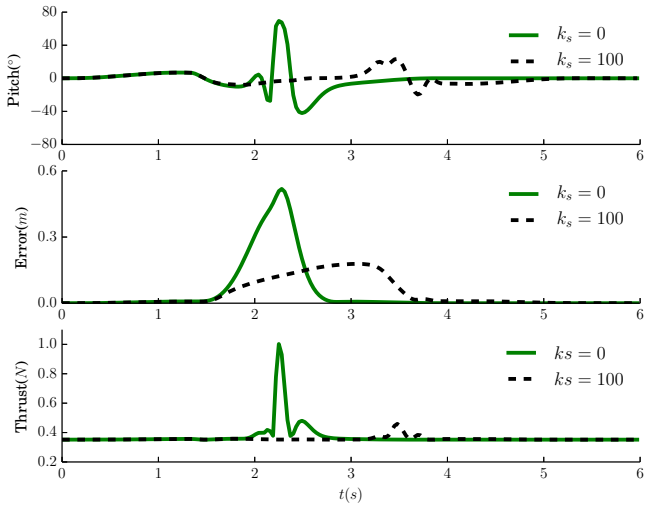


Figure 4.6: Comparisons of control efforts for the quadrotor using ($k_s = 100$) or without using ($k_s = 0$) virtual vehicle parameterization.

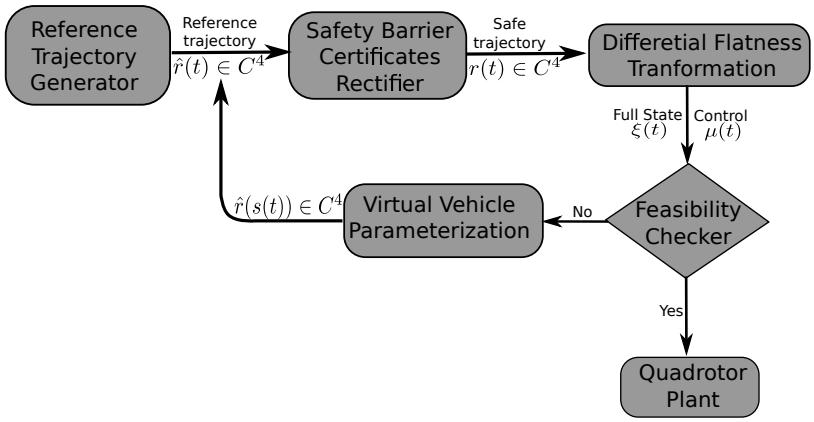


Figure 4.7: Flowchart of safe trajectory generation strategy.

tory that respects both safety and actuator constraints. In the end, the generated feasible safe trajectory is sent to execute on the team of quadrotors.

4.5 Experimental Implementations

The *Safety Barrier Certificates* are implemented on a team of five palm-sized quadrotors (Crazyflie 2.0). All communication channels between different devices and control programs are coordinated by a ROS server. The real-time positions and Euler angles of quadrotors are tracked by the Optitrack motion capture system with an update rate of 50Hz. The

50Hz quadrotor motion controller is developed based on the ROS driver for Crazyflie 2.0 built by ACTLab at USC [87]. To ensure stable trajectory tracking behavior, Euler angles and Euler angle rates generated with the differential flatness property are sent to quadrotors as control commands.

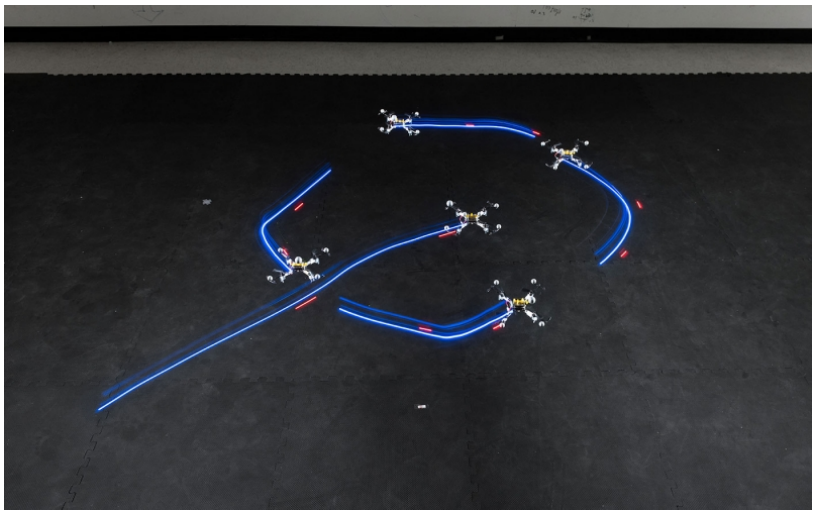


Figure 4.8: Long exposure photo of the experiment. The blue lights illustrate trajectories of the quadrotors. The video of this experiment is available online [88].

The overall quadrotor control diagram is shown in Fig. 4.9.

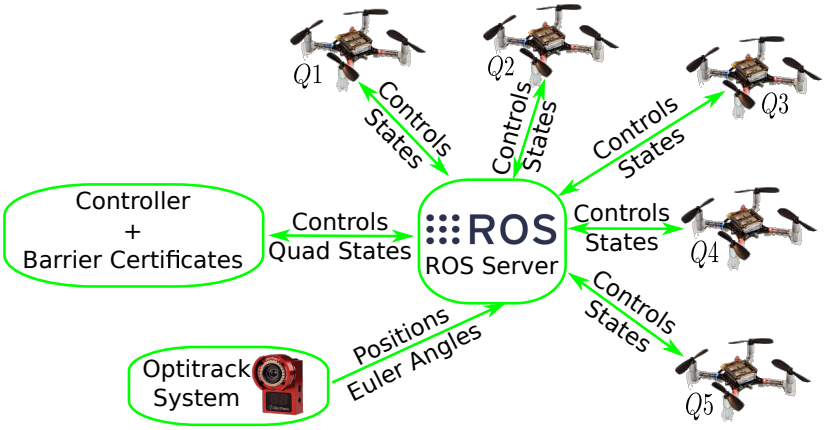


Figure 4.9: Quadrotor control system diagram

4.5.1 Showcase: Flying Through a Static Formation

In the first experiment, one of the quadrotor ($Q5$) is commanded to fly through a static formation consisting of four quadrotors ($Q1-Q4$) as shown in Fig. 4.10.

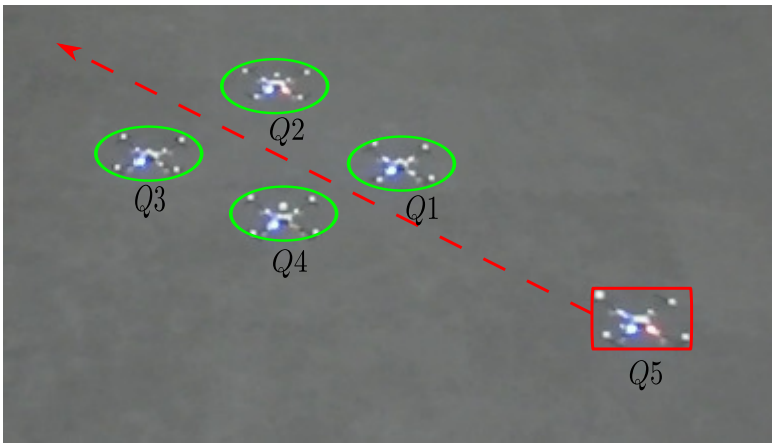


Figure 4.10: Snapshot from a experiment of quad $Q5$ flying through a static formation consisting of four quads $Q1-Q4$. The video of this experiment is available online [88].

Quadrotors ($Q1-Q4$) are designed to hover at four places with reference trajectories given as

$$\hat{r}_1(t) = \begin{bmatrix} 0.25 \\ 0 \\ -0.8 \end{bmatrix}, \quad \hat{r}_2(t) = \begin{bmatrix} 0 \\ 0.25 \\ -0.8 \end{bmatrix},$$

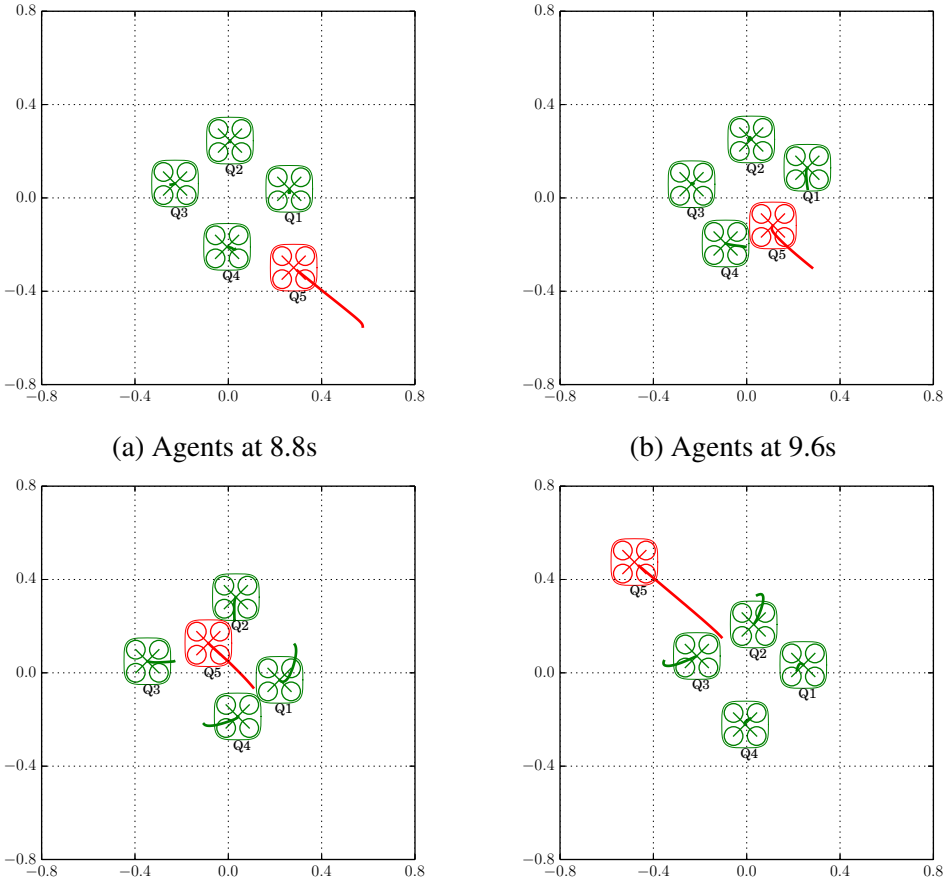
$$\hat{r}_3(t) = \begin{bmatrix} -0.25 \\ 0 \\ -0.8 \end{bmatrix}, \hat{r}_4(t) = \begin{bmatrix} 0 \\ -0.25 \\ -0.8 \end{bmatrix}.$$

Another quadrotor ($Q5$) is designed to go from $p_0 = [0.6, -0.6, -0.8]^T$ to $p_1 = [-0.6, 0.6, -0.8]^T$. The nominal trajectory can be generated as

$$\hat{r}_5(t) = \text{BezierInterp}(p_0, p_1),$$

where the BezierInterp function stands for the Bezier curve interpolation algorithm between two waypoints.

The safety distance between quadrotors is specified as $D_s = 25cm$ to account for the tracking frames and controller tracking errors. Intuitively, quadrotors will collide with each other if nominal trajectories are directly executed. During the experiment, the *Safety Barrier Certificates* are applied to modify the nominal trajectories in a minimally invasive way to avoid collisions. As demonstrated in Fig. 4.11, $Q5$ successfully navigated through the static formation of four quadrotors within 3.4s.



(a) Agents at 8.8s

(b) Agents at 9.6s

(c) Agents at 11.0s

(d) Agents at 12.2s

Figure 4.11: Experimental data of the team of quadrotors plotted in the X-Y plane. The tail of each quadrotor illustrates its trajectory in the past 0.8s.

4.5.2 Showcase: Flying Through a Spinning Formation

Similar to the previous experiment, the quadrotor $Q5$ is commanded to fly through a spinning formation as illustrated in Fig. 4.12.

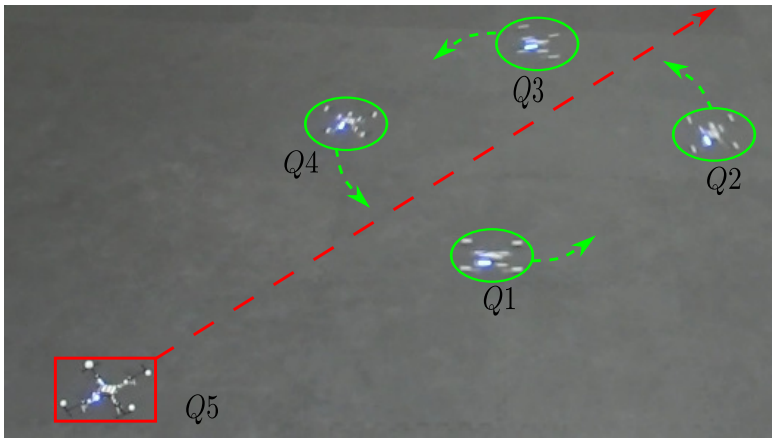


Figure 4.12: Snapshot from a experiment of quad $Q5$ flying through a spinning formation consisting of four quads $Q1-Q4$. The video of this experiment is available online [88].

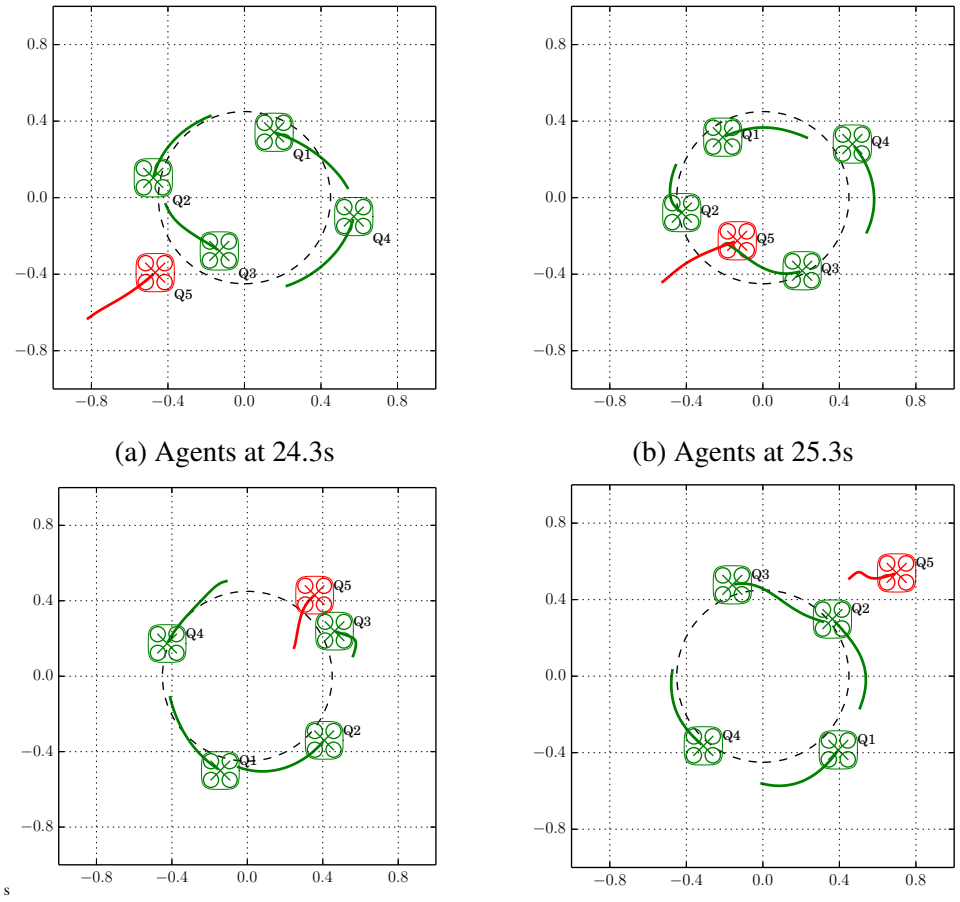
The reference trajectories of quadrotors are designed as

$$\hat{r}_1(t) = \begin{bmatrix} 0.45\sin(\frac{\pi}{2}t - \frac{\pi}{2}) \\ 0.45\cos(\frac{\pi}{2}t - \frac{\pi}{2}) \\ -0.8 \end{bmatrix}, \hat{r}_2(t) = \begin{bmatrix} 0.45\cos(\frac{\pi}{2}t) \\ 0.45\sin(\frac{\pi}{2}t) \\ -0.8 \end{bmatrix},$$

$$\hat{r}_3(t) = \begin{bmatrix} 0.45\cos(\frac{\pi}{2}t + \frac{\pi}{2}) \\ 0.45\sin(\frac{\pi}{2}t + \frac{\pi}{2}) \\ -0.8 \end{bmatrix}, \hat{r}_4(t) = \begin{bmatrix} 0.45\cos(\frac{\pi}{2}t + \pi) \\ 0.45\sin(\frac{\pi}{2}t + \pi) \\ -0.8 \end{bmatrix},$$

$$\hat{r}_5(t) = \text{BezierInterp}\left(\begin{bmatrix} -0.9 \\ -0.9 \\ -0.8 \end{bmatrix}, \begin{bmatrix} 0.9 \\ 0.9 \\ -0.8 \end{bmatrix} \right).$$

As shown in Fig. 4.13, $Q5$ successfully navigated through the spinning formation with minimal impact on the other four quadrotors by applying the *Safety Barrier Certificates*.



(a) Agents at 24.3s

(b) Agents at 25.3s

(c) Agents at 26.7s

(d) Agents at 27.5s

Figure 4.13: Experimental data of the team of quadrotors plotted in the X-Y plane. The tail of each quadrotor illustrates its trajectory in the past 0.6s.

4.5.3 Showcase: Online Formation Adaptation

The team of quadrotors are commanded to change different formations on the fly in this experiment. Instead of using a precomputed formation switching library, the avoidance behaviors are generated online using the safety barrier certificates.

The reference trajectories of the quadrotors are designed intentionally to make the quadrotors to fly directly over each other. With the following trajectory design, the team will always fly in a close formation. Meanwhile, the quadrotor 2 and 4 will directly fly over

the quadrotor 1 and 3 with a close distance of $0.3m$ if no avoidance measures are taken.

$$\hat{r}_1(t) = \hat{r}_5(t) + \text{BezierInterp}\left(\begin{bmatrix} 0.4 \\ 0 \\ -0.15 \end{bmatrix}, \begin{bmatrix} 0 \\ 0.4 \\ -0.15 \end{bmatrix}\right),$$

$$\hat{r}_2(t) = \hat{r}_5(t) + \text{BezierInterp}\left(\begin{bmatrix} 0 \\ 0.4 \\ 0.15 \end{bmatrix}, \begin{bmatrix} 0.4 \\ 0 \\ 0.15 \end{bmatrix}\right),$$

$$\hat{r}_3(t) = \hat{r}_5(t) + \text{BezierInterp}\left(\begin{bmatrix} -0.4 \\ 0 \\ -0.15 \end{bmatrix}, \begin{bmatrix} 0 \\ -0.4 \\ -0.15 \end{bmatrix}\right),$$

$$\hat{r}_4(t) = \hat{r}_5(t) + \text{BezierInterp}\left(\begin{bmatrix} 0 \\ -0.4 \\ 0.15 \end{bmatrix}, \begin{bmatrix} -0.4 \\ 0 \\ 0.15 \end{bmatrix}\right),$$

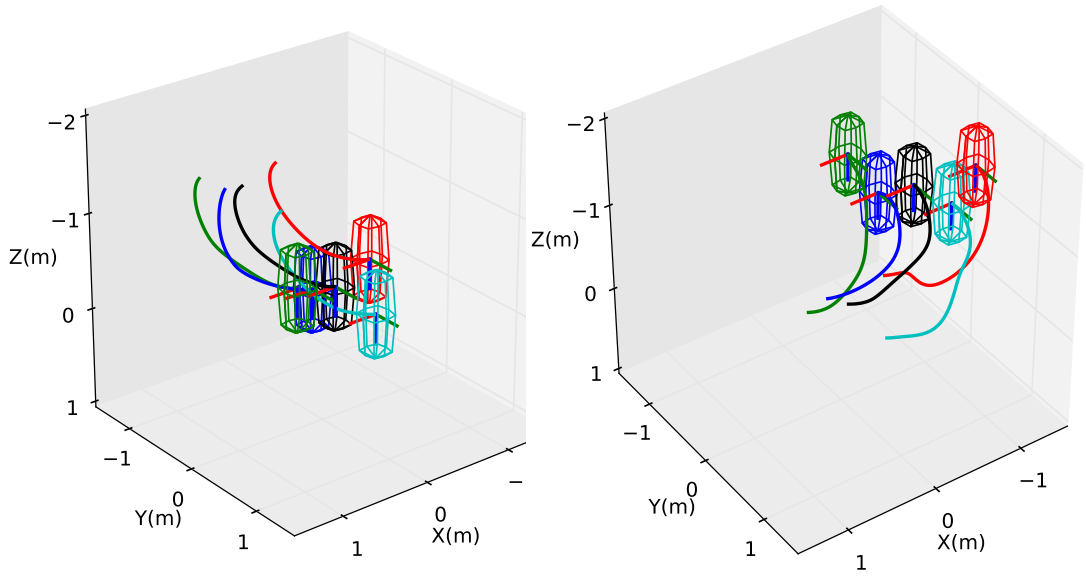
$$\hat{r}_5(t) = \left(\begin{bmatrix} 0.8\cos(\frac{\pi}{3}t) \\ 0.8\sin(\frac{\pi}{3}t) \\ 0.3\sin(\frac{\pi}{2}t) - 1.0 \end{bmatrix}\right),$$

This set of reference trajectories was executed several times without the barrier certificates. Each time, the quadrotor flying below crashed due to the strong down-wash wind disturbance. When the barrier certificates were activated, the actual trajectories of the quadrotors during the experiment are visualized in Fig. 4.14. Since the super-ellipsoidal safe region took care of both the collision and down-wash avoidance requirements, the team successfully changed formations safely with the barrier certificates.

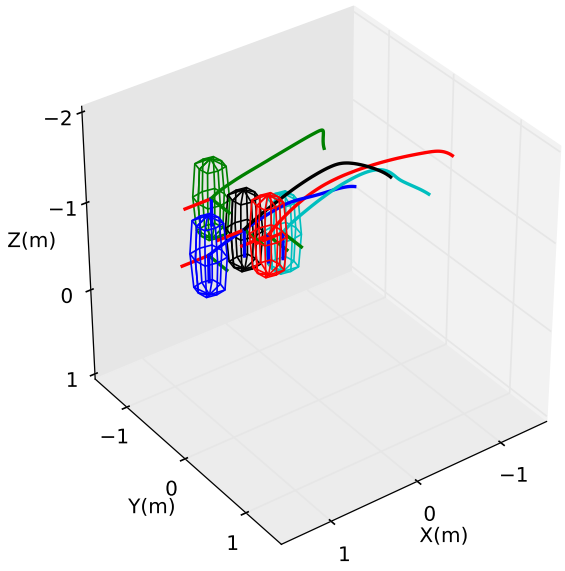
Experimental results provided in this section demonstrate that the *Safety Barrier Certificates* can save flight planners the hassle of considering collision avoidance when designing the higher level multi-robot coordination algorithm. This strategy can be easily used in

conjunction with other complicated motion planning strategies, e.g., optimal control algorithms, temporal/spatial assignment algorithms, to provide desired safety guarantees. The minimally invasive enforcement of the *Safety Barrier Certificates* ensures that desired controller will not be rectified unless truly necessary.

In this chapter, a flight trajectory modification strategy was presented to ensure collision-free manuevers for teams of differential flatness based quadrotors. The nominal flight trajectories, which are generated with existing control and planning algorithms, were modified in a minimally invasive way using the Safety Barrier Certificates to avoid collisions. The effectiveness of the proposed strategy was validated with experimental implementations of the Safety Barrier Certificates on a team of five quadrotors.



(a) $t = 13s$ (b) $t = 15s$



(c) $t = 17s$

Figure 4.14: A team of five quadrotors adapting to different formations on the fly. Each quadrotor is visualized as a super-ellipsoid centered at the quadrotor's center of mass. The tail of each quadrotor represents its flight trajectory in the past $2s$. The team successfully executed collisoin-free trajectories to change formations without prior safety planning. A video of this experiment is available online [74].

CHAPTER 5

BARRIER CERTIFICATES FOR MULTI-OBJECTIVE COMPOSITIONS

Teams of robots often need to simultaneously satisfy multiple objectives, such as area coverage, object tracking, collision avoidance, connectivity maintenance and battery power recharging [89, 90, 91]. Safety barrier certificates were synthesized in previous chapters to ensure collision free motions in teams of robots. But sometimes other objectives need to be achieved together with the safety constraint. In this section, we will explore how to systematically compose multiple barriers representing multiple non-negotiable objectives, and apply it to teams of mobile robots.

5.1 Piecewise Smooth Barrier Functions

To encode more objectives, we will introduce methods to compose barrier functions with AND and OR logical operators. After the composition, these originally smooth barrier functions might become piecewise smooth. Thus, we will first state the result for Piecewise Barrier Functions (PBF) [19].

To ensure easy logical compositions, we redefine the safe set $\mathscr{C} \subseteq \mathscr{D}$ as

$$\mathscr{C} = \{x \in \mathbb{R}^n \mid B(x) > 0\}, \qquad \mathscr{C}^C = \{x \in \mathbb{R}^n \mid B(x) = 0\}, \tag{5.1}$$

where the PC^r-function [92] $B : \mathscr{D} \to \mathbb{R}_0^+$ is positive in $\mathscr{C}$ and zero outside.

Definition 2.3: Given a dynamical system defined in (3.1) and a set $\mathscr{C} \subseteq \mathscr{D}$ defined in (5.1), the PC^r-function $B : \mathscr{D} \to \mathbb{R}_0^+$ is a Piecewise Barrier Function (PBF) if there exists a class $\mathscr{K}$ function α such that

$$\sup_{u \in U}[-B'(x; -f(x) - g(x)u) + \alpha(B(x))] \geq 0, \tag{5.2}$$

for all $x \in \mathscr{C}$.

Note that $B'(x; -f(x) - g(x)u)$ is the B-derivative of $B(x)$ at x in the direction $(-f(x) - g(x)u)$ [93]. When $B(x)$ is smooth, we have $-B'(x; -f(x) - g(x)u) = L_f B(x) + L_g B(x)u$.

With the definition of PBFs, the admissible control space for the control system is

$$K(x) = \{u \in U \mid -B'(x; -f(x) - g(x)u) + \alpha(B(x)) \geq 0\} \tag{5.3}$$

Theorem 5.1.1. *Given a set $\mathscr{C} \subseteq \mathscr{D}$ defined by (5.1) with the associated PBF $B : \mathscr{D} \to \mathbb{R}_0^+$, any Lipschitz continuous controller $u(x) \in K(x)$ for the dynamical system (3.1) render $\mathscr{C}$ forward invariant, i.e., $x(t) \in \mathscr{C}, \forall t \geq t_0$, if $x(t_0) \in \mathscr{C}$.*

Proof. See Appendix. □

To sum up, we can get set invariance properties similar to [16, 62] using PBFs.

5.2 Boolean Logical Composition of Barriers

Let us define $\mathscr{C}_i \subseteq \mathscr{D}, i = 1, 2$, similar to (5.1),

$$\mathscr{C}_i = \{x \in \mathbb{R}^n \mid B_i(x) > 0\}, \qquad \mathscr{C}_i^C = \{x \in \mathbb{R}^n \mid B_i(x) = 0\}. \tag{5.4}$$

Let $B_\cup = B_1 + B_2$ and $B_\cap = B_1 B_2$,

$$\mathscr{E} = \{x \in \mathbb{R}^n \mid B_\cup(x) > 0\}, \qquad \mathscr{F} = \{x \in \mathbb{R}^n \mid B_\cap(x) > 0\}. \tag{5.5}$$

Lemma 5.2.1. *Given $\mathscr{C}_i, i = 1, 2$ defined in (5.4), $\mathscr{E}$ and $\mathscr{F}$ defined in (5.5), $\mathscr{E} = \mathscr{C}_1 \cup \mathscr{C}_2$ and $\mathscr{F} = \mathscr{C}_1 \cap \mathscr{C}_2$.*

Proof. Pick any elements $x_1 \in \mathscr{E}$, $x_2 \in \mathscr{F}$, we have

$$B_{\cup}(x_1) = B_1(x_1) + B_2(x_1) > 0, \tag{5.6}$$

$$B_{\cap}(x_2) = B_1(x_2)B_2(x_2) > 0. \tag{5.7}$$

From the definition (5.4), $B_1(x)$ and $B_2(x)$ are always non-negative. Thus, (5.6) implies $B_1(x_1) > 0$ or $B_2(x_1) > 0$, i.e. $x_1 \in \mathscr{C}_1 \cup \mathscr{C}_2$. (5.7) implies $B_1(x_2) > 0$ and $B_2(x_2) > 0$, i.e. $x_2 \in \mathscr{C}_1 \cap \mathscr{C}_2$. This means $\mathscr{E} \subseteq \mathscr{C}_1 \cup \mathscr{C}_2$ and $\mathscr{F} \subseteq \mathscr{C}_1 \cap \mathscr{C}_2$.

Conversely, we can show that $\mathscr{C}_1 \cup \mathscr{C}_2 \subseteq \mathscr{E}$ and $\mathscr{C}_1 \cap \mathscr{C}_2 \subseteq \mathscr{F}$. This completes the proof. □

Next, we can compose two objectives with AND or OR logical operators.

Theorem 5.2.2. *Given $\mathscr{C}_i, i = 1,2$, defined in (5.4), $\mathscr{E}$ defined in (5.5), and a valid PBF $B_{\cup}$ on $\mathscr{E}$, then any Lipschitz continuous controller $u(x) \in K_{\cup}(x)$ for the dynamical system (3.1) render $\mathscr{C}_1 \cup \mathscr{C}_2$ forward invariant, where*

$$K_{\cup}(x) = \{u \in U \mid -B'_{\cup}(x; -f(x) - g(x)u) + \alpha(B_{\cup}(x)) \geq 0\}.$$

Proof. $B_{\cup}$ is the summation of two PC^r-functions, thus still a PC^r-function [92]. The B-derivative for $B_{\cup}$ is well-defined at $\forall x \in \mathscr{E}$. Since $B_1(x)$ and $B_2(x)$ are always non-negative, $B_{\cup}$ is also non-negative, i.e., $B_{\cup} > 0$ in $\mathscr{E}$, $B_{\cup} = 0$ outside of $\mathscr{E}$.

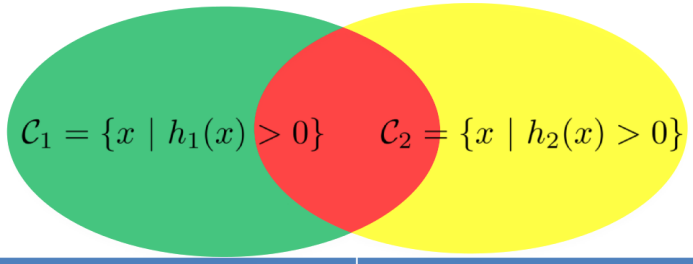
When $u(x) \in K_{\cup}(x)$, we have $\partial_- B_{\cup}x(t) \geq -\alpha(B_{\cup}x)$. Apply *Theorem* 5.1.1, $\mathscr{E}$ is forward invariant. Use *Lemma* 5.2.1, we can get $\mathscr{C}_1 \cup \mathscr{C}_2$ is also forward invariant. □

Theorem 5.2.3. *Given $\mathscr{C}_i, i = 1,2$, defined in (5.4), $\mathscr{F}$ defined in (5.5), and a valid PBF $B_{\cap}$ on $\mathscr{F}$, then any Lipschitz continuous controller $u(x) \in K_{\cap}(x)$ for the dynamical system (3.1) render $\mathscr{C}_1 \cap \mathscr{C}_2$ forward invariant, where*

$$K_{\cap}(x) = \{u \in U \mid -B'_{\cap}(x; -f(x) - g(x)u) + \alpha(B_{\cap}(x)) \geq 0\}.$$

The proof of this theorem is similar to *Theorem* 5.2.2.

Now we have the conditions to check whether the objectives are composable using the AND or OR logical operators. These two logical operators provides easy ways to compose multiple objectives as shown in Fig. 5.1. Next, the compositional barrier functions will be applied to safety and connectivity maintenance for teams of mobile robots.



Logical operation	Barrier composition
$\mathcal{C}_\cup = \mathcal{C}_1 \cup \mathcal{C}_2$	$h_\vee = h_1 + h_2 > 0$
$\mathcal{C}_\cap = \mathcal{C}_1 \cap \mathcal{C}_2$	$h_\wedge = h_1 h_2 > 0$

Figure 5.1: Barrier compositions using AND and OR logical operators.

5.3 Barrier Compositions for Safe and Connected Team of Robots

Similar to Section 3.2, we consider a team of N mobile robots modeled with double integrator dynamics. Two robots i and j need to always keep a safety distance D_s away from each other to avoid collision, meanwhile stay within a connectivity distance D_c of each other to communicate.

Considering the worst case scenario that the maximum braking force is applied to avoid collisions, a pairwise safety constraint between robots i and j can be written as

$$h_{ij}(x) = 2\sqrt{\alpha(\|\Delta p_{ij}\| - D_s)} + \frac{\Delta p_{ij}^T}{\|\Delta p_{ij}\|}\Delta v_{ij} > 0.$$

The detailed derivation of this pairwise safety constraint can be found in [66]. A pairwise

safe set $\mathscr{C}_{ij}$ and a PBF candidate $B_{ij}(x)$ are defined as

$$\mathscr{C}_{ij} = \{x \mid B_{ij}(x) > 0\}, \qquad B_{ij}(x) = \max\{h_{ij}(x), 0\}. \tag{5.8}$$

To guarantee that *all* pairwise collisions are prevented, the safe set $\mathscr{C}$ for the team of mobile robots can be written as the intersection of *all* pairwise safe sets.

$$\mathscr{C} = \bigcap_{\substack{j \in \mathscr{M} \\ j>i}} \mathscr{C}_{ij}. \tag{5.9}$$

Let $\mathscr{G} = (V, E)$ be the required connectivity graph, where $V = \{1, 2, ..., N\}$ is the set of N mobile robots, E is the required edge set. The presence of a required edge (i, j) indicates that robots i and j should always stay within a connectivity distance of D_c.

Similarly, a pairwise connectivity constraint can be developed by considering the maximum acceleration to avoid breaking connectivity, i.e.,

$$\bar{h}_{ij}(x) = 2\sqrt{\alpha(D_c - \|\Delta p_{ij}\|)} - \frac{\Delta p_{ij}^T}{\|\Delta p_{ij}\|}\Delta v_{ij} > 0.$$

The corresponding pairwise connectivity set $\bar{\mathscr{C}}_{ij}$ and PBF candidate are

$$\bar{\mathscr{C}}_{ij} = \{x \mid \bar{B}_{ij}(x) > 0\}, \qquad \bar{B}_{ij}(x) = \max\{\bar{h}_{ij}(x), 0\}. \tag{5.10}$$

In order to maintain *all* required edges, the connectivity set $\bar{\mathscr{C}}$ for the team of mobile robots can be written as

$$\bar{\mathscr{C}} = \bigcap_{(i,j) \in E} \bar{\mathscr{C}}_{ij}. \tag{5.11}$$

In order for the team of mobile robots to stay *safe* and *connected*, the ensemble state x shall stay within

$$\mathscr{T} = \bigcap_{\substack{i,j \in \mathscr{M} \\ j>i}} \mathscr{C}_{ij} \bigcap_{(i,j) \in E} \bar{\mathscr{C}}_{ij}, \tag{5.12}$$

for all time $t \geq 0$. Since $\mathscr{T}$ is the intersection of multiple sets, the compositional barrier function can be used to ensure the forward invariance of $\mathscr{T}$. The composite PBF for safety and connectivity maintenance is

$$B(x) = \prod_{\substack{i,j \in \mathscr{M} \\ j>i}} B_{ij}(x) \prod_{(i,j)\in E} \bar{B}_{ij}(x). \tag{5.13}$$

Before using this composite PBF, we need to check whether $B(x)$ is a valid PBF, which is ensured by the following lemma.

Lemma 5.3.1. *The composite barrier function candidate $B(x)$ defined in (5.13) is a valid PBF, i.e.,*

$$\sup_{u\in U}[-B'(x;-f(x)-g(x)u)+\alpha(B(x))] \geq 0, \tag{5.14}$$

for all $x \in \mathscr{T}$.

Proof. See Appendix. □

Lemma 5.3.1 also implies that the admissible control space,

$$K_{\mathscr{T}}(x) = \{u \in U \mid L_f B(x) + L_g B(x)u + \alpha(B(x)) \geq 0\}, \tag{5.15}$$

is always non-empty. With this result, we will present the theorem for safety and connectivity maintenance.

Theorem 5.3.2. *Given any required connectivity graph $\mathscr{G} = (V,E)$, a PBF $B(x)$ defined in (5.13), any Lipschitz continuous controller $u(x) \in K_{\mathscr{T}}(x)$ for the dynamical system (3.2) guarantees that the team of mobile robots are safe and connected.*

Proof. Lemma 5.3.1 ensures that $B(x)$ is a valid PBF defined for the set $\mathscr{T}$ in (5.12). Thus when $\mathbf{u}(x) \in K_{\mathscr{T}}(x)$, $\mathscr{T}$ is forward invariant from *Theorem* 5.1.1, i.e., $B(x) > 0, \forall t > 0$. From definitions (5.8), (5.10), and (5.13), all PBFs are constructed to be non-negative.

Therefore,

$$B_{ij} > 0, \quad \forall i, j \in \mathscr{M}, j > i, \quad \forall t > 0,$$
$$\bar{B}_{ij} > 0, \qquad \forall (i,j) \in E, \qquad \forall t > 0.$$

Both $\mathscr{C}$ and $\bar{\mathscr{C}}$ are forward invariant. $\mathscr{C}$ encodes that all agents do not collide with each other, while $\bar{\mathscr{C}}$ encodes that all connectivity requirements specified by the graph $\mathscr{G}$ are satisfied, i.e., the team of mobile robots are *safe* and *connected*. □

Next, an optimization based controller will be presented to inject higher level goals, e.g., visiting waypoints, form certain shapes, and covering area, into the controller design. This is achieved by running the following QP-based controller,

$$\begin{aligned} u^* = \underset{u}{\text{argmin}} \quad & J(u) = \sum_{i=1}^{N} \|u_i - \hat{u}_i\|^2 \\ \text{s.t.} \quad & L_f B(x) + L_g B(x) u + \alpha(B(x)) \geq 0, \\ & \|u_i\|_\infty \leq \alpha_i, \ \forall i \in \mathscr{M}. \end{aligned} \tag{5.16}$$

which is similar to what we did in Section 3.2.

5.3.1 Maintaining Dynamical Connectivity Graphs

Due to the dynamically changing environment and robot states, it would sometimes be favourable to allow the robots to switch between different connectivity graphs [94]. Motivated by the need of maintaining dynamically changing connectivity graphs, composite safety and connectivity barrier certificates are proposed to ensure safety and dynamical connectivity of the team of mobile robots.

Let $\tilde{\mathscr{G}} = \{\mathscr{G}_1, \mathscr{G}_2, ..., \mathscr{G}_M\}$ denote the set of all allowable connectivity graphs, where $\mathscr{G}_i = (V, E_i), i \in \mathscr{P}$, $\mathscr{P} = \{1, 2, ..., M\}$ is the index set of $\tilde{\mathscr{G}}$. To stay connected, the team of mobile robots needs to satisfy at least one of these allowable connectivity graphs. The set

that encodes the dynamical connectivity graph requirement is

$$\tilde{\mathscr{C}} = \bigcup_{k\in\mathscr{P}} \bigcap_{(i,j)\in E_k} \bar{\mathscr{C}}_{ij} \tag{5.17}$$

Definition 4.3: Given a set of allowable connectivity graphs $\tilde{\mathscr{G}}$, the team of N mobile robots with dynamics given in (3.2) is *dynamically connected*, if the ensemble state x stays in the set $\tilde{\mathscr{C}}$ for all time $t \geq 0$.

In order for the team of mobile robots to stay both *safe* and *dynamically connected*, the ensemble state x shall stay in

$$\tilde{\mathscr{T}} = \left(\bigcap_{\substack{i,j\in\mathscr{M} \\ j>i}} \mathscr{C}_{ij} \right) \left(\bigcup_{k\in\mathscr{P}} \bigcap_{(i,j)\in E_k} \bar{\mathscr{C}}_{ij} \right), \tag{5.18}$$

for all time $t \geq 0$. Safety and dynamical connectivity guarantees similar to *Theorem* 5.3.2 can be achieved by using a composite PBF introduced in Section 5.2,

$$\tilde{B}(x) = \left(\prod_{\substack{i,j\in\mathscr{M} \\ j>i}} B_{ij}(x) \right) \left(\sum_{k\in\mathscr{P}} \prod_{(i,j)\in E_k} \bar{B}_{ij}(x) \right). \tag{5.19}$$

It can be shown that $\tilde{B}(x)$ is a valid PBF on $\tilde{\mathscr{T}}$ using the same techniques like *Lemma* 5.3.1, i.e., the admissible control space

$$K_{\tilde{\mathscr{T}}}(x) = \{\mathbf{u} \in U \mid L_f\tilde{B}(x) + L_g\tilde{B}(x)\mathbf{u} + \alpha(\tilde{B}(x)) \geq 0\}, \tag{5.20}$$

is always non-empty.

Theorem 5.3.3. *Given a set of allowable connectivity graphs $\tilde{\mathscr{G}} = \{\mathscr{G}_1, \mathscr{G}_2, ..., \mathscr{G}_M\}$, a PBF $\tilde{B}(x)$ defined in (5.19), any Lipschitz continuous controller $\mathbf{u}(x) \in K_{\tilde{\mathscr{T}}}(x)$ for the dynamical system (3.2) guarantees that the team of mobile robots are safe and dynamically connected.*

The proof of this theorem is similar to *Lemma* 5.3.1, *Theorem* 5.2.2, and *Theorem* 5.3.2.

5.4 Experimental Implementation

The composite safety and connectivity barrier certificates were tested on a team of four Khepera robots. The real-time positions of the robots are tracked by the Optitrack Motion Capture System. The mutli-robot communications and controls are executed on the Robot Operating System (ROS).

The nominal controller was designed as a waypoint controller, which used a go-to-goal behavior to visit the specified waypoints without considering safety and connectivity. As illustrated in Fig. 5.2, each robot needs to visit three waypoints sequencially. Those waypoints are intentionally designed to make robots collide at multiple places.

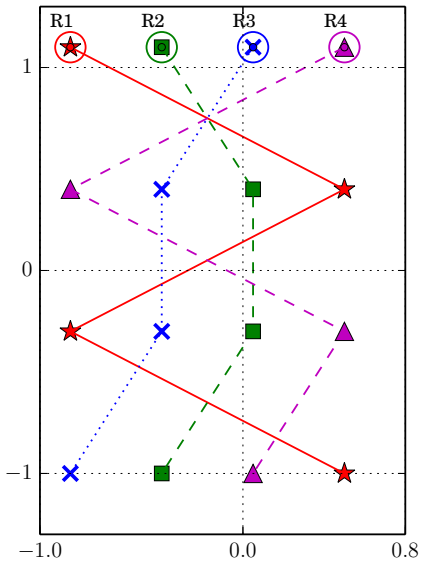


Figure 5.2: Planned waypoints for four robot agents. Ri stands for robot i, where $i = 1,2,3,4$. The lines represent the nominal trajectories of the robots if they execute the nominal waypoint controller.

5.4.1 Composite Safety Barrier Certificates

In the first experiment, the composite safety barrier certificates were wrapped around the nominal waypoint controller using the QP-based strategy (5.16). The composite PBF was formulated as

$$B = B_{12}B_{13}B_{14}B_{23}B_{24}B_{34},$$

so that all possible pairwise collisions are avoided. No connectivity constraints were considered in this experiment.

As shown in Fig. 5.3, all the inter-robot distances are always larger than the safety distance D_s, i.e., no collision happened during the experiment. Fig. 5.5 are snapshots taken by an overhead camera and plotted robot trajectories. All robots successfully visited the specified waypoints without colliding into each other. Note that without the connectivity constraints, the mobile robot team sometimes got disconnected during the experiment, e.g., the team split into two parts in 5.5a.

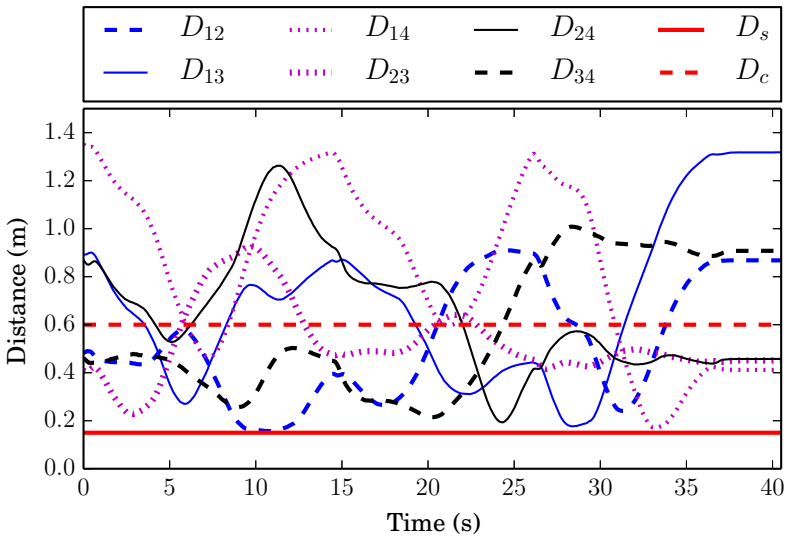


Figure 5.3: Evolution of the inter-robot distances during the experiment. D_{ij} represents the distance between robot i and robot j. $D_s = 0.15m$ and $D_c = 0.6$ are the safety and connectivity distance. $D_{ij} > D_s$ implies that robots i and j did not collide.

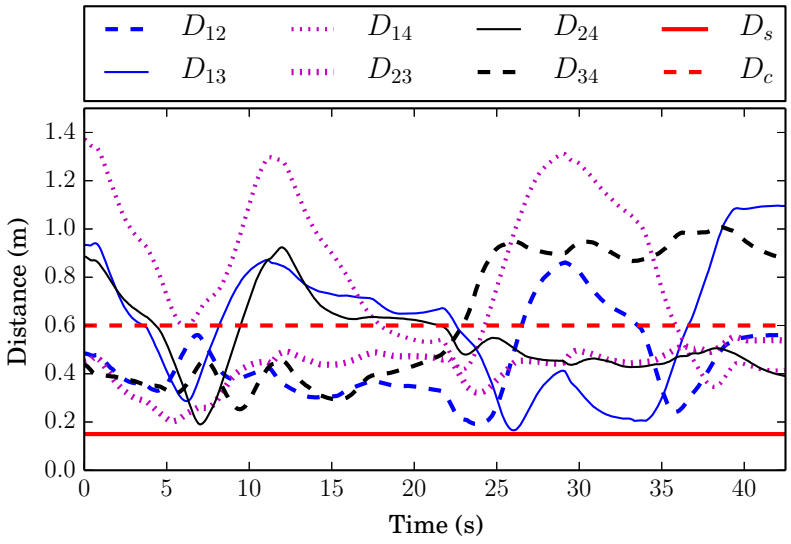


Figure 5.4: Evolution of the inter-robot distances during the experiment. D_{ij} represents the distance between robot i and robot j. $D_s = 0.15m$ and $D_c = 0.6$ are the safety and connectivity distance. $D_{ij} > D_s$ implies that robots i and j do not collide. $D_{ij} < D_c$ implies that robots i and j are in connectivity range.

5.4.2 Composite Safety and Connectivity Barrier Certificates

During the second experiment, the composite safety and connectivity barrier certificates were wrapped around the waypoint controller using the QP-based strategy (5.16). The composite PBF is designed as

$$B = B_{12}B_{13}B_{14}B_{23}B_{24}B_{34}\bar{B}_{23}(\bar{B}_{12}+\bar{B}_{13})(\bar{B}_{24}+\bar{B}_{34}),$$

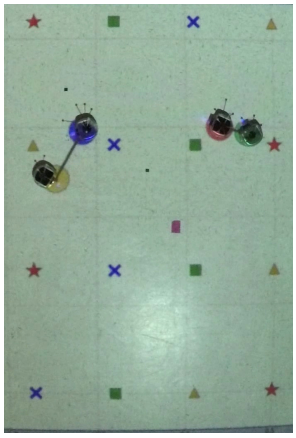
which encodes that: 1) there should be no inter-robot collisions; 2) robot 2 and 3 should always be connected; 3) robot 1 should be connected to robot 2 or 3; 4) robot 4 should be connected to robot 2 or 3.

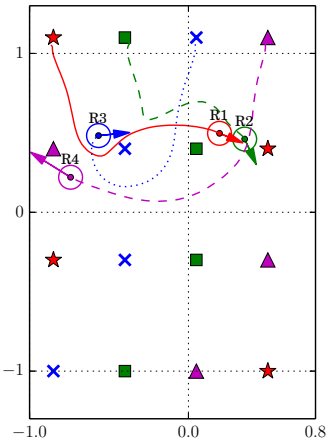
As shown in Fig. 5.4, the inter-robot distances were always larger than D_s, i.e., the team of mobile robots did not collide with each other during the experiment. At the same time, all the connectivity constraints were satisfied, i.e., 1) D_{23} was always smaller than D_c; 2) $\min\{D_{12}, D_{13}\}$ was always smaller than D_c; 2) $\min\{D_{24}, D_{34}\}$ was always smaller than D_c.

The team of mobile robots satisfied all the safety and connectivity requirements specified by the safety and connectivity barrier certificates.

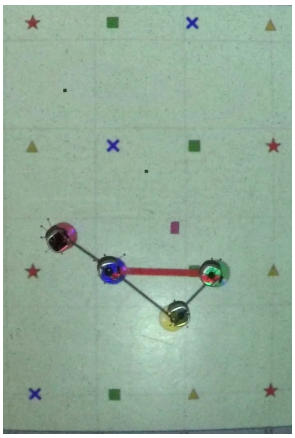
The snapshots during the experiment in Fig. 5.6 illustrated that the robots visited all specified waypoints except the last one. This is because the last set of waypoints violated the connectivity constraints, i.e., robot 1 can't reach its waypoint without breaking its connectivity to robot 2 and 3. This experiment also indicates that not all higher level objectives are compatible with the safety and connectivity constraints.

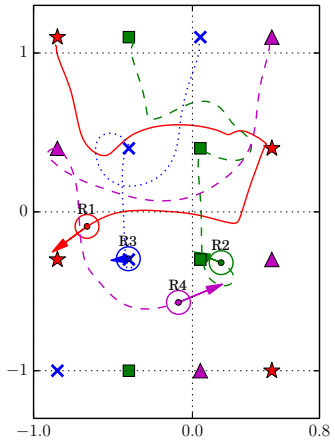
In this section, a systematic way to compose multiple objectives using the compositional barrier functions was presented. AND and OR logical operators were designed to provably compose multiple non-negotiable objectives, with conditions for composibility provided. Since the non-negotiable objectives are not guaranteed to be achieved, a natural problem to address is that how to attain all the objectives simultaneously. We will explore the solution to this problem in the following section in a computational framework.



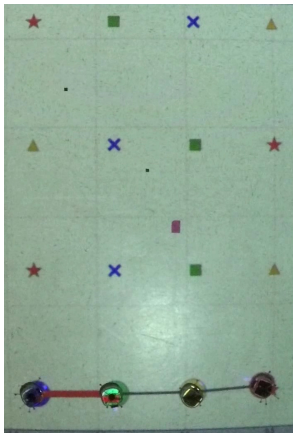


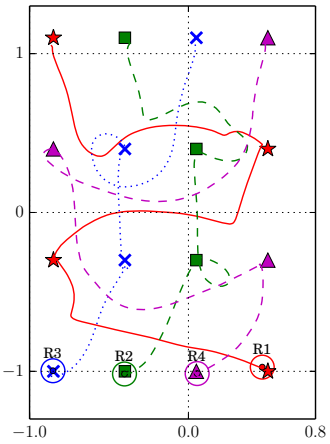
(a) Agents at 10.0s





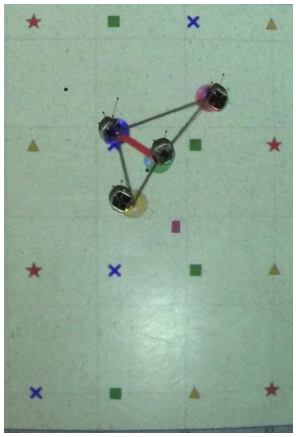
(b) Agents at 23.0s



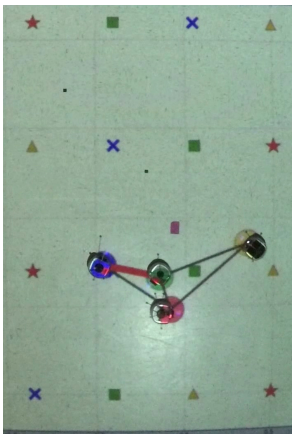


(c) Agents at 36.0s

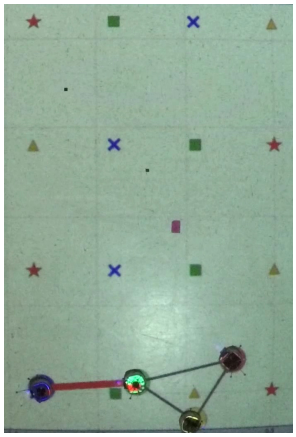
Figure 5.5: Experiement of four mobile robots executing waypoint controller regulated by safety barrier certificates. Pictures on the left are taken by an overhead camera. The star, square, cross and triangular markers representing waypoints are projected onto the ground. A straght line connecting two robots were projected onto the ground if the two robots are closer than $D_c = 0.6m$.



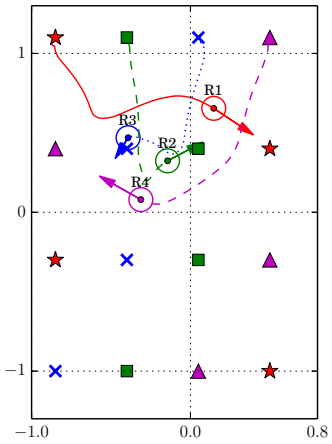
(a) Agents at 8.0s

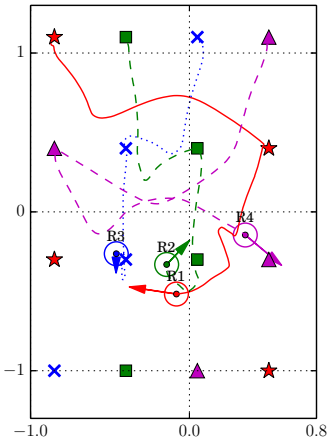


(b) Agents at 25.0s



(c) Agents at 42.5s





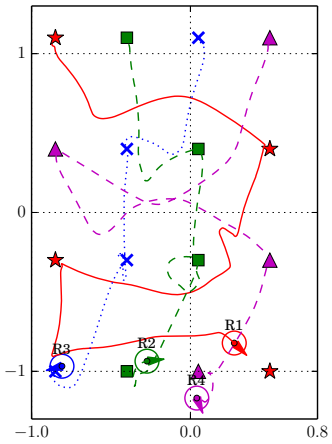


Figure 5.6: Experiment of four mobile robots executing waypoint controllers regulated by safety and connectivity barrier certificates. The safety and connectivity distances are $D_s = 0.15m$ and $D_c = 0.6m$. The lines representing inter-robot connectivity are projected onto the ground using a projector. A video of the experiment can be found online [95].

5.5 Permissive Barrier Certificates and Sum-of-Squares Programming

The compositional barrier certificates, encoded with CBFs, provides a way to simultaneously ensure multiple non-negotiable safety constraints. The negotiable higher level objectives, encoded with CLFs or other similar tools, are enforced with minimal modifications possible, i.e., not necessarily achieved.

Since a common control that satisfies both the CBFs and the CLFs does not necessarily exist, a typical way to unite the pre-designed CLF and CBF is to use a QP-based controller [63, 16, 18], i.e.,

$$\begin{aligned} u^* \quad &= \underset{u\in\mathbb{R}^n}{\operatorname{argmin}}\ J(u)+k_\delta\delta^2 \\ \text{s.t.} \quad &\frac{\partial V(x)}{\partial x}g(x)u \le -\frac{\partial V(x)}{\partial x}f(x)+\delta, \\ &-\frac{\partial h(x)}{\partial x}g(x)u \le \frac{\partial h(x)}{\partial x}f(x)+\kappa(h(x)), \end{aligned} \tag{5.21}$$

where δ is a CLF relaxation factor, such that the non-negotiable safety constraint is always satisfied. However, simultaneous stabilization and safety enforcement are not guaranteed. In certain cases described in Section 3.4, deadlocks might occur so that the higher level objectives can never be achieved without appropriate perturbations.

In this Chapter, instead of relaxing the stabilization term, we will compute an estimate of the region of safe stabilization with permissive barrier certificates, such that both the stabilization and safety constraints are strictly respected [20].

5.5.1 Safe Stabilization for Autonomous Dynamical Systems

Computing estimates of the region of safe stabilization is closely related to computing estimates of DoA, because both try to maximize the volume of interested region where certain matrix inequalities are satisfied. In this section, we will show that the DoA estimate derived with barrier certificates is strictly larger than the maximum contractive sublevel set of the Laypunov function. An iterative optimization algorithm based on SOS program is provided to numerically compute the most permissive barrier certificates for polynomial

systems. Building upon the results developed in this section, permissive barrier certificates for safe stabilization will be presented in Section 5.5.2.

Estimating the region of safe stabilization is closely related to estimating the DoA of an equilibrium state, except for the extra consideration of safety constraints. Among the various DoA approximation methods proposed in the literature, methods using the subset of Lyapunov-like functions, such as quadratic Lyapunov functions [96] and rational polynomial Lyapunov functions [97], are proved to be effective [98]. Further improvements on the Lyapunov sublevel set based methods are developed in [99, 100, 101, 102] to reduce the conservativeness with invariant sets. In this section, the set invariance property is established with barrier certificates, which are allowed to take arbitrary shapes rather than the sublevel set of the Lypapunov function. This method leads to a non-conservative estimate of the DoA.

Expanding Estimate of DoA with Barrier Certificates

For generality of discussion, the autonomous dynamical system dealt with here is

$$\dot{x} = f(x), \tag{5.22}$$

where $x \in \mathscr{X}$, and f is locally Lipschitz continuous. Similarly, the control dynamical system is considered as

$$\dot{x} = f(x) + g(x)u, \tag{5.23}$$

where $x \in \mathscr{X}$ and $u \in U$ are the state and control of the system, and f and g are both locally Lipschitz continuous.

Assume the system (5.22) is locally asymptotically stable at the origin. Let $\psi(t;x_0)$ denote the state trajectory of the system (5.22) starting from x_0. The DoA of the origin is defined as the set of all initial states which eventually converge to the origin as time goes

to infinity,

$$\mathscr{D} = \{x_0 \in \mathscr{X} \mid \lim_{t\to\infty} \psi(t;x_0) = 0\}.$$

A commonly used method to estimate the DoA is to compute the sublevel set of a given Lyapunov function $V(x)$. This Lyapunov function should be positive definite, and its derivative should be locally negative definite. Let $\mathscr{V}(c) = \{x \in \mathscr{X} \mid V(x) \leq c\}$ be a sublevel set of $V(x)$. The largest inner estimate of the DoA using the sublevel set of the Lyapunov function can be computed with

$$\begin{aligned} c^* = \quad & \max_{c\in\mathbb{R}} c \\ \text{s.t.} \quad & -\frac{\partial V(x)}{\partial x} f(x) > 0, \quad \forall x \in \mathscr{V}(c) \setminus \{0\}. \end{aligned} \tag{5.24}$$

The estimate $\mathscr{V}(c^*)$ is straightforward to compute, but often conservative compared to invariant set based methods. This is because the shape of $\mathscr{V}(c^*)$ is restricted to the Lyapunov sublevel set.

Next, we will show that the estimate of DoA can be further expanded using barrier certificates and the given Lyapunov function. This is achieved by allowing the barrier certificates to take an arbitrary shape instead of the sublevel set of $V(x)$. The most permissive barrier certified region $\mathscr{C} = \{x \in \mathscr{X} \mid h(x) \geq 0\}$ can be computed as,

$$\begin{aligned} h^*(x) = \operatorname*{argmax}_{h(x)\in\mathscr{P}} \quad & \mu(\mathscr{C}) \\ \text{s.t.} \; -\frac{\partial V(x)}{\partial x} f(x) > 0, \quad & \forall x \in \mathscr{C} \setminus \{0\}, \\ \frac{\partial h(x)}{\partial x} f(x) \geq -\kappa(h(x)), \quad & \forall x \in \mathscr{C}, \end{aligned} \tag{5.25}$$

where $\mu(\mathscr{C})$ is the volume of $\mathscr{C}$. The largest estimate of the DoA with barrier certificates is achieved with $\mathscr{C}^* = \{x \in \mathscr{X} \mid h^*(x) \geq 0\}$. By maximizing the volume of the barrier certified region, $\mathscr{C}^*$ is guaranteed to be larger than $\mathscr{V}(c^*)$. This fact can be shown with the following lemma.

Lemma 5.5.1. *Given an autonomous system (5.22) that is locally asymptotically stable at the origin, the estimate of DoA with barrier certificates is larger than the estimate with the sublevel set of Lyapunov function, i.e.,* $\mu(\mathscr{V}(c^*)) \leq \mu(\mathscr{C}^*)$.

Proof. The largest inner estimate of DoA using the sublevel set of a given Lyapunov function is $\mathscr{V}(c^*) = \{x \in \mathscr{X} \mid V(x) \leq c^*\}$. A candidate barrier certificate can be designed as $\bar{h}(x) = c^* - V(x)$, and the corresponding certified safe region is $\bar{\mathscr{C}} = \{x \in \mathscr{X} \mid \bar{h}(x) \geq 0\}$. The time derivative of $\bar{h}(x)$ is

$$\frac{\partial \bar{h}(x)}{\partial x} f(x) = -\frac{\partial V(x)}{\partial x} f(x), \quad \forall x \in \bar{\mathscr{C}},$$

which is always nonnegative within $\bar{\mathscr{C}}$. By definition, $\bar{h}(x)$ is also nonnegative in $\bar{\mathscr{C}}$, i.e.,

$$\frac{\partial \bar{h}(x)}{\partial x} f(x) \geq 0 \geq -\kappa(\bar{h}(x)), \quad \forall x \in \bar{\mathscr{C}},$$

which means $\bar{h}(x)$ is a valid barrier certificate and a feasible solution to (5.25). But $\bar{h}(x)$ is not necessarily the optimal solution. So we have $\mu(\mathscr{V}(c^*)) = \mu(\bar{\mathscr{C}}) \leq \mu(\mathscr{C}^*)$. □

Remark 1: With *Lemma* 5.5.1, (5.24) can be reformulated into an optimization problem similar to (5.25), i.e.,

$$\begin{aligned} c^* = \quad & \max_{c \in \mathbb{R}} \ c \\ \text{s.t.} \quad & -\frac{\partial V(x)}{\partial x} f(x) > 0, && \forall x \in \mathscr{V}(c) \setminus \{0\}, \\ & \frac{\partial (c - V(x))}{\partial x} f(x) \geq -\kappa(c - V(x)), && \forall x \in \mathscr{V}(c). \end{aligned}$$

We can see that (5.24) also searches for a maximum barrier certificate. The shape of the certified region is constrained to be a sublevel set of $V(x)$. Since a specific shape of the certified region is not required, (5.25) is more permissive than (5.24). In addition, $h(x)$ is allowed to decrease within the estimated DoA instead of monotone increasing.

The fact that $\mathscr{C}^*$ is an inner estimate of the DoA can be established with the following theorem.

Theorem 5.5.2. *Given an autonomous dynamical system (5.22) that is locally asymptotically stable at the origin, the estimate of the DoA with barrier certificates, $\mathscr{C}^*$, is a subset of the true DoA $\mathscr{D}$. And $\mathscr{C}^*$ is guaranteed to be non-empty.*

Proof. Given an arbitrary initial state $x_0 \in \mathscr{C}^*$, the trajectory of the state $\psi(t;x_0), t \in [0,\infty)$, is guaranteed to be contained within $\mathscr{C}^*$, due to the forward invariance property of barrier certificates.

By the construction of $\mathscr{C}^*$ in (5.25), $\frac{\mathrm{d}V(\psi(t;x_0))}{\mathrm{d}t}$ is negative definite for $\psi(t;x_0) \in \mathscr{C}^*$. Therefore, $V(\psi(t;x_0))$ is strictly decreasing along the trajectory $\psi(t;x_0), t \in [0,\infty)$, except at 0_n. Since $V(x_0)$ is bounded and 0_n is the only equilibrium point in $\mathscr{C}^*$, we can get $\lim_{t\to\infty} \psi(t;x_0) = 0_n$. By the definition of the DoA, $x_0 \in \mathscr{D}$ for any $x_0 \in \mathscr{C}^*$,which means $\mathscr{C}^* \subseteq \mathscr{D}$.

It is shown in [42] that $\mathscr{V}(c^*)$ is non-empty. From Lemma 5.5.1, $\mu(\mathscr{V}(c^*)) \leq \mu(\mathscr{C}^*)$, thus $\mathscr{C}^*$ is also non-empty. □

Iterative Search of Permissive Barrier Certificates

The optimization problem (5.25) is difficult to solve for general systems, since checking non-negativity is often computationally intractable [41]. However, if non-negativity constraints are relaxed to SOS constraints, (5.25) can be converted to a numerically efficient convex optimization problem. To this end, we will restrict (5.22) to polynomial dynamical systems.

Let $\mathscr{P}$ be the set of polynomials for $x \in \mathbb{R}^n$. The polynomial $l(x)$ can be written in Square Matrix Representation (SMR) [42] as $Z^T(x)QZ(x)$, where $Z(x)$ is a vector of monomials, and $Q \in \mathbb{R}^{k\times k}$ is a symmetrical coefficient matrix. A polynomial function $l(x)$ is non-negative if $l(x) \geq 0, \forall x \in \mathbb{R}^n$. Furthermore, $p(x)$ is a SOS polynomial if $p(x) = \sum_{i=1}^{m} p_i^2(x)$

for some $p_i(x) \in \mathscr{P}$. $\mathscr{P}^{\text{SOS}}$ is the set of SOS polynomials. If written in SMR form, $p(x)$ has a positive semidefinite coefficient matrix $Q \succeq 0$. The trace and determinant of a square matrix $A \in \mathbb{R}^{n\times n}$ are trace(A) and det(A), respectively.

Since the proposed method is an under-approximation method, we would like to maximize the volume of $\mathscr{C}$ such that the best estimate of DoA can be achieved. However, this objective max(vol$(\mathscr{C})$) is non-convex and usually cannot be described by an explicit mathematical expression. In order to solve this issue, a typical way adopted in the literature is to approximate the volume by using trace(Q), where $h(x) = Z(x)^T QZ(x)$. Here, we would like to maximize trace(Q) to get the largest $\mathscr{C}$ similar to [42].

To deal with nonnegativity constraints over semialgebraic sets, we will introduce the Positivestellensatz (P-satz).

Lemma 5.5.3. *([103]) For polynomials $a_1,\ldots,a_m$, $b_1,\ldots,b_l$ and p, define a set*

$$\begin{aligned}\mathscr{B} \;=\; & \{x \in \mathbb{R}^n : a_i(x) = 0,\ \forall i = 1,\ldots,m,\\ & b_i(x) \geq 0,\ \forall j = 1,\ldots,l\}.\end{aligned}$$

Let $\mathscr{B}$ be compact. The condition $p(x) > 0, \forall x \in \mathscr{B}$ holds if the following condition is satisfied:

$$\begin{cases} \exists r_1,\ldots,r_m \in \mathscr{P},\ s_1,\ldots,s_l \in \mathscr{P}^{SOS}, \\ p - \sum_{i=1}^{m} r_i a_i - \sum_{i=1}^{l} s_i b_i \in \mathscr{P}^{SOS}. \end{cases}$$

This lemma provides an important perspective that any strictly positive polynomial $p(x) \in \mathscr{F}$ is actually in the cone generated by a_i and b_i. Using the Real P-satz and the

SMR form of $h(x)$, (5.25) can be formulated into a SOS program,

$$\begin{aligned} &\max_{\substack{h(x)\in\mathscr{P},\ L_1(x)\in\mathscr{P}^{\mathrm{SOS}}\\ L_2(x)\in\mathscr{P}^{\mathrm{SOS}}}} && \mathrm{Trace}(Q) \\ \text{s.t.} \quad & -\frac{\partial V(x)}{\partial x}f(x) - L_1(x)h(x) \in \mathscr{P}^{\mathrm{SOS}}, \\ & \frac{\partial h(x)}{\partial x}f(x) + \gamma h(x) - L_2(x)h(x) \in \mathscr{P}^{\mathrm{SOS}}, \end{aligned} \tag{5.26}$$

where a linear function $\kappa(x) = \gamma x$ is adopted. The SOS program (5.26) involves bilinear decision variables. It can be solved efficiently by splitting into several smaller SOS programs, which leads to the following iterative search algorithm.

Remark 2: Notice that (5.26) requires an initial value of $h(x)$ to start with. From *Lemma* 5.5.1, a good initial value can be picked as $\bar{h}(x) = c^* - V(x)$. This SOS program is guaranteed to generate a barrier certificate better than $\bar{h}(x)$.

Algorithm 1:

Step 1: Calculate an initial value for $h(x)$

Specify a Lyapunov function $V(x)$, and find c^* using the bilinear search method, i.e.,

$$\begin{aligned} c^* = \quad & \max_{c\in\mathbb{R}, L(x)\in\mathscr{P}^{\mathrm{SOS}}} c \\ \text{s.t.} \quad & -\frac{\partial V(x)}{\partial x}f(x) - L(x)(c - V(x)) \in \mathscr{P}(x)^{\mathrm{SOS}}. \end{aligned}$$

Set the initial value for $h(x)$ as $\bar{h}(x) = c^* - V(x)$.

Step 2: Fix $h(x)$*, and search for* $L_1(x)$ *and* $L_2(x)$

Using the $h(x)$ from previous step, we can search for $L_1(x)$ and $L_2(x)$ that give the

largest margin on the barrier constraint. This is achieved by solving

$$\begin{aligned} & \max_{\substack{\varepsilon\geq 0,\ L_1(x)\in\mathscr{P}^{\mathrm{SOS}} \\ L_2(x)\in\mathscr{P}^{\mathrm{SOS}}}} \quad \varepsilon \\ \text{s.t.} \quad & -\frac{\partial V(x)}{\partial x} f(x) - L_1(x)h(x) \in \mathscr{P}^{\mathrm{SOS}}, \\ & \frac{\partial h(x)}{\partial x} f(x) + \gamma h(x) - L_2(x)h(x) - \varepsilon \in \mathscr{P}^{\mathrm{SOS}}. \end{aligned}$$

Step 3: Fix $L_1(x)$ and $L_2(x)$, and search for $h(x)$

With $L_1(x)$ and $L_2(x)$ from previous step, a most permissive barrier certificate can be searched for. The barrier certificate is written in the SMR form $h(x) = Z(x)^T Q Z(x)$. The most permissive barrier certificate is computed by maximizing the trace of Q,

$$\begin{aligned} & \max_{h(x)\in\mathscr{P}} \quad \mathrm{trace}(Q) \\ \text{s.t.} \quad & -\frac{\partial V(x)}{\partial x} f(x) - L_1(x)h(x) \in \mathscr{P}^{\mathrm{SOS}}, \\ & \frac{\partial h(x)}{\partial x} f(x) + \gamma h(x) - L_2(x)h(x) \in \mathscr{P}^{\mathrm{SOS}}. \end{aligned}$$

This searching process is terminated if $\mathrm{trace}(Q)$ stops increasing, otherwise go back to *Step 2*.

Remark 3: In *Step 2*, the common approach is to just search for feasible $L_1(x)$ and $L_2(x)$. However, there are multiple $L_1(x)$ and $L_2(x)$ available. By maximizing the margin ε of the barrier constraint, better options of $L_1(x)$ and $L_2(x)$ can be chosen. This method will expand the feasible space of $h(x)$ for optimization in *Step 3*, which can help speed up the optimization procedure.

Simulation Results for Autonomous Dynamical Systems

The iterative search algorithm **1** is implemented on two examples of autonomous dynamical systems. In the simulation, the Matlab toolboxes SeDuMi [104], SMRSOFT [42],

SOSTOOLS[105], and YALMIP [106] are used for solving the semidefinite and SOS programming problems.

Example 1: Given the two-dimensional autonomous system

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ -x_1 - x_2 - x_1^3 \end{bmatrix},$$

which has a locally stable equilibrium at the origin. A forth order Lyapunov function for this system can be picked as $V(x) = x_1^2 + x_1x_2 + x_2^2 + x_1^4 + x_2^4$. Using the sublevel set of $V(x)$, we can get the largest estimate of DoA as

$$\mathscr{A}_1 = \{x \in \mathbb{R}^2 \mid V(x) \leq 0.9759\}.$$

With the iterative search algorithm for barrier certificates, a larger estimate of DoA can be obtained as

$$\begin{aligned} \mathscr{A}_2 = \{x \in \mathbb{R}^2 \mid h(x) = 0.0428 + 0.0033x_1^2 - 0.1396x_1x_2 \\ +0.0206x_2^2 - 0.0976x_1^4 - 0.0913x_2^4 - 0.0079x_1^3x_2 \\ +0.0061x_1x_2^3 + 0.0779x_1^2x_2^2 \geq 0\}. \end{aligned}$$

For comparison under the same condition, the order of the barrier certificate is also restricted to be forth-order. As illustrated in Fig. 5.7, the barrier certificate expands the estimate of DoA significantly.

Example 2: Consider the three-dimensional system

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} -x_1 + x_2x_3^2 \\ -x_2 \\ -x_3 \end{bmatrix},$$

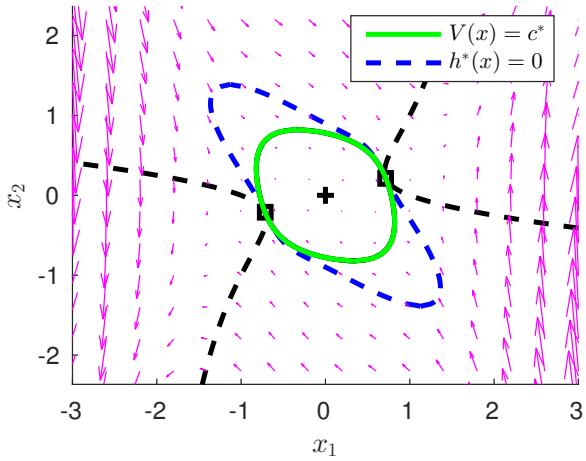


Figure 5.7: Estimates of DoA for a two-dimensional autonomous dynamical system. The barrier certified DoA estimate (region enclosed by the dashed blue curve) is significantly larger than the Lyapunov sublevel set based DoA estimate (region enclosed by the solid green curve).

which has a locally stable equilibrium at the origin. A Lyapunov function for this system can be picked as $V(x) = x_1^2 + x_2^2 + x_3^2$. The largest estimate of DoA based on the sublevel set of Lyapunov function is

$$\mathscr{A}_1 = \{x \in \mathbb{R}^3 \mid V(x) \leq 8\}.$$

With barrier certificates, the largest estimate of the DoA is

$$\begin{aligned}\mathscr{A}_2 = \{x \in \mathbb{R}^3 \mid h(x) = 7.9999 - 1.2828x_3^2 - 0.2850x_1^2 \\ -0.5652x_2^2 - 0.6685x_1x_2 \geq 0\}.\end{aligned}$$

The barrier certificate is restricted to the same order as $V(x)$. Both estimates of DoA are illustrated in Fig.5.8. Since both regions are ellipsoids, the volume of the estimated DoA can be analytically calculated. With the barrier certificate, the volume of the estimated region is increased by $\frac{\mu(\mathscr{A}_2)-\mu(\mathscr{A}_1)}{\mu(\mathscr{A}_1)} = 297.4\%$.

From these two examples, we can see that the barrier certificate based method provides a more permissive estimate of the DoA than the Lyapunov sublevel set based method.

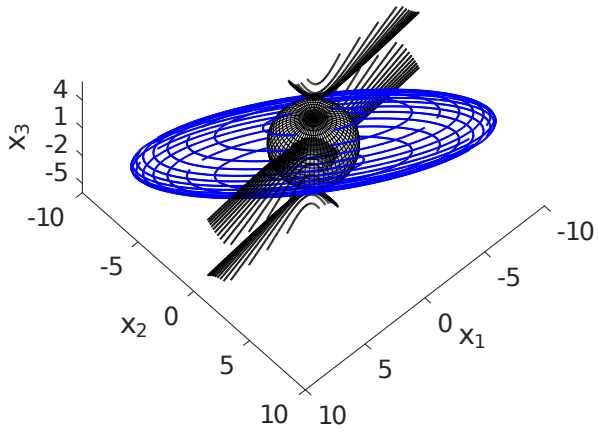


Figure 5.8: Estimates of DoA for a three-dimensional autonomous dynamical system. The black and blue ellipsoids represent the largest estimate of DoA based on the Lyapunov function sublevel set and barrier certificates, respectively.

5.5.2 Safe Stabilization for Control Dynamical Systems

Permissive barrier certificates are developed in this section to maximize the estimated region of safe stabilization, where the system state is both stabilized and contained within the safe set. Based on the DoA estimation method for autonomous systems in section 5.5.1, the safe stabilization of control dynamical systems is addressed.

We will consider the safe stabilization problem described by (5.21) for a locally stabilizable control-affine dynamical system (5.23). Instead of relaxing the stabilization term with δ to resolve conflicts, we will synthesize a permissive barrier certificate with the maximum volume possible that strictly respects both the stabilization and safety constraints. This permissive barrier certificate can be found using

$$
\begin{aligned}
h^*(x) = \underset{h(x)\in\mathscr{P},u(x)\in\mathscr{P}}{\operatorname{argmax}} \quad & \mu(\mathscr{C}) \\
\text{s.t.} \quad -\frac{\partial V(x)}{\partial x} f(x) - \frac{\partial V(x)}{\partial x} g(x)u(x) > 0, \quad & \forall x \in \mathscr{C} \setminus \{0\}, \\
\frac{\partial h(x)}{\partial x} f(x) + \frac{\partial h(x)}{\partial x} g(x)u(x) + \kappa(h(x)) \geq 0, \quad & \forall x \in \mathscr{C},
\end{aligned}
\tag{5.27}
$$

where $\mu(\mathscr{C})$ is the volume of the certified safe region ($\mathscr{C} = \{x \in \mathscr{X} \mid h(x) \geq 0\}$). Note that (5.27) is a semi-infinite program that generates a feedback controller $u(x)$ for every $x \in \mathscr{C}$,

while (5.21) only products a point-wise optimal controller.

To enforce the safety constraints, it is required that the barrier certified region is contained within the complement of the unsafe region, i.e., $\mathscr{C} \subseteq \mathscr{X}_u^c$. For generality, the unsafe region is encoded with multiple polynomial inequalities,

$$\mathscr{X}_u = \{x \in \mathscr{X} \mid q_i(x) < 0,\ \forall i \in \mathscr{M}\}, \tag{5.28}$$

where $q_i(x)$ are polynomials, and $\mathscr{M} = \{1, 2, ..., M\}$ is the index set of all the safety constraints.

Similar to *Lemma* 5.5.1, we can show that the region of safe stabilization estimated with barrier certificates is larger than the estimated region with Lyapunov sublevel set in [43].

Lemma 5.5.4. *Given a dynamical control system (5.23) that is locally stabilizable at the origin, the barrier certified region of safe stabilization estimate is larger than the estimated region of safe stabilization using sublevel set of the Lyapunov function, i.e,* $\mu(\mathscr{V}(c^*)) \leq \mu(\mathscr{C}^*)$.

Proof. Similar to Lemma 5.5.1. □

In order to maximize the volume of the safe operating region, the barrier certificate is rewritten into SMR form, i.e., $h(x) = Z(x)^T Q Z(x)$. Using the Real P-satz, the optimization problem (5.27) is formulated into a SOS program,

$$\begin{aligned}
&\max_{\substack{h(x)\in\mathscr{P},\ u(x)\in\mathscr{P}\\ L_1(x)\in\mathscr{P}^{\mathrm{SOS}},\ L_2(x)\in\mathscr{P}^{\mathrm{SOS}}\\ J_i(x)\in\mathscr{P}^{\mathrm{SOS}}, i\in\mathscr{M}}} && \mathrm{Trace}(Q)\\
\text{s.t.}\quad & -\frac{\partial V(x)}{\partial x}(f(x)+g(x)u(x)) - L_1(x)h(x) \in \mathscr{P}^{\mathrm{SOS}}, \\
& \frac{\partial h(x)}{\partial x}(f(x)+g(x)u(x)) + \gamma h(x) - L_2(x)h(x) \in \mathscr{P}^{\mathrm{SOS}}, \\
& -h(x) + J_i(x)q_i(x) \in \mathscr{P}^{\mathrm{SOS}}, \forall i \in \mathscr{M}.
\end{aligned} \tag{5.29}$$

The optimal barrier certificate obtained by solving the SOS program (5.29) is denoted by $h^*(x)$. The corresponding controller is $u^*(x)$. The following theorem shows that guaranteed safe stabilization can be achieved within the barrier certified region $\mathscr{C}^*$.

Theorem 5.5.5. *Given a dynamical control system (5.23) that is locally stabilizable at the origin, a Lyapunov function $V(x)$, an unsafe region $\mathscr{X}_u$ in (5.28), and the solution $h^*(x)$ to (5.29), for any initial state x_0 in $\mathscr{C}^* = \{x \in \mathscr{X} \mid h^*(x) \geq 0\}$, there always exists a controller that drives the system to the origin without violating safety constraints.*

Proof. Starting from any state $x_0 \in \mathscr{C}^*$, the state trajectory of the system (5.23) is denoted by $\psi(t;x_0)$ when the controller $u^*(x)$ from (5.29) is applied.

By Real P-satz, the second constraint in (5.29) implies that the barrier constraint in (5.27) is always satisfied, which ensures that the state trajectory $\psi(t;x_0)$ is always contained in $\mathscr{C}^*$. Similarly, the first constraint in (5.29) implies that $\frac{\mathrm{d}V(\psi(t;x_0))}{\mathrm{d}t}$ is always negative in $\mathscr{C}^*$ except at the origin. Thus $\lim_{t\to\infty} \psi(t;x_0) = 0$.

The third constraint in (5.29) ensures that "if $-q_i(x) > 0$, then $-h(x) > 0$". Consider the contrapositive of this statement, we have "if $h(x) \geq 0$, then $q_i(x) \geq 0$". This statement holds for any state $x \in \mathscr{C}^*$ and any safety constraint $i \in \mathscr{M}$, which means $\mathscr{C}^* \subseteq \mathscr{X}_u^c$. Because $\psi(t;x_0)$ is contained in $\mathscr{C}^*$, $\psi(t;x_0)$ is also contained in the safe space $\mathscr{X}_u^c$.

Combining these statements above, the controller $u^*(x)$ from (5.29) will drive any state in $\mathscr{C}^*$ to the origin without violating any safety constraint. □

Remark 4: With the generated permissive barrier certificates, it is guaranteed by construction that the QP-based controller (5.21) is always feasible when δ is set to zero. This is because $u^*(x)$ is always a feasible solution for any $x \in \mathscr{C}^*$. The advantage of using a QP-based controller (5.21) instead of $u^*(x)$ is that it minimizes the control effort by leveraging the part of nonlinear dynamics that contributes to stabilization.

The optimization problem (5.29) contains bilinear decision variables and requires a

feasible initial barrier certificate. It can be split into several SOS programs and solved with the following iterative search algorithm.

Algorithm 2:

Step 1: Calculate an initial guess for $h(x)$

Specify a Lyapunov function $V(x)$, and find c^* using bilinear search

$$
\begin{aligned}
c^* = \quad & \max_{\substack{c\in\mathbb{R}^+,\ u(x)\in\mathscr{P},\ L(x)\in\mathscr{P}^{\mathrm{SOS}} \\ J_i(x)\in\mathscr{P}^{\mathrm{SOS}},\ i\in\mathscr{M}}} c \\
\text{s.t.} \quad & -\frac{\partial V(x)}{\partial x}(f(x)+g(x)u(x)) - L(x)(c-V(x)) \in \mathscr{P}^{\mathrm{SOS}}, \\
& -(c-V(x)) + J_i(x)q_i(x) \in \mathscr{P}^{\mathrm{SOS}}, i\in\mathscr{M}.
\end{aligned}
$$

With the result of the bilinear search, set the initial guess for the barrier certificate as $\bar{h}(x) = c^* - V(x)$,

Step 2: Fix h(x), search for $u(x)$, $L_1(x)$, and $L_2(x)$

Using the $h(x)$ from previous step, we can search for feasible $u(x)$, $L_1(x)$, and $L_2(x)$, while maximizing the barrier constraint margin ε.

$$
\begin{aligned}
& \max_{\substack{\varepsilon\geq 0,\ u(x)\in\mathscr{P} \\ L_1(x)\in\mathscr{P}^{\mathrm{SOS}},\ L_2(x)\in\mathscr{P}^{\mathrm{SOS}}}} \varepsilon \\
\text{s.t.} \quad & -\frac{\partial V(x)}{\partial x}(f(x)+g(x)u(x)) - L_1(x)h(x) \in \mathscr{P}^{\mathrm{SOS}}, \\
& \frac{\partial h(x)}{\partial x}(f(x)+g(x)u(x)) + \gamma h(x) - L_2(x)h(x) - \varepsilon \in \mathscr{P}^{\mathrm{SOS}}.
\end{aligned}
$$

Step 3: Fix $u(x)$, $L_1(x)$, and $L_2(x)$, search for $h(x)$

Rewrite the barrier certificate into SMR form $h(x) = Z(x)^T QZ(x)$. With the $u(x)$, $L_1(x)$, and $L_2(x)$ from the previous step, we can search for the maximum volume barrier certificate

that respects all the safety constraints,

$$\begin{aligned} &\max_{\substack{h(x)\in\mathscr{P}\\ J_i(x)\in\mathscr{P}^{\mathrm{SOS}},\ i\in\mathscr{M}}} && \mathrm{trace}(Q) \\ \text{s.t.} \quad & -\frac{\partial V(x)}{\partial x}(f(x)+g(x)u(x))-L_1(x)h(x)\in\mathscr{P}^{\mathrm{SOS}}, \\ & \frac{\partial h(x)}{\partial x}(f(x)+g(x)u(x))+\gamma h(x)-L_2(x)h(x)\in\mathscr{P}^{\mathrm{SOS}}, \\ & -h(x)+J_i(x)q_i(x)\in\mathscr{P}^{\mathrm{SOS}}, i\in\mathscr{M}. \end{aligned}$$

Terminate if $\mathrm{trace}(Q)$ stops increasing, otherwise go back to *Step 2*.

Remark 5: In *Step 2*, the safety constraints $q_i(x)\geq 0, i\in\mathscr{M}$ do not need to be included. This is because $h(x)$ from previous step already satisfies these safety constraints.

Remark 6: To avoid unbounded control inputs, an additional constraint can be added to limit the magnitude of the coefficients of the polynomial controller $u(x)$.

This iterative search algorithm is implemented on two control dynamical systems to achieve safe stabilization.

Example 3: Consider the simple two-dimensional mechanical dynamical system,

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\end{bmatrix}=\begin{bmatrix}x_2\\ -x_1\end{bmatrix}+\begin{bmatrix}0\\ 1\end{bmatrix}u, \tag{5.30}$$

where $x=[x_1,x_2]^T\in\mathbb{R}^2$ and $u\in\mathbb{R}$ are the state and control of the system. A Lyapunov function $V(x)=x_1^2+x_1x_2+x_2^2$ can be picked for the system.

The unsafe area of the state space is encoded with multiple polynomial inequalities,

i.e., $\mathscr{X}_u = \{x \in \mathbb{R}^2 \mid q_i(x) < 0, i = 1,2,3\}$, where

$$q_1(x) = (x_1 - 3)^2 + (x_2 - 1)^2 - 1 < 0,$$
$$q_2(x) = (x_1 + 3)^2 + (x_2 + 4)^2 - 1 < 0,$$
$$q_3(x) = (x_1 + 4)^2 + (x_2 - 5)^2 - 1 < 0.$$

The largest estimate of the region of safe stabilization with sublevel set of V(x) can be obtained as

$$\mathscr{A}_1 = \{x \in \mathbb{R}^2 \mid V(x) \leq 5.8628\}.$$

With the barrier certificate, this estimate can be enlarged to

$$\mathscr{A}_2 = \{x \in \mathbb{R}^2 \mid h(x) = 0.5189 - 0.0669x_1 - 0.1196x_2 \\ -0.0546x_1^2 - 0.0630x_1x_2 - 0.0294x_2^2 \geq 0\}.$$

For comparison purpose, the barrier certificate is restricted to be second order polynomial. These estimates are illustrated in Fig. 5.9. By allowing the barrier certificate to be not centered around the equilibrium, the estimate of the region of safe stabilization is expanded significantly.

Example 4: Consider the three-dimensional system with multiple inputs,

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} x_2 - x_3^2 \\ x_3 - x_1^2 + u_1 \\ -x_1 - 2x_2 - x_3 + x_2^3 + u_2 \end{bmatrix}, \tag{5.31}$$

where $x = [x_1, x_2, x_3]^T \in \mathbb{R}^3$ and $u = [u_1, u_2]^T \in \mathbb{R}^2$ are the state and control of the system.

A Lyapunov function for the system is picked to be

$$V(x) = 5x_1^2 + 10x_1x_2 + 2x_1x_3 + 10x_2^2 + 6x_2x_3 + 4x_3^2.$$

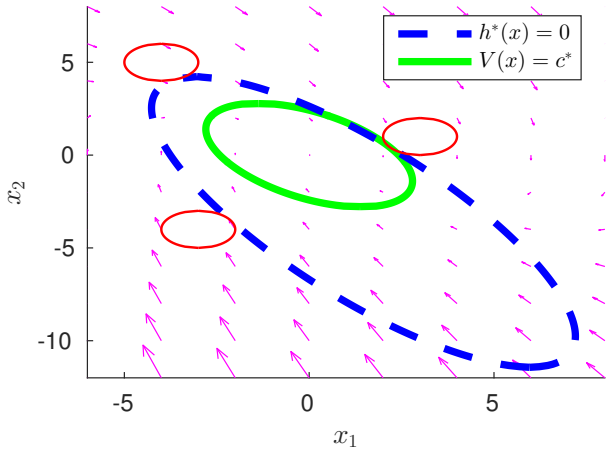


Figure 5.9: Region of safe stabilization estimates for system (5.30). The red circles represent unsafe regions. The magenta vector field represents the system dynamics when $u^*(x)$ is applied. The barrier certified region of safe stabilization (dashed blue ellipse) is significantly larger than the estimated region (solid green ellipse) with Lyapunov sublevel set based methods.

The unsafe region $\mathscr{X}_u = \{x \in \mathbb{R}^3 \mid q_i(x) < 0, i = 1,2,3,4\}$ is represented with polynomial inequalities

$$\begin{aligned} q_1(x) &= (x_1-2)^2 + (x_2-1)^2 + (x_3-2)^2 - 1 < 0, \\ q_2(x) &= (x_1+1)^2 + (x_2+2)^2 + (x_3+1)^2 - 1 < 0, \\ q_3(x) &= (x_1+0)^2 + (x_2-0)^2 + (x_3-6)^2 - 9 < 0, \\ q_4(x) &= (x_1+0)^2 + (x_2+0)^2 + (x_3+5)^2 - 9 < 0. \end{aligned}$$

The region of safe stabilization estimated with sublevel set of Lyapunov is

$$\mathscr{A}_1 = \{x \in \mathbb{R}^3 \mid V(x) \leq 13.0124\}.$$

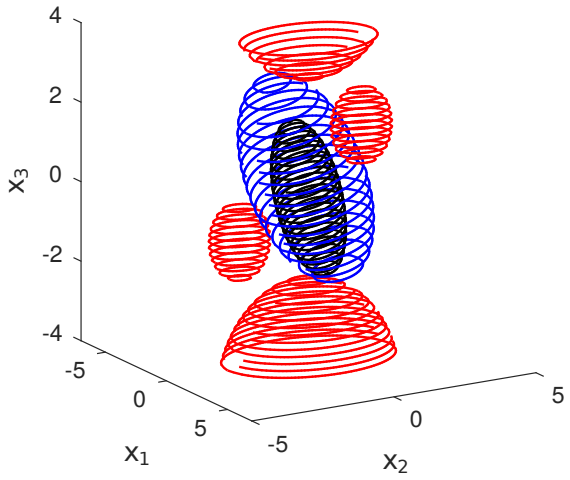


Figure 5.10: Region of safe stabilization estimates for system (5.31). The red spheres represent unsafe regions. The barrier certified region of safe stabilization (blue ellipsoid) is significantly larger than the region (black ellipsoid) obtained with Lyapunov sublevel sets.

Using the iterative search algorithm, the maximum permissive barrier certificate is

$$\begin{aligned}\mathscr{A}_2 = \{x \in \mathbb{R}^3 \mid h(x) = 114.3555 + 1.4686x_1 + 7.2121x_2 \\ +19.8479x_3 - 24.5412x_3^2 - 14.7734x_1^2 - 26.0129x_1x_2 \\ -15.5440x_1x_3 - 28.3492x_2^2 - 27.5651x_2x_3 \geq 0\}.\end{aligned}$$

The results for region of safe stabilization estimates are shown in Fig. 5.10. In both examples, the Lyapunov sublevel set search terminates as soon as the boundary of one safety constraint is reached, while the barrier certificate search terminates when all safety boundaries are touched. This also demonstrates the non-conservativeness of barrier certificates.

With the theoretical framework developed in this section, permissive barrier certified can be used to strictly ensure simultaneous stabilization and safety enforcement of dynamical systems. Iterative search algorithms using SOS programming techniques were designed to compute the most permissive barrier certificates. In addition, the proposed barrier certificates based method significantly expands the DoA estimate for both autonomous and control dynamical systems.

CHAPTER 6

SAFE LEARNING USING BARRIER CERTIFICATES

Machine learning based control approaches are becoming increasingly popular as a way to deal with inaccurate models, due to their abilities to infer unknown models from data and actively improve the performance of the controller with the learned model. However, the system is often subject to unsafe perturbations during the exploration phase. To extend this powerful tool to safety critical systems, explicit safety designs are required. In terms of the notion for safety, there are multiple methods proposed, e.g., Lyapunov stability and reachability based safety design. In this chapter, we will show that barrier certificates are good notions for the safety design of learning based controller [21].

As discussed in Section 3.1, barrier certificates expand the certified safe control space significantly by allowing $h(x)$ to decrease within $\mathscr{C}$ as opposed to strictly increasing [62, 2]. Compared with Lyapunov sublevel set based safe region, barrier certificates provide a more permissive notion of safety. As a result, barrier certificates based safe learning controllers have more freedom to efficiently explore those unknown states. This fact can be illustrated with the following example.

Example 1: Consider an autonomous dynamical system

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} x_2 + 0.8x_2^2 \\ -x_1 - x_2 + x_1^2 x_2 \end{bmatrix}, \tag{6.1}$$

the safe region of this system is estimated with both the Lyapunov sublevel set and barrier certificates.

Since (6.1) is a polynominal system, the safe sets can be computed directly with Sum-of-Squares programs using YALMIP [106] and SMRSOFT [42] solvers. Both the Lyapunov function and barrier certificates are limited to second order polynomials for fair

comparison. The safe region estimated with the optimal polynomial Lyapunov function is

$$\mathscr{A}_1 = \{x \mid V^*(x) \leq 1\},$$

where $V^*(x) = 1.343x_1^2 + 0.5155x_1x_2 + 1.152x_2^2$.

The safe region estimated with barrier certificates is

$$\mathscr{A}_2 = \{x \mid h^*(x) \geq 0\},$$

where $h^*(x) = 1 - 0.4254x_1 - 0.3248x_2 - 0.7549x_2^2 - 0.8616x_1^2 - 0.2846x_1x_2$.

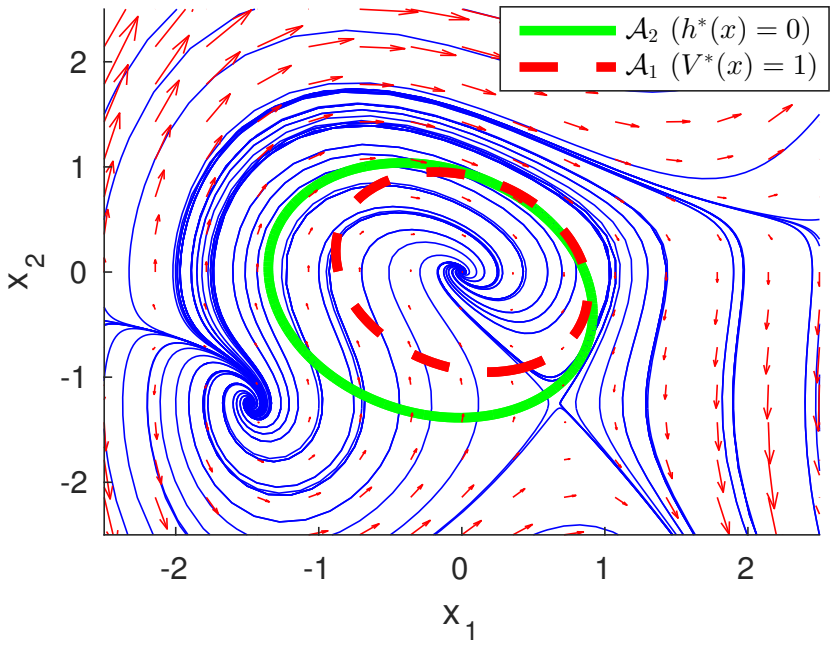


Figure 6.1: Estimates of safe regions for system (6.1). The regions enclosed by the dashed red ellipse and solid green ellipse are estimated safe regions with optimal polynomial Lyapunov function $V^*(x)$ and barrier certificates $h^*(x)$, respectively.

From Fig. 6.1, it can be observed that the barrier certified safe region $\mathscr{A}_2$ is much larger than the Lyapunov based safe region $\mathscr{A}_1$. Consequently, safe learning controller based on barrier certificates are allowed to explore more states of the system. In this Chapter, we will leverage the non-conservative safety guarantee of barrier certificates to allow a much

richer set of safe learning control options.

6.1 Learning Unknown Dynamics with Gaussian Process

6.1.1 Gaussian Processes

A GP is a nonparametric regression method that can capture complex unknown functions [107]. With a GP, every point in the state space is associated with a normally distributed random variable, which allows us to derive high probability statements about the system.

Adding some unknown dynamics $d(x)$ to the original class of control-affine systems (5.23), we now consider a system with partially unknown dynamics, i.e.,

$$\dot{x} = f(x) + g(x)u + d(x), \tag{6.2}$$

where $x \in \mathscr{X} \subseteq \mathbb{R}^n$ and $u \in \mathscr{U} \subseteq \mathbb{R}^m$ are the state and control of the system. Although the proposed method applies to general dynamical systems, here we restrict our attention to the class of systems that can be addressed with existing computation tools. It is also assumed that $d(x)$ is Lipschitz continuous. This assumption is necessary, because we want to generalize the learned dynamics to states that are not explored before.

Since the unmodeled dynamics $d(x)$ is n dimensional, each dimension is approximated with a GP model $\mathscr{GP}(0, k(x,x'))$ with a prior mean of zero and a covariance function of $k(x,x')$, where $k(x,x')$ is the kernel function to measure the similarity between any two states $x, x' \in \mathscr{X}$. In order to make GP inferences on the unknown dynamics, we need to get measurements of $d(x)$. This measurement $\hat{d}(x)$ is obtained indirectly by subtracting the inaccurate model prediction $[f(x) + g(x)u]$ from the noisy measurement of the system dynamics $[\dot{x} + \mathscr{N}(0, \sigma_n^2)]$. Since any finite number of data points form a multivariate normal distribution, we can obtain the posterior distribution of $d(x_*)$ at any query state $x_* \in \mathscr{X}$ by conditioning on the past measurements [107].

Given a collection of w measurements $y_w = [\hat{d}(x_1), \hat{d}(x_2), ..., \hat{d}(x_w)]^T$, the mean $m(x_*)$

and variance $\sigma^2(x_*)$ of $d(x_*)$ at the query state x_* are

$$m(x_*) = k_*^T(K+\sigma_n^2 I)^{-1}y_w, \tag{6.3}$$

$$\sigma^2(x_*) = k(x_*,x_*) - k_*^T(K+\sigma_n^2 I)^{-1}k_*, \tag{6.4}$$

where $\lfloor K \rfloor_{(i,j)} = k(x_i,x_j)$ is the kernel matrix, and $k_* = [k(x_1,x_*),k(x_2,x_*),...,k(x_w,x_*)]^T$.

With the learned system dynamics based on GP, a high probability confidence interval of the unmodeled dynamics $d(x)$ can be established as

$$\mathscr{D}(x) = \{d \mid m(x) - k_\delta \sigma(x) \leq d \leq m(x) + k_\delta \sigma(x)\}, \tag{6.5}$$

where k_δ is a design parameter to get $(1-\delta)$ confidence, $\delta \in (0,1)$. For instance, 95.5% and 99.7% confidence are achieved at $k_\delta = 2$ and $k_\delta = 3$, respectively.

6.1.2 Sparse Gaussian Process Methods

In this Chapter, the objective is to design safety strategy for online learning based control. One of the main practical limitation of Gaussian Process is that it is computationally expensive to implement, as the computation complexity is $O(n^2)$ for the kernel matrix inversion operation with respect to sample data size n. In Section 6.3.2, we will present an online recursive GP learning method. In addition, various computationally efficient approximation methods for GP are proposed in the literature.

These GP approximation methods reduce the kernel matrix dimension in various ways. One family of approximation method is based on a reduced set of inducing points called pseudo-inputs [108]. It is assumed that with finite amount of pseudo-inputs, the full kernel matrix can be well recovered. However, this method requires twice as many hyperparameters to optimize. In [109], eigenfunctions, which are known to be the most compact representation among all orthogonal basis functions, are used to approximate the unknown function. But EigenGP requires learning of both the eigenfunctions and hyperparameters.

Sparse Spectrum Gaussian Process (SSGP) are presented in [110] to approximate the unknown function with finite pairs of trigonometric basis functions. The SSGP method can approximate any stationary full GP with high prediction accuracy. To deal with real world robotics and engineering applications, input uncertainty can be incorporated into the SSGP model using analytic moment-based approaches with closed-form expressions [111]. The SSGP method and recursive GP learning method are adopted in the simulation and experiment in Section 6.4.

6.2 Safe Learning Using Barrier Certificates

In order to ensure that the learning based controller never enters the unsafe region, we will learn barrier certificates for the system and use the learned certificates to regulate the controller. As discussed in Section 3.1, the barrier certificates certify a safe region that is forward invariant. We can first start with an conservative barrier certificate with certificated safe region $\mathscr{C}_0(x)$, then gradually expand this certificated safe region with the collected data until it stops growing. This incremental learning process is visualized in Fig. 6.2.

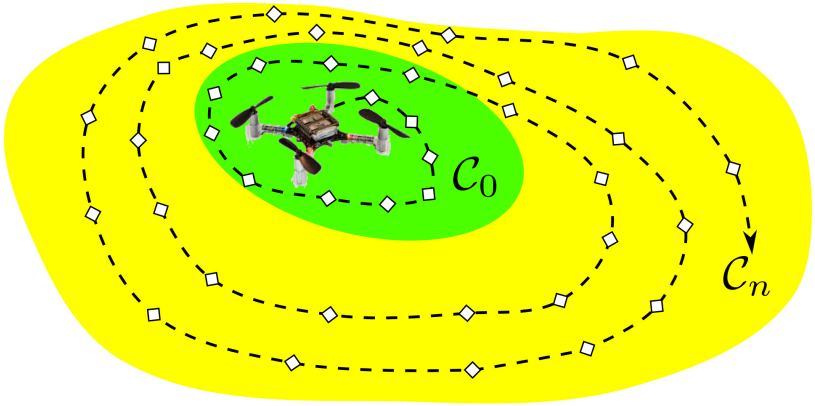


Figure 6.2: Incremental learning of the barrier certificates. The green region $\mathscr{C}_0$ and the yellow regions $\mathscr{C}_n$ are the initial and final barrier certified safe regions, respectively. The barrier certified safe region gradually grows as more and more data points are sampled in the state space.

More concretely, the goal of the learning process is to maximize the volume of the

barrier certified safe region $\mathscr{C}$ by adjusting $h(x)$, i.e.,

$$\begin{aligned} &\max_{h(x)} \quad \text{vol}(\mathscr{C}) \\ \text{s.t.} \quad &\max_{u\in\mathscr{U}} \min_{d\in\mathscr{D}(x)} \left\{ \frac{\partial h}{\partial x}(f(x)+g(x)u+d)+\gamma h(x) \right\} \geq 0, \\ &\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \forall x \in \mathscr{C}. \end{aligned}$$

Since u and d are independent from each other, we can rewrite this optimization problem into

$$\begin{aligned} &\max_{h(x)} \quad \text{vol}(\mathscr{C}) \\ \text{s.t.} \quad &\max_{u\in\mathscr{U}} \left\{ \frac{\partial h_k}{\partial x} g(x)u \right\} + \min_{d\in\mathscr{D}(x)} \left\{ \frac{\partial h}{\partial x} d \right\} \\ &\qquad\qquad + \frac{\partial h}{\partial x} f(x) + \gamma h(x) \geq 0, \forall x \in \mathscr{C} \end{aligned} \tag{6.6}$$

Using the high confidence interval $\mathscr{D}(x)$ in (6.5), the barrier certificates constraint can be considered as

$$\begin{aligned} &\max_{h(x)} \quad \text{vol}(\mathscr{C}) \\ \text{s.t.} \quad &\max_{u\in\mathscr{U}} \left\{ \frac{\partial h}{\partial x} g(x)u \right\} + \frac{\partial h}{\partial x} m(x) - k_\delta \left| \frac{\partial h}{\partial x} \right| \sigma(x) \\ &\qquad\qquad + \frac{\partial h}{\partial x} f(x) + \gamma h(x) \geq 0, \forall x \in \mathscr{C}. \end{aligned} \tag{6.7}$$

When more data points are collected about the system dynamics, the uncertainty $\sigma(x)$ will gradually decrease. As a result, more states will satisfy the barrier certificates constraint. The goal of the exploration task is to actively collect data to reduce $\sigma(x)$ and maximize the volume of $\mathscr{C}$.

It should be pointed out that the barrier certified region maximization problem (6.7) is a non-convex, infinite dimensional optimization problem, which is intractable to solve in practice. We will make two simplifications to make it solvable, namely by employing adaptive sampling of the state space and parameterization of the shape of $\mathscr{C}$.

6.2.1 Adaptive Sampling of the State Space

Due to the Lipschitz continuity of the system dynamics, the safety of the system in $\mathscr{X}$ can be evaluated by only sampling a finite number of points in $\mathscr{X}$. Inspired by [3], we will show that we can adaptively sample the state space without losing safety guarantees. Similar to *Lemma 4* in [3], it can be shown that $h(x)$ and $\dot{h}(x)$ are Lipschitz continuous in x with Lipschitz constants L_h and $L_{\dot{h}}$, respectively.

Let $\mathscr{X}_\tau \subset \mathscr{X}$ be a discretization of the state space $\mathscr{X}$. The closest point in $\mathscr{X}_\tau$ to $x \in \mathscr{X}$ is denoted as $[x]_\tau$, where $\|x - [x]_\tau\| \leq \frac{\tau}{2}$.

Lemma 6.2.1. *If the following condition holds for all $x \in \mathscr{X}_\tau$,*

$$\begin{aligned} \max_{u \in \mathscr{U}} \left\{ \frac{\partial h}{\partial x} g(x) u \right\} + \frac{\partial h}{\partial x} m(x) - k_\delta \left| \frac{\partial h}{\partial x} \right| \sigma(x) \\ + \frac{\partial h}{\partial x} f(x) + \gamma h(x) \geq (L_{\dot{h}} + \gamma L_h)\tau, \end{aligned} \tag{6.8}$$

then the safety barrier constraint

$$\max_{u \in \mathscr{U}} \min_{d \in \mathscr{D}(x)} \left\{ \frac{\partial h}{\partial x} (f(x) + g(x)u + d) + \gamma h(x) \right\} \geq 0 \tag{6.9}$$

is satisfied for all $x \in \mathscr{X}$ with probability $(1-\delta)$, $\delta \in (0,1)$.

Proof. With the definition of the high confidence interval $\mathscr{D}(x)$, (6.8) can be rewritten as

$$\max_{u \in \mathscr{U}} \min_{d \in \mathscr{D}(x)} \left\{ \frac{\partial h}{\partial x} (f(x) + g(x)u + d) + \gamma h(x) \right\} \geq (L_{\dot{h}} + \gamma L_h)\tau,$$

with a probability of $(1-\delta)$, for all $x \in \mathscr{X}_\tau$. This is equivalent to

$$\dot{h}(x) + \gamma h(x) \geq (L_{\dot{h}} + \gamma L_h)\tau,$$

for all $x \in \mathscr{X}_\tau$.

Because of the Lipschitz continuity of $h(x)$ and $\dot{h}(x)$, we have for any $x \in \mathscr{X}$,

$$\begin{aligned} \dot{h}(x) + \gamma h(x) &\geq (\dot{h}([x]_\tau) - L_{\dot{h}}\tau) + \gamma(h([x]_\tau) - L_h\tau) \\ &\geq 0. \end{aligned}$$

This means that the safety barrier constraint is satisfied for any $x \in \mathscr{X}$, if (6.8) holds for all $x \in \mathscr{X}_\tau$. □

With the discretization of the state space, we only need to sample a finite number of points to validate the barrier certificates. However, the number of required sampling points is still very large. The following adaptive sampling strategy further reduces the number of sampling points required.

Proposition 6.2.2. *If the following condition is satisfied at $x \in \mathscr{X}$,*

$$\begin{aligned} \max_{u \in \mathscr{U}} \left\{ \frac{\partial h}{\partial x} g(x) u \right\} + \frac{\partial h}{\partial x} m(x) - k_\delta \left| \frac{\partial h}{\partial x} \right| \sigma(x) \\ + \frac{\partial h}{\partial x} f(x) + \gamma h(x) \geq (L_{\dot{h}} + \gamma L_h) k_\tau \tau, \end{aligned} \tag{6.10}$$

with $k_\tau \geq 0$, then the safety barrier constraint (6.9) is satisfied for all $y \in \mathscr{X}$ such that $\|x - y\| \leq k_\tau \tau$.

Proof. The proof is similar to *lemma* 6.2.1. □

Leveraging the Lipschitz continuity of the barrier certificates, we can adaptively sample the state space without losing safety guarantees. Sparse sampling is performed at places with large safety margin, while dense sampling is only required at places with small safety margin.

6.2.2 Parameterization of the Barrier Certificates

Because maximizing the volume of $\mathscr{C}$ is a non-convex problem in general, we can parameterize the barrier certificate $h_\mu(x)$ with μ to simplify the optimization problem. For example, $h_\mu(x)$ can be formulated as $1 - Z(x)^T \mu Z(x)$, where $Z(x)$ is the vector of monomials, and μ is a positive semi-definite matrix. Then maximizing $\text{vol}(\mathscr{C})$ is equivalent to minimize the trace of μ. Further simplification can be made to fix the shape of $\mathscr{C}$ (by optimizing only with the known dynamics) and enlarge the level set of barrier certificates.

With the shape parameterization and adaptive sampling technique, the barrier certificate maximization problem (6.7) can be written as

$$\begin{aligned} &\max_{\mu} \quad \text{vol}(\mathscr{C}) \\ \text{s.t.} \quad &\max_{u \in \mathscr{U}} \left\{ \frac{\partial h_\mu}{\partial x} g(x) u \right\} + \frac{\partial h_\mu}{\partial x} m(x) - k_\delta \left| \frac{\partial h_\mu}{\partial x} \right| \sigma(x) \\ &\quad + \frac{\partial h_\mu}{\partial x} f(x) + \gamma h_\mu(x) \geq (L_{\dot{h}} + \gamma L_h)\tau, \forall x \in \mathscr{C} \cap \mathscr{X}_\tau. \end{aligned} \tag{6.11}$$

In order to increase the learning efficiency during the exploration phase, the most uncertain state in $\mathscr{C}$ is sampled,

$$x_{\text{next}} = \underset{x \in \mathscr{C} \cap \mathscr{X}_\tau}{\text{argmax}} \quad \sigma(x). \tag{6.12}$$

It is assumed that a nominal exploration controller $\hat{u}$ can always be designed to drive the system from the current state x to x_{next}, i.e., $\hat{u} = GoTo(x, x_{\text{next}})$. Then the safety barrier certificates are enforced through a QP-based controller to "rectify" the nominal control

such that the system is always safe,

$$
\begin{aligned}
u^* = \underset{u\in\mathscr{U}}{\operatorname{argmin}} \quad & J(u) = \|u-\hat{u}\|^2 \\
\text{s.t.} \quad & \frac{\partial h}{\partial x}g(x)u + \frac{\partial h}{\partial x}m(x) - k_\delta \left|\frac{\partial h}{\partial x}\right| \sigma(x) \\
& \qquad + \frac{\partial h}{\partial x}f(x) + \gamma h(x) \geq 0.
\end{aligned} \tag{6.13}
$$

Therefore, the actual exploration controller u^* tries to stay as close as possible to the desired controller $\hat{u}$, while always honoring the safety requirements. The exploration phase ends when the safe region $\mathscr{C}$ does not grow any more. The learned maximum barrier certificates can be further used to regulate other control tasks the system want to achieve.

6.2.3 Overview of the Safe Learning Algorithm

An overview of the barrier certificates based safe learning algorithm is provided in **Algorithm 2** in the Appendix. At the beginning, a conservative barrier certified safe region $\mathscr{C}_0$ is provided. The most uncertain state x_{next} is computed based on the current GP model. Then, the QP based controller (6.13) is used to ensure that the system is driven to x_{next} without ever leaving $\mathscr{C}_n$. After updating the GP model with the sampled data at x_{next}, the barrier certificate optimization problem (6.11) is solved. The adaptive sampling technique (6.10) is adopted here to reduce the number of states to be sampled. This process is repeated until the safe region $\mathscr{C}_n$ stops growing.

6.3 Learning Based Control for Quadrotor System

The safe learning approach developed in Section 6.2 relies on a learning controller that drives the system to explore interested states. The challenge of designing this learning controller is that the 3D quadrotor system considered in this Chapter is highly nonlinear and unstable. In this section, we will present a recursive learning controller based on GP to learn the complex quadrotor dynamics online.

6.3.1 Differential Flatness of 3D Quadrotor Dynamics

The quadrotor is a well-modelled dynamical system with forces and torques generated by four propellers and gravity [82]. The relevant coordinate frames and Euler angles (roll ϕ, pitch θ, and yaw ψ) are illustrated in Fig. 6.3. The world, body, and intermediate frames (after yaw angle rotation) are denoted by the subscripts w, b, and c, respectively.

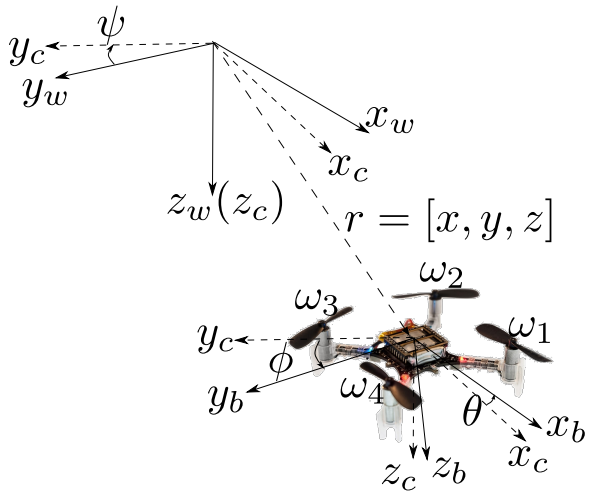


Figure 6.3: Quadrotor coordinate frames.

The Euler angles are defined with the ZYX convention. Hence, the rotation matrix from the body frame to the world frame can be written as

$$R = \begin{bmatrix} c\theta c\psi & s\phi s\theta c\psi - c\phi s\psi & c\phi s\theta c\psi + s\phi s\psi \\ c\theta s\psi & s\phi s\theta s\psi + c\phi c\psi & c\phi s\theta s\psi - s\phi c\psi \\ -s\theta & s\phi c\theta & c\phi c\theta \end{bmatrix},$$

where $s\theta$ and $c\theta$ stand for $\sin\theta$ and $\cos\theta$, respectively.

Here, we adopt the quadrotor model used in [5] to describe the nonlinear quadrotor

dynamics,

$$\begin{cases} \ddot{r} & = & g z_w + \frac{1}{m} R z_w f_z, \\ \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} & = & \begin{bmatrix} 1 & s\phi t\theta & c\phi t\theta \\ 0 & c\phi & -s\phi \\ 0 & s\phi sc\theta & c\phi sc\theta \end{bmatrix} \omega, \end{cases} \tag{6.14}$$

where $z_w = [0\ 0\ 1]^T$, and $r = [x, y, z]^T$, m, and g are the position of the center of mass, the mass, and the gravitational acceleration of the quadrotor, respectively. $t\theta$ and $sc\theta$ are short for $\tan\theta$ and $\sec\theta$. The control inputs of the quadrotor are the body rotational rates ($\omega = [\omega_x, \omega_y, \omega_z]^T$) and the thrust ($f$). It is assumed that the body rotational rates of quadrotor are directly controllable through the fast response onboard controller, due to the small rotational inertia and high torque features of quadrotors [5].

Similar to [82], the dynamics in (6.14) is differentially flat with the flat output chosen as $\eta = [r^T, \psi^T]^T$. The full state $q = [r^T, \dot{r}^T, \theta, \phi, \psi]^T$ and control $u = [f, \omega^T]^T$ can be represented as an algebraic function of $[\eta^T, \dot{\eta}^T, \ddot{\eta}^T, \dddot{\eta}^T]$. With the differential flatness property, quadrotor trajectory planning can be simplified as smooth parametric curves. Given a desired trajectory $\eta_d(t) \in C^3$ that is three times differentiable, the feed forward control $u_{FF} = [f_{FF}, \omega_{FF}^T]$ can be derived by inverting the dynamics in (6.14),

$$\begin{cases} f_{FF} & = & -m\|\ddot{r}_d - g z_w\|, \\ \omega_{FF} & = & \begin{bmatrix} 1 & 0 & -s\theta_d \\ 0 & c\phi_d & s\phi_d c\theta_d \\ 0 & -s\phi_d & c\phi_d c\theta_d \end{bmatrix} \begin{bmatrix} \dot{\phi}_d \\ \dot{\theta}_d \\ \dot{\psi}_d \end{bmatrix} \end{cases}$$

where $\theta_d = \text{atan2}(\beta_a, \beta_b)$, $\phi_d = \text{atan2}(\beta_c, \sqrt{\beta_a^2 + \beta_b^2})$, $\beta_a = -\ddot{x}_d \cos\psi_d - \ddot{y}_d \sin\psi_d$, $\beta_b = -\ddot{z}_d + g$, and $\beta_c = -\ddot{x}_d \sin\psi_d + \ddot{y}_d \cos\psi_d$.

Differential flatness only gives the feed forward control u_{FF}. In addition, the unknown

model error and tracking error need to be handled by a feedback control u_{FB}. The actual control applied to the quadrotor is $u = u_{FF} + u_{FB}$, where

$$\begin{cases} f_{FB} &= K_p < Rz_w, r_d - r > + K_d < Rz_w, \dot{r}_d - \dot{r} >, \\ \omega_{FB} &= K_p \begin{bmatrix} \phi_d - \phi \\ \theta_d - \theta \\ \psi_d - \psi \end{bmatrix} + K_d \begin{bmatrix} \dot{\phi}_d - \dot{\phi} \\ \dot{\theta}_d - \dot{\theta} \\ \dot{\psi}_d - \dot{\psi} \end{bmatrix} + \bar{K}_p \begin{bmatrix} y_d - y \\ x - x_d \\ 0 \end{bmatrix} \end{cases}$$

Note that with an inaccurate model, a high-gain feedback controller is needed to counteract both the model error and disturbances. As a better model is learned over time, only a low-gain feedback controller is needed with an improved tracking performance [112].

The previous section deals with precise quadrotor models. But it is often difficult to acquire accurate parameters for quadrotor systems. In addition, the model (6.14) neglects the uncertain effects of damping, drag force, and wind disturbances. Here, we will use GP models to learn the unmodeled dynamics. The unmodeled dynamics can be captured with six GPs along each dimension in the state space, i.e.,

$$\begin{cases} \ddot{r} &= g z_w + \frac{1}{m} R z_w f_z + \begin{bmatrix} \mathscr{GP}_1(0, k(q,q')) \\ \mathscr{GP}_2(0, k(q,q')) \\ \mathscr{GP}_3(0, k(q,q')) \end{bmatrix}, \\ \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} &= \begin{bmatrix} 1 & s\phi t\theta & c\phi t\theta \\ 0 & c\phi & -s\phi \\ 0 & s\phi sc\theta & c\phi sc\theta \end{bmatrix} \omega + \begin{bmatrix} \mathscr{GP}_4(0, k(q,q')) \\ \mathscr{GP}_5(0, k(q,q')) \\ \mathscr{GP}_6(0, k(q,q')) \end{bmatrix}, \end{cases}$$

where the input to the GPs is $q = [r^T, \dot{r}^T, \theta, \phi, \psi]^T$, and the observations for the GPs are $s = [\ddot{r}^T, \dot{\phi}, \dot{\theta}, \dot{\psi}]^T$, respectively. At a new query point q_*, the mean $m_i(q_*)$ and variance $\sigma_i^2(q_*)$ of the unknown dynamics can be inferred with (6.3). Based on the learned dynamics, a

differential flatness based feed forward controller can be derived as,

$$\begin{cases} f_{FF} & = & -m\|\ddot{r}_d - [m_1(q), m_2(q), m_3(q)]^T - gz_w\|, \\ \omega_{FF} & = & \begin{bmatrix} 1 & 0 & -s\theta_d \\ 0 & c\phi_d & s\phi_d c\theta_d \\ 0 & -s\phi_d & c\phi_d c\theta_d \end{bmatrix} \begin{bmatrix} \dot{\phi}_d - m_4(q) \\ \dot{\theta}_d - m_5(q) \\ \dot{\psi}_d - m_6(q) \end{bmatrix}, \end{cases}$$

where $\theta_d = \text{atan2}(\bar{\beta}_a, \bar{\beta}_b)$, $\phi_d = \text{atan2}(\bar{\beta}_c, \sqrt{\bar{\beta}_a^2 + \bar{\beta}_b^2})$, $\bar{\beta}_a = -(\ddot{x}_d - m_1(q))\cos\psi_d - (\ddot{y}_d - m_2(q))\sin\psi_d$, $\bar{\beta}_b = -(\ddot{z}_d - m_3(q)) + g$, and $\bar{\beta}_c = -(\ddot{x}_d - m_1(q))\sin\psi_d + (\ddot{y}_d - m_2(q))\cos\psi_d$.

6.3.2 Recursive Online GP Learning

One issue with the GP regression is that the time complexity of GP inference is $O(N^3)$, where N is the number of data points. The majority of the time is used to compute the inverse of the kernel matrix K. While various approximation methods can be used to reduce the GP inference time, it is still challenging to perform online GP inference for complex dynamically systems like quadrotor. Here, we propose a recursive online GP Learning method to compute the exact GP inference.

As the quadrotor moves forward, we will actively add multiple relevant data points into the kernel matrix at each time step. At the same time, the data points that contribute the least to the inference are deleted. The recursive data addition and deletion operations are described as following.

Adding Multiple New Data to the Kernel Matrix

Let the kernel matrix at the ith time step be K_i, we can save the matrix inverse result from the previous step as $L_i = (K_i + \sigma_n^2 I)^{-1}$. Denote the number of new data to be added as M.

With the new data y_{i+1} and kernal vector k_{i+1}, we have

$$\begin{aligned} L_{i+1} &= \begin{bmatrix} L_i^{-1} & k_{i+1} \\ k_{i+1}^T & c_{i+1}+\sigma_n^2 I \end{bmatrix}^{-1} \\ &= \begin{bmatrix} L_i + L_i k_{i+1}(c_{i+1}+\sigma_n^2 I - k_{i+1}^T L_i k_{i+1})^{-1} k_{i+1}^T L_i & \\ \quad -(c_{i+1}+\sigma_n^2 I - k_{i+1}^T L_i k_{i+1})^{-1} k_{i+1}^T L_i & \\ & L_i k_{i+1}(c_{i+1}+\sigma_n^2 I - k_{i+1}^T L_i k_{i+1})^{-1} \\ & (c_{i+1}+\sigma_n^2 I - k_{i+1}^T L_i k_{i+1}) \end{bmatrix}. \end{aligned}$$

Notice that inversion operation only needs to be performed on a $M \times M$ matrix rather than a large $N \times N$ matrix.

Deleting Multiple Old Data from the Kernel Matrix

After deleting M data points from the old Kernel matrix inversion $L_i = (K_i + \sigma_n^2 I)^{-1}$, the new inverse of the kernel matrix becomes $\bar{L}_i = (\bar{K}_i + \sigma_n^2 I)^{-1}$.

First, the data to be deleted is permuted to the bottom of the kernel matrix with a permutation matrix P_π, where $\pi : \mathbb{N} \to \mathbb{N}$ is a permutation of N elements. The permuted kernel matrix is $K_i^p = P_\pi K_i P_\pi^T$, which can be written into a block matrix form,

$$K_i^P = \begin{bmatrix} \bar{K}_i & E_i \\ E_i^T & F_i \end{bmatrix},$$

where E_i, F_i are the known parts to be deleted. Similarly,

$$\begin{aligned} L_i^P &= P_\pi L_i P_\pi^T \\ &= \begin{bmatrix} \bar{L}_i^{-1} & E_i \\ E_i^T & F_i + \sigma_n^2 I \end{bmatrix}^{-1}. \end{aligned}$$

Since L_i^P is known, it can be written into block matrix form with the same block dimensions with (6.3.2),

$$L_i^P = \begin{bmatrix} A_i & B_i \\ B_i^T & C_i \end{bmatrix}.$$

With the block matrix inversion rule, $\bar{L}_i$ can be recovered as

$$\bar{L}_i = A_i - B_i C_i^{-1} B_i^T,$$

which means to perform the deletion operation, the only matrix inverse required is $C_i^{-1} \in \mathbb{R}^{M\times M}$.

With the recursive data addition and deletion method, the GP inference can be obtained efficiently online.

6.4 Simulation and Experiment Results

The GP based learning algorithm is validated on a simulated quadrotor model as well as an actual palm-sized quadrotor (Crazyflie 2.0). In the simulation, the actual weight of the quadrotor is 1.4 times the weight used in the computation. In addition, an unknown constant wind of $0.1g$ is applied in the environment as illustrated in Fig. 6.4. Since the standard fixed pitch quadrotor cannot generate reverse thrust, the thrust control is limited to $f_z \in [-1.8mg, 0]$. This simulation setup is very challenging, because the learning based quadrotor controller needs to deal with very inaccurate model and limited thrust.

During the actual experiment, the quadrotor is commanded to fly through a Dyson bladeless fan as shown in Fig. 6.5. Since the actuators of the palm-sized quadrotor are relatively weak compared with the wind force, it is very challenging to counteract the wind disturbance and follow desired trajectories.

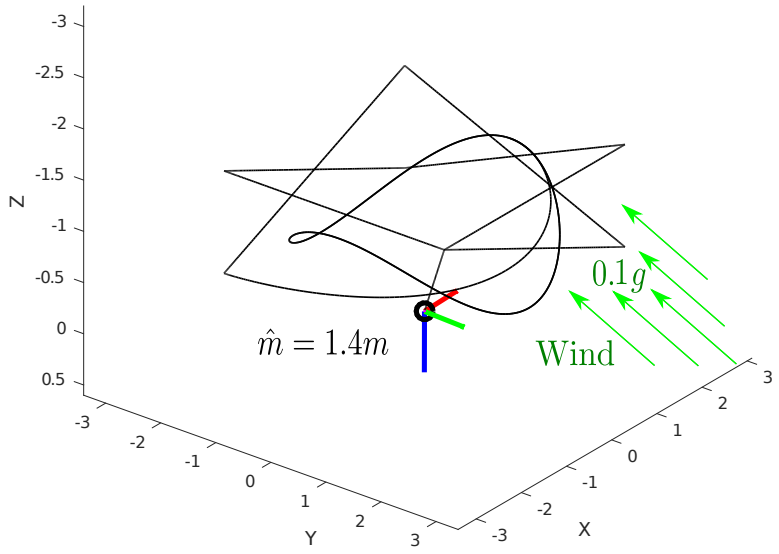


Figure 6.4: A simulated quadrotor flies in an unknown wind field with an inaccurate model.

6.4.1 Online Learning of Quadrotor Dynamics

In the first example, the quadrotor is commanded to track a nominal trajectory (illustrated in Fig. 6.4) using a differential flatness based controller with the given inaccurate model. A PD controller is wrapped around to stabilize the quadrotor. During the simulation, the quadrotor is intentionally pushed to unknown regions that has not been explored before. This will help us evaluate the scalability of the algorithm.

The desired trajectory of the quadrotor is given as $\hat{\eta} = [\hat{r}(t)^T, \hat{\psi}(t)] \in C^3$, while the actual trajectory is $\eta = [r(t)^T, \psi(t)]$. In practice, the actual trajectory might deviate significantly from the desired trajectory when the model is very inaccurate. To track the desired trajectory, the nominal trajectory is designed with a pole placement controller,

$$\ddddot{r}_i = \ddddot{\hat{r}}_i - K \cdot [(r_i - \hat{r}_i),\ (\dot{r}_i - \dot{\hat{r}}_i),\ (\ddot{r}_i - \ddot{\hat{r}}_i)]^T.$$

In the simulation, the sample size of the recursive GP model is fixed at 300 data points. At each time step, the most irrelevant data point is thrown away, and the most relevant data point is added to the GP model. The data relevance is decided by the kernel function

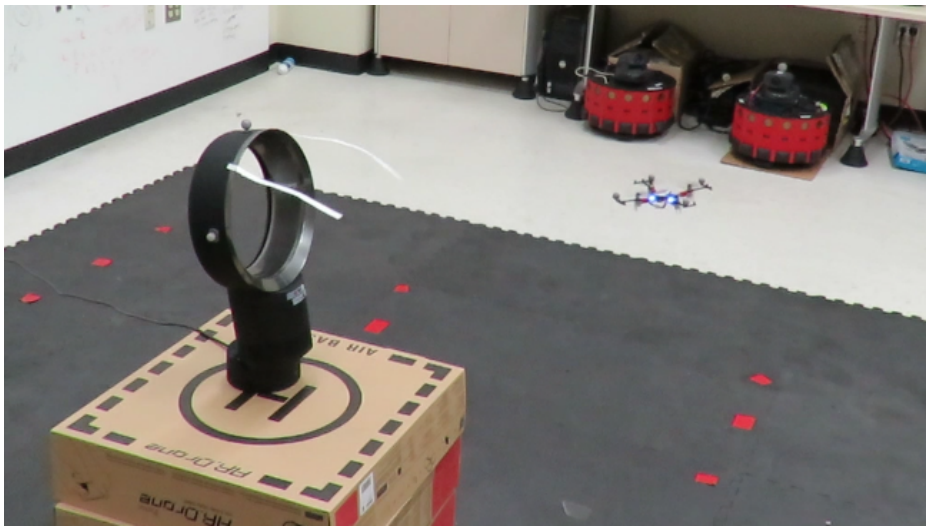
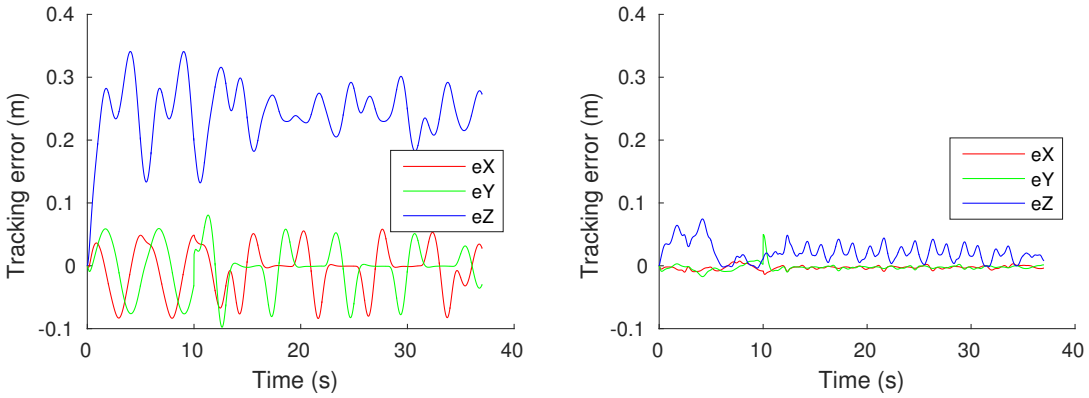


Figure 6.5: A plam-sized quadrotor, Crazyflie, flew through a Dyson fan and hovered in the wind field with the learning based controller.

$k(q,q^*)$, where $q = [r^T, \dot{r}^T, \theta, \phi, \psi]^T$. It can observed that the tracking error of the learning based controller is significantly smaller than the tracking error without GP inference, as shown in Fig. 6.6.



(a) Tracking error without GP

(b) Tracking error with GP

Figure 6.6: Tracking error of the differential flatness based flight controller with and without GP inference.

With the recursive learning strategy, it is demonstrated in Fig. 6.7 that the GP inference time is always kept below $20ms$. Thus, the recursive GP inference method is very suitable for online learning of quadrotor dynamics.

By pushing the quadrotor to unexplored regions, we can found that learning with $q' =$

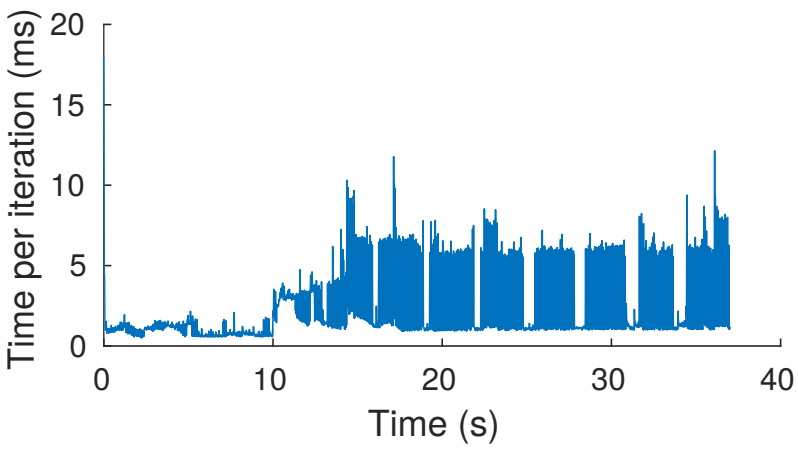


Figure 6.7: Recursive GP inference time per iteration.

$[\dot{r}^T, \theta, \phi, \psi]^T$ yields much better scalability than learning with $q = [r^T, \dot{r}^T, \theta, \phi, \psi]^T$. The reason might be the position r is not as important as other features in the current simulation setup.

6.4.2 Learning Unknown Dynamics from Real Flight Data

In this experiment, a palm-size quadrotor (Crazyflie 2.0) is commanded to fly through the center of a Dyson bladeless fan as shown in Fig. 6.5. The flight trajectory of quadrotor is planned using spline interpolation. Then a learning based controller is used to track this reference trajectory. To make sure that the quadrotor does not collide with the fan, a safety corridor constraint is added to the path planning problem to constrain the motion of the quadrotor as illustrated in Fig. 6.8.

As the quadrotor is flying in the wind field created by the Dyson fan, the unmodeled dynamics is learned with Gaussian Process. Since the learning based controller needs updates in real time, a sparse spectrum GP prediction method [111] is adopted to make online updates. As shown in Fig. 6.9, the sparse spectrum GP method produces accurate predictions similar to the full GP model. In addition, the undesired noise in the flight data are smoothed out by the GP model automatically. Because the state is multi-dimensional with different importance, the Automatic Relevance Determination (ARD) method [113] is used

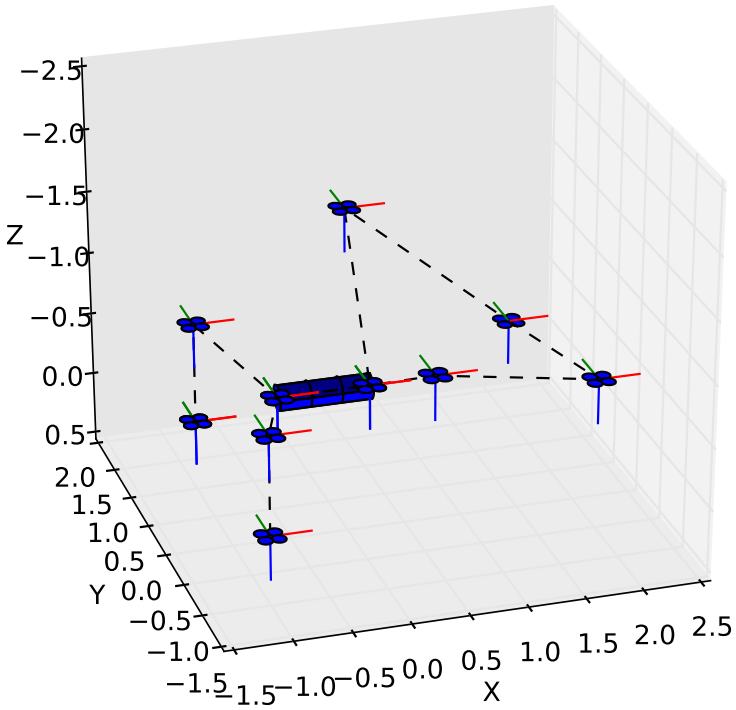


Figure 6.8: Planned flight trajectory for the quadrotor to flying through a Dyson fan. The blue meshed tube is placed at the center of the fan.

to perform automatic feature selection.

With the learning based controller, the quadrotor successfully flew through the Dyson fan and hovered in the unknown wind field without crashing as shown in Fig. 6.5. The video of the experiment can be viewed at [74].

6.4.3 Learning Safety Barrier Certificates

In this example, the motion of the quadrotor is constrained within an ellipsoid safe region, i.e.,

$$\frac{x^2}{0.16}+\frac{y^2}{0.16}+\frac{(z+0.8)^2}{0.36}\leq 1.$$

The quadrotor is controlled to fly back and forth on a vertical path inside the ellipsoid. The goal is to learn how aggressively the quadrotor can fly in the z direction with an inaccurate

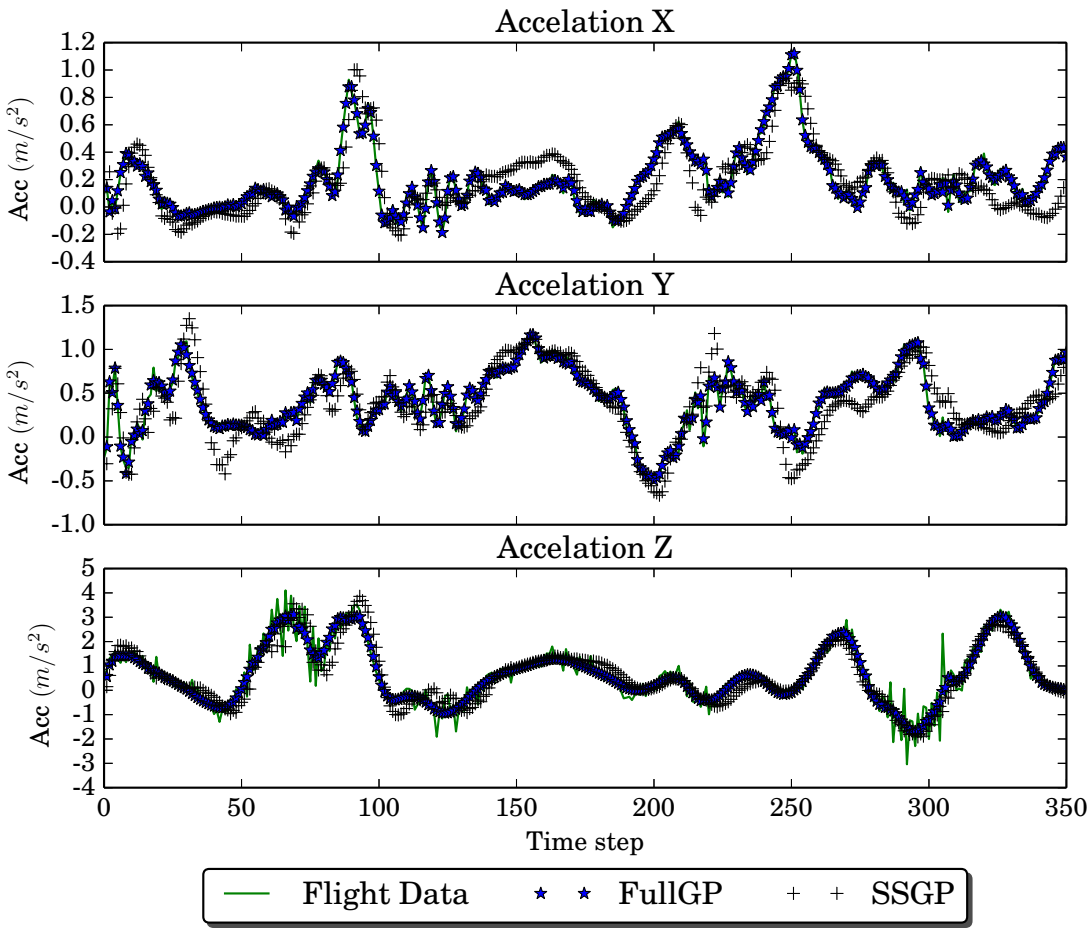


Figure 6.9: Unknown dynamics learned using Gaussian Process from actual flight data. The predicted dynamics using full GP and sparse spectrum GP are compared against the real flight data. The SSGP prediction can run in real time with similar accuracy with full GP model.

model and limited thrust.

The barrier certificates are parameterized as

$$\begin{aligned} h_\mu(r) &= 1 - \frac{(z+0.8)^2}{0.36} - \mu\dot{z}^2 \\ &\quad - \frac{x^2}{0.16} - \frac{y^2}{0.16} - \frac{\dot{x}^2}{0.25} - \frac{\dot{y}^2}{0.25} \geq 0, \end{aligned}$$

where μ is the barrier parameter to regulate how fast the quadrotor can fly in the z direction. Small values of μ correspond to large admissible speed $\dot{z}$, which means more aggressive flight behavior. Thus, the objective of the learning process is to minimize μ with the col-

lected data.

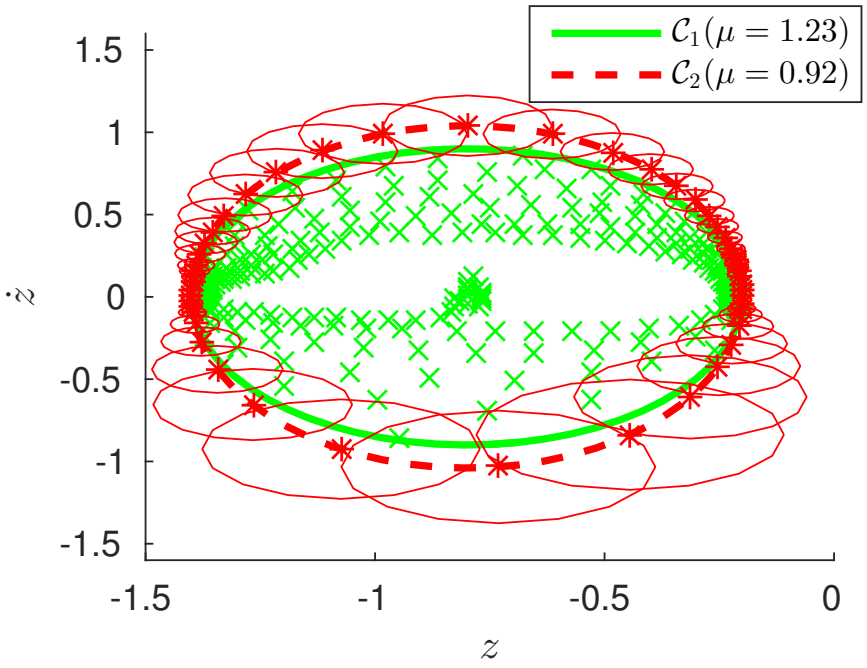


Figure 6.10: Adaptive sampling of the state space. The region enclosed by the solid green ellipse $\mathscr{C}_1$ is the current safe region, while the region enclosed by the dashed red ellipse $\mathscr{C}_2$ is the optimized next safe region. The green cross markers and red asterisk markers are the data points already sampled and to be sampled, respectively. The red circles centered at those sample points are the confident safe regions. All the unexplored region between $\mathscr{C}_1$ and $\mathscr{C}_2$ are covered by the circular confident safe region.

To reduce the number of required sample points, the adaptive sampling strategy developed in Section 6.2.1 was adopted. An illustrative example of the adaptive sampling strategy is given in Fig. 6.10. It can be observed that places closer to the boundary of the safe region ($z = -1.2$ and $z = -0.2$) are sampled much denser than the place closer to the center of the safe region ($z = 0$). Furthermore, downward speed ($\dot{z} > 0$) is sampled much denser than the upward speed ($\dot{z} < 0$). This might be caused by the lack of reverse thrust to counter the unmodeled dynamics.

A conservative barrier certificate ($\mu = 6.3$) is provided at the beginning of the learning process. Then, the quadrotor gradually explores the safe region $\mathscr{C}_0$ and expands it to $\mathscr{C}_n$ ($\mu = 0.6$), as illustrated in Fig. 6.11. The nominal exploration controller is always regulated

by the barrier certificates using the QP-based controller in (6.13). During the learning process, the quadrotor never leaves the barrier certified safe region.

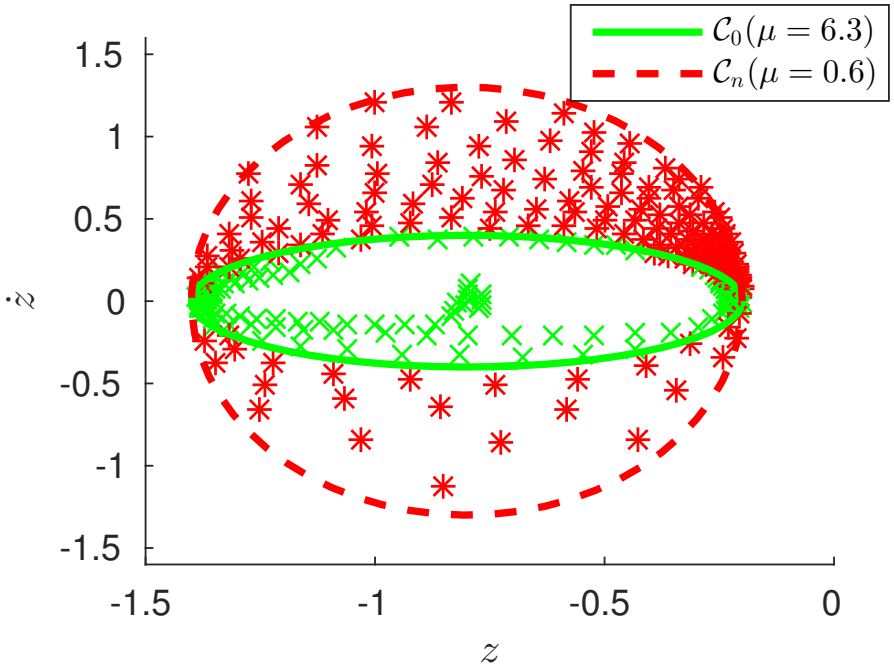


Figure 6.11: Initial and final barrier certificates. The regions enclosed by the solid green ellipse ($\mathscr{C}_0$) and dashed red ellipse ($\mathscr{C}_n$) are the initial and final barrier certified safe regions, respectively. The green cross markers and red asterisk makers are the sampled data points.

In this chapter, a high confidence safe learning algorithm based on barrier certificates was presented to explicitly address the safety challenge in learning based control. The learning controller is regulated by the barrier certificates, such that the system never enters the unsafe region. The unmodel dynamics of the system was approximated with a Gaussian Process, from which a high probability safety guarantee for the dynamical system was derived. The barrier certified safe region is gradually expanded as the uncertainty of the system dynamics is reduced with more data. This safe learning technique was validated on a quadrotor system with 3D nonlinear dynamics.

CHAPTER 7
CONCLUSIONS AND FUTURE WORKS

A formal safety framework for multi-robot coordination and learning-based control was developed in this dissertation. This framework provides the target system with provable safety guarantee using barrier certificates, which cause minimal modifications to its higher level objective. The barrier certificates based safety algorithms are designed in a computationally efficient way such that real-time applications on large scale systems are possible.

For the purpose of safe multi-robot coordination, a general framework of minimally invasive collision avoidance for multi-robot systems was formally synthesized using control barrier functions. The computation and sensing requirements were reduced significantly by distributing safety barrier certificates to each individual agents and only considering neighboring agents without losing the safety guarantee. Then a series of problems related to safety barrier certificates, i.e., the conservativeness of the certificates, the feasibility of the QP-based controller and deadlock-avoidance, were addressed. The proposed safety barrier certificates were validated through various simulations, and then implemented on real multi-robot systems consisting of multiple Khepera robots, Magellan Pro robot, GRITSBots, and Crazyflie quadrotors.

As teams of robots often need to deal with multiple objectives simultaneously, a systematic way to compose multiple objectives using the compositional barrier functions was presented. AND and OR logical operators were designed to provably compose multiple non-negotiable objectives, with conditions for composibility provided. The composite safety and connectivity barrier certificates were synthesized using the compositional barrier functions to formally ensure safety and connectivity for teams of mobile robots. The resulting barrier certificates were then combined with the general higher level objectives using an optimization-based controller. Robotic experimental implementations validated

the effectiveness of the proposed method. When safety and stabilization requirements are both mandatory, a theoretical framework to generate permissive barrier certified region of safe stabilization was developed to strictly ensure simultaneous stabilization and safety enforcement of dynamical systems. Iterative search algorithms using SOS programming techniques were designed to compute the most permissive barrier certificates. In addition, the proposed barrier certificates based method significantly expands the DoA estimate for both autonomous and control dynamical systems. The effectiveness of the iterative search algorithm was demonstrated with simulation results.

In terms of learning based control, a safe learning algorithm based on barrier certificates was developed in this dissertation. The learning based controller is regulated by the barrier certificates, such that the system never enters the unsafe region. The unmodeled dynamics of the system was approximated with Gaussian Processes, from which a high probability safety guarantee for the dynamical system was derived. The barrier certified safe region is gradually expanded as the uncertainty of the system dynamics is reduced with more data. This safe learning technique was applied on a quadrotor system with 3D nonlinear dynamics. The computation time of this learning method is reduced significantly with an adaptive sampling strategy and sparse Gaussian Process inference method. Simulation and experimental results demonstrated the effectiveness of the proposed method.

A formal safety framework for multi-robot coordination and learning based control was developed in this dissertation using barrier certificates. It also gives rise to several interesting future directions. The feasibility of the barrier certificates for teams of robots was established without considering actuation limits. In addition, the actuation limits were addressed in a heuristic way by parameterizing the nominal trajectories for the robots to follow. A more systematic way to synthesize the barrier certificates is to incorporate the actuation limits directly at the design stage.

The permissive barrier certificates were presented to deal with multiple objectives for control dynamical systems using Sum-of-Squares programming. However, when it comes

to large multi-agent networks, this method requires solving a large centralized semi-definite programming problem, which is very computationally expensive. Future works might be devoted to distribute the permissive barrier certificates computation problem to each individual agents.

The safe learning approach presented in this dissertation deals with stationary dynamical systems currently. Since the environment that the robot interact with might be non-stationary, it would be beneficial to modify the safety framework in an adaptive manner. To work with high dimensional and complex dynamical systems, it is desirable to further reduce the computational complexity of the safe learning algorithm without compromising the performance of the safe learning algorithm.

Appendices

APPENDIX A

PROOF OF THEOREMS

A.1 Proof of Theorem 3.2.2

Proof. Consider any agent k that is not a neighbor of agent i, i.e., $D_{ik} = \|\Delta\mathbf{p}_{ik}\| > D^i_{\mathcal{N}}$. We will prove that agent k is guaranteed to satisfy the pairwise safety barrier constraint with agent i no matter what control action is taken.

Since $\dot{D}_{ik} = \|\Delta\dot{\mathbf{p}}_{ik}\| = \frac{\Delta\mathbf{p}_{ik}^T}{\|\Delta\mathbf{p}_{ik}\|}\Delta\mathbf{v}_{ik}$, h_{ik} in (3.4) can be reformulated in terms of D_{ik} and $\dot{D}_{ik}$,

$$h_{ik} = \dot{D}_{ik} + \sqrt{2(\alpha_i + \alpha_k)(D_{ik} - D_s)}.$$

The derivative of h_{ik} is given by

$$\dot{h}_{ik} = \ddot{D}_{ik} + \sqrt{\frac{\alpha_i + \alpha_k}{2(D_{ik} - D_s)}}\dot{D}_{ik}.$$

With the velocity and acceleration limits of both agents, the lower bounds of h_{ik} and $\dot{h}_{ik}$ can be derived by considering the worst case scenario ($\ddot{D}_{ik} = -\alpha_i - \alpha_k$, $\dot{D}_{ik} = -\beta_i - \beta_k$). Since agent k can be any agent in the multi-robot system, these lower bounds can be further relaxed with the bounds on all agents' acceleration and speed limits.

$$\begin{aligned}
h_{ik} &\geq \sqrt{2(\alpha_i + \alpha_k)(D_{ik} - D_s)} - \beta_i - \beta_k \\
&\geq \sqrt{2(\alpha_i + \alpha_{\min})(D_{ik} - D_s)} - \beta_i - \beta_{\max}, \\
\dot{h}_{ik} &\geq -\alpha_i - \alpha_k - \sqrt{\frac{\alpha_i + \alpha_k}{2(D_{ik} - D_s)}}(\beta_i + \beta_k) \\
&\geq -\alpha_i - \alpha_{\max} - \sqrt{\frac{\alpha_i + \alpha_{\max}}{2(D_{ik} - D_s)}}(\beta_i + \beta_{\max}).
\end{aligned}$$

From $D_{ik} > D^i_{\mathscr{N}}$, we get $\sqrt{2(\alpha_i+\alpha_{\min})(D_{ik}-D_s)} > \beta_i+\beta_{\max}$ and $h_{ik} > \sqrt[3]{\frac{2(\alpha_i+\alpha_{\max})}{\gamma}}$. Therefore

$$\begin{aligned} \dot{h}_{ik} &\geq -\alpha_i - \alpha_{\max} - \sqrt{(\alpha_i+\alpha_{\max})(\alpha_i+\alpha_{\min})} \\ &\geq -2(\alpha_i+\alpha_{\max}) \geq -\gamma h_{ik}^3. \end{aligned}$$

This means that no matter what control action agent k takes, it always satisfies the pairwise safety barrier constraint, with agent i. Therefore, there is no need for agent i to consider agent k, and the result follows. □

A.2 Proof of Theorem 4.4.1

Proof. If any pair of quadrotors does not collide with each other, we have $r_i \neq r_j$ and $A_{ij}(q_i,q_j) \neq 0$, $\forall i < j$. Thus, $h_{ij}(q_i,q_j)$ are individually valid control barrier functions. However, this does not imply that a common controller exists such that all constraints are satisfied.

In order to prove that a common solution exists, let $H \in \mathbb{R}^{1\times 3m}$ be a convex combination of $-A_{ij}(q_i,q_j) \in \mathbb{R}^{1\times 3m}$,

$$H = -\sum_{i<j} \alpha_{ij} A_{ij}(q_i,q_j), \tag{A.1}$$

where $[\alpha_{ij}] \in \mathscr{D}$, $\mathscr{D} = \{[\alpha_{ij}] \mid \sum_{i<j} \alpha_{ij} = 1, \alpha_{ij} \geq 0\}$.

It can be observed that H is the gradient of a convex function $F(r) : \mathbb{R}^{3m} \to \mathbb{R}$, where $r = [r_1^T, r_2^T, ..., r_m^T]^T$ denotes the aggregate position of the quadrotors,

$$F(r) = \sum_{i<j} \alpha_{ij}[(x_i-x_j)^4 + (y_i-y_j)^4 + (\frac{z_i-z_j}{c})^4]. \tag{A.2}$$

Notice that $F(r)$ is non-negative and has a global minimum of 0, when $\alpha_{ij}(r_i-r_j) = 0, \forall i < j$. Since all local minimums of the convex function $F(r)$ are global minimums [68], it can be deduced that its gradient $\nabla F(r) = H = 0$ if and only if $\alpha_{ij}(r_i-r_j) = 0, \forall i < j$.

Since $r_i \neq r_j$ (quadrotors do not collide), it can be further inferred that $H = 0$ if and only if $\alpha_{ij} = 0, \forall i < j$, which violates the fact that $\sum_{i<j} \alpha_{ij} = 1$. Thus we have shown by contradiction that $H \neq 0$. Using this fact, it is guaranteed that $Hv + \sum_{i<j} \alpha_{ij} b_{ij}(q_i, q_j) > 0$ always has a solution for any convex combination of $-A_{ij}(q_i, q_j)$, i.e.,

$$\min_{[\alpha_{ij}] \in \mathscr{D}} \sup_{v \in \mathbb{R}^{3m}} \{ \sum_{i<j} \alpha_{ij} [-A_{ij}(q_i, q_j)v + b_{ij}(q_i, q_j)] \} > 0.$$

Notice that $\mathscr{D}$ is closed and bounded, and $\mathbb{R}^{3m}$ is closed. In this case, we can exchange min and sup by using the minmax theorem [114], i.e.,

$$\begin{aligned} & \min_{[\alpha_{ij}] \in \mathscr{D}} \sup_{v \in \mathbb{R}^{3m}} \{ \sum_{i<j} \alpha_{ij} [-A_{ij}(q_i, q_j)v + b_{ij}(q_i, q_j)] \} \\ = & \sup_{v \in \mathbb{R}^{3m}} \min_{[\alpha_{ij}] \in \mathscr{D}} \{ \sum_{i<j} \alpha_{ij} [-A_{ij}(q_i, q_j)v + b_{ij}(q_i, q_j)] \} \\ = & \sup_{v \in \mathbb{R}^{3m}} \min_{i<j} \{ -A_{ij}(q_i, q_j)v + b_{ij}(q_i, q_j) \} > 0, \end{aligned}$$

which is equivalent to say that a common controller v that satisfies all the pairwise barrier constraints always exists, i.e., K_{safe} is non-empty. □

A.3 Proof of Theorem 5.1.1

Proof. If the controller satisfies $u(x) \in K(x)$, then $-B'(x; -f(x) - g(x)u) \geq -\alpha(B(x))$. Apply the chain rule for B-derivative [93], it can be shown that

$$\begin{aligned} \partial_- B(x(t)) &= -(B \circ x)'(t; -1) \\ &= -B'(x(t); x'(t; -1)) \\ &= -B'(x(t); -f(x) - g(x)u), \end{aligned}$$

where $\partial_- B(x(t)) = \lim_{a \to t^-} \frac{B(x(t)) - B(x(a))}{t - a}$ is the left time derivative of $B(x(t))$. Therefore, $\partial_- B(x(t)) \geq -\alpha(B(x))$.

Consider the differential equation $\dot{z}(t) = -\alpha(z(t))$ with $z(t_0) = B(x(t_0)) > 0$, its solution is given by

$$z(t) = \sigma(z(t_0), t),$$

due to Lemma 4.4 of [115], where σ is a class $\mathscr{K}\mathscr{L}$ function.

With the Comparison Lemma [115][1], we can get

$$B(x(t)) \geq \sigma(z(t_0), t).$$

Using the properties of class $\mathscr{K}\mathscr{L}$ function, it can be shown that $B(x(t)) > 0, \forall t \geq 0$. Thus $\mathscr{C}$ is forward invariant. □

A.4 Proof of Lemma 5.3.1

Proof. The composite barrier function candidate $B(x)$ defined on $\mathscr{T}$ is a C^r function. Thus it is equivalent to show that

$$\sup_{u \in U} [L_f B(x) + L_g B(x)\mathbf{u} + \alpha(B(x))] \geq 0, \tag{A.3}$$

Note that $B(x), B_{ij}(x)$ and $\bar{B}_{ij}(x)$ are all positive in $\mathscr{T}$. Take the logarithm of $B(x)$ and differentiate using the chain rule, we get

$$\begin{aligned} \ln(B(x)) &= \sum_{\substack{i,j \in \mathscr{M} \\ j>i}} \ln(B_{ij}) + \sum_{(i,j)\in E} \ln(\bar{B}_{ij}), \\ \frac{\dot{B}}{B} &= \sum_{\substack{i,j \in \mathscr{M} \\ j>i}} \frac{\dot{B}_{ij}}{B_{ij}} + \sum_{(i,j)\in E} \frac{\dot{\bar{B}}_{ij}}{\bar{B}_{ij}}. \end{aligned}$$

[1] Comparison Lemma also works for functions with left or right differentiability. The proof is similar to [115], and thus omitted here.

Thus the Lie Derivative along g direction is

$$\begin{aligned}
&\frac{L_g B}{B}\mathbf{u} = \sum_{\substack{i,j\in\mathscr{M}\\ j>i}} \frac{L_g B_{ij}}{B_{ij}}\mathbf{u} + \sum_{(i,j)\in E} \frac{L_g \bar{B}_{ij}}{\bar{B}_{ij}}\mathbf{u},\\
&= \sum_{\substack{i,j\in\mathscr{M}\\ j>i}} \frac{\Delta\mathbf{p}_{ij}}{B_{ij}\|\Delta\mathbf{p}_{ij}\|}\Delta\mathbf{u}_{ij} - \sum_{(i,j)\in E} \frac{\Delta\mathbf{p}_{ij}}{\bar{B}_{ij}\|\Delta\mathbf{p}_{ij}\|}\Delta\mathbf{u}_{ij},\\
&= \sum_{(i,j)\in E} \frac{\bar{B}_{ij}-B_{ij}}{B_{ij}\bar{B}_{ij}}\frac{\Delta\mathbf{p}_{ij}}{\|\Delta\mathbf{p}_{ij}\|}\Delta\mathbf{u}_{ij} + \sum_{(i,j)\notin E} \frac{\Delta\mathbf{p}_{ij}}{B_{ij}\|\Delta\mathbf{p}_{ij}\|}\Delta\mathbf{u}_{ij},\\
&= \sum_{i\in\mathscr{M}} \left[\sum_{j|(i,j)\in E} \frac{\bar{B}_{ij}-B_{ij}}{B_{ij}\bar{B}_{ij}}\frac{\Delta\mathbf{p}_{ij}}{\|\Delta\mathbf{p}_{ij}\|} + \sum_{j|(i,j)\notin E} \frac{\Delta\mathbf{p}_{ij}}{B_{ij}\|\Delta\mathbf{p}_{ij}\|} \right] \mathbf{u}_i.
\end{aligned}$$

When $L_f B = \mathbf{0}$, we have

$$\sum_{j|(i,j)\in E} \frac{\bar{B}_{ij}-B_{ij}}{B_{ij}\bar{B}_{ij}\|\Delta\mathbf{p}_{ij}\|}\Delta\mathbf{p}_{ij} + \sum_{j|(i,j)\notin E} \frac{\Delta\mathbf{p}_{ij}}{B_{ij}\|\Delta\mathbf{p}_{ij}\|} = 0, \forall i \in \mathbf{M}. \tag{A.4}$$

Define a diagonal weight matrix $W = diag(\omega_{ij}) \in \mathbb{R}^{\frac{N(N-1)}{2}\times\frac{N(N-1)}{2}}$ for a complete graph, i.e., all vertexes are connected to each other, where

$$\omega_{ij} = \begin{cases} \frac{\bar{B}_{ij}-B_{ij}}{B_{ij}\bar{B}_{ij}\|\Delta\mathbf{p}_{ij}\|} & , \quad \text{if } (i,j)\in E,\\ \frac{1}{B_{ij}\|\Delta\mathbf{p}_{ij}\|}, & \text{if } (i,j)\notin E, \end{cases}$$

Let $W^{1/2} = diag(\sqrt{\omega_{ij}})$, note ω_{ij} can be negative, in which case $W^{1/2}$ contains imaginary elements. Denote $D = [D_{ij}] \in \mathbb{R}^{N\times\frac{N(N-1)}{2}}$ as the incidence matrix for a complete graph with random orientations,

$$D_{ij} = \begin{cases} 1 & , \quad \text{if vertex } i \text{ is the tail of edge } j,\\ -1 & , \quad \text{if vertex } i \text{ is the tail of edge } j. \end{cases}$$

Then (A.4) can be written as

$$DWD^T[\mathbf{p}_1, \mathbf{p}_2, ..., \mathbf{p}_N]^T = 0,$$

which implies $W^{1/2}D^T[\mathbf{p}_1, \mathbf{p}_2, ..., \mathbf{p}_N]^T = 0$.

If $\exists \omega_{ij} \neq 0$, then $\mathbf{p}_i = \mathbf{p}_j$. This is impossible, because agents i and j can't be on top of each other in $\mathscr{C}_{ij}$. Therefore, in almost all cases, we have $L_g B \neq 0$. A control action $\mathbf{u}$ can always be found that shows (A.3) is satisfied.

If $\nexists \omega_{ij} \neq 0$, i.e., all weights ω_{ij} are zero, then the required connectivity graph is a complete graph and $\bar{B}_{ij} = B_{ij}, \forall i \neq j$. It can be shown that $L_f B$ is non-negative in this case. Therefore, in this trivial case, we have $L_g B = 0, L_f B > -\alpha(B)$ for any class $\mathscr{K}$ function α. Any control action $\mathbf{u}$ can validate that (A.3) is satisfied.

To sum up, the composite safety and connectivity barrier function $B(x)$ satisfies (A.3) $\forall x \in \mathscr{T}$, and is thus a valid PBF. □

APPENDIX B

ALGORITHMS

B.1 Algorithm 1: Decentralized Deadlock Detection Resolution

Algorithm 1 Decentralized_Deadlock_Detection_Resolution

Input: $\mathbf{u}_i, \hat{\mathbf{u}}_i, \mathbf{v}_i$
Output: $\bar{\mathbf{u}}_i, Flag\_lock$

Initialization : $Flag\_lock = False$
$\delta_{LP} = Decentralized\_LP$
if $\|\hat{\mathbf{u}}_i\| \neq 0$ AND $\|\mathbf{u}_i\| == 0$ AND $\|\mathbf{v}_i\| == 0$ **then**
  $Flag\_lock = True$
end if
if $Flag\_lock == True$ **then**
  if $\delta_{LP} > 0$ AND $\mathbf{u}_i \in$ vertex$(\mathscr{P}_i)$ **then**
    DType = 1
  else if $\delta_{LP} > 0$ AND $\mathbf{u}_i \in$ edge$(\mathscr{P}_i)$ **then**
    DType = 2
  else
    DType = 3
  end if
  switch (DType)
  case 1**:**
    $\bar{\mathbf{u}}_i$ = Decentralized_QP($k_{\gamma(\text{left})} > 1, k_{\gamma(\text{right})} < 1$)
  case 2**:**
    $\bar{\mathbf{u}}_i$ = Decentralized_QP($\hat{\mathbf{u}}_i + \delta^{\perp}$)
  default:
    $\bar{\mathbf{u}}_i = \mathbf{u}_i$
  end switch
else
  $\bar{\mathbf{u}}_i = \mathbf{u}_i$
end if
return $\bar{\mathbf{u}}_i, Flag\_lock$

B.2 Algorithm 2: Barrier Certificates based Safe Learning

Algorithm 2 Barrier Certificates based Safe Learning

Input: Initial safe set $\mathscr{C}_0 \subseteq \mathscr{X}$, GP model $\mathscr{GP}(0, k(x,x'))$, discretization $\mathscr{X}_\tau$, tolerance ε
Output: Final safe set $\mathscr{C}_n$

Initialization : $n = 0, x = x_0$

repeat
  $n = n + 1$
  Find x_{next} with (6.12)
  Design nominal controller $\hat{u} = GoTo(x, x_{\text{next}})$
  Drive to x_{next} with (6.13)
  Sample x_{next}, update GP
  Expand vol($\mathscr{C}_n$) with (6.11)
until vol($\mathscr{C}_n$)-vol($\mathscr{C}_{n-1}$)$\leq \varepsilon$
return $\mathscr{C}_n$

REFERENCES

[1] N. Hovakimyan, C. Cao, E. Kharisov, E. Xargay, and I. M. Gregory, "L1 adaptive controlfor safety-critical systems", *IEEE Control Systems*, vol. 31, no. 5, pp. 54–104, 2011.

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

[3] F. Berkenkamp, R. Moriconi, A. P. Schoellig, and A. Krause, "Safe learning of regions of attraction for uncertain, nonlinear systems with gaussian processes", in *Decision and Control (CDC), 2016 IEEE 55th Conference on*, IEEE, 2016, pp. 4661–4666.

[4] S. I. Roumeliotis and M. J. Mataric, "'Small-World" Networks of Mobile Robots", in *AAAI/IAAI*, 2000, p. 1093.

[5] M. Hehn and R. DAndrea, "Real-time trajectory generation for quadrocopters", *IEEE Transactions on Robotics*, vol. 31, no. 4, pp. 877–892, 2015.

[6] D. Mellinger and V. Kumar, "Minimum snap trajectory generation and control for quadrotors", in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, IEEE, 2011, pp. 2520–2525.

[7] A. Jaimes, S. Kota, and J. Gomez, "An approach to surveillance an area using swarm of fixed wing and quad-rotor unmanned aerial vehicles uav (s)", in *System of Systems Engineering, 2008. SoSE'08. IEEE International Conference on*, IEEE, 2008, pp. 1–6.

[8] T. Tomic, K. Schmid, P. Lutz, A. Domel, M. Kassecker, E. Mair, I. L. Grixa, F. Ruess, M. Suppa, and D. Burschka, "Toward a fully autonomous uav: research platform for indoor and outdoor urban search and rescue", *IEEE robotics & automation magazine*, vol. 19, no. 3, pp. 46–56, 2012.

[9] C. Zhang and J. M. Kovacs, "The application of small unmanned aerial systems for precision agriculture: a review", *Precision agriculture*, vol. 13, no. 6, pp. 693–712, 2012.

[10] R. M. Murray, M. Rathinam, and W. Sluis, "Differential flatness of mechanical control systems: a catalog of prototype systems", in *ASME international mechanical engineering congress and exposition*, Citeseer, 1995.

[11] M Van Nieuwstadt, M Rathinam, and R. Murray, "Differential flatness and absolute equivalence of nonlinear control systems", *SIAM Journal on Control and Optimization*, vol. 36, no. 4, pp. 1225–1239, 1998.

[12] J. R. Lawton, R. W. Beard, and B. J. Young, "A Decentralized Approach to Formation Maneuvers", *Robotics and Automation, IEEE Transactions on*, vol. 19, no. 6, pp. 933–941, 2003.

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

[14] M. Mesbahi and M. Egerstedt, *Graph Theoretic Methods in Multiagent Networks*. Princeton University Press, 2010.

[15] S. Prajna, A. Jadbabaie, and G. J. Pappas, "A Framework for Worst-case and Stochastic Safety Verification Using Barrier Certificates", *Automatic Control, IEEE Transactions on*, vol. 52, no. 8, pp. 1415–1428, 2007.

[16] A. D. Ames, J. W. Grizzle, and P. Tabuada, "Control Barrier Function Based Quadratic Programs with Application to Adaptive Cruise Control", in *Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on*, 2014, pp. 6271–6278.

[17] X. Xu, P. Tabuada, J. W. Grizzle, and A. D. Ames, "Robustness of control barrier functions for safety critical control", in *Analysis and Design of Hybrid Systems(Volume 48)*, 2015, pp. 54–61.

[18] Q. Nguyen and K. Sreenath, "Exponential control barrier functions for enforcing high relative-degree safety-critical constraints", in *2016 American Control Conference (ACC)*, IEEE, 2016, pp. 322–328.

[19] L. Wang, A. D. Ames, and M. Egerstedt, "Multi-objective compositions for collision-free connectivity maintenance in teams of mobile robots", in *Decisions and Control Conference (CDC)*, 2016, pp. 2659–2664.

[20] L. Wang, D. Han, and M. Egerstedt, "Permissive barrier certificates for safe stabilization using sum-of-squares", in *2018 American Control Conference (ACC)*, 2018, to appear.

[21] L. Wang, E. A. Theodorou, and M. Egerstedt, "Safe learning of quadrotor dynamics using barrier certificates", in *IEEE International Conference on Robotics and Automation (ICRA)*, 2018, to appear.

[22] A. Agrawal and K. Sreenath, "Discrete control barrier functions for safety-critical control of discrete systems with application to bipedal robot navigation", *Robotics: Science and Systems (RSS)*, 2017.

[23] P. Glotfelter, J. Cortés, and M. Egerstedt, "Nonsmooth barrier functions with applications to multi-robot systems", *IEEE control systems letters*, vol. 1, no. 2, pp. 310–315, 2017.

[24] W. Ren and R. W. Beard, *Distributed Consensus in Multi-vehicle Cooperative Control*. Springer, 2008.

[25] M. G. Park, J. H. Jeon, and M. C. Lee, "Obstacle avoidance for mobile robots using artificial potential field approach with simulated annealing", in *Industrial Electronics, 2001. Proceedings. ISIE 2001. IEEE International Symposium on*, IEEE, vol. 3, 2001, pp. 1530–1535.

[26] E. Rimon and D. E. Koditschek, "Exact Robot Navigation using Artificial Potential Functions", *Robotics and Automation, IEEE Transactions on*, vol. 8, no. 5, pp. 501–518, 1992.

[27] J.-O. Kim and P. K. Khosla, "Real-time obstacle avoidance using harmonic potential functions", *IEEE Transactions on Robotics and Automation*, vol. 8, no. 3, pp. 338–349, 1992.

[28] D. Fox, W. Burgard, and S. Thrun, "The Dynamic Window Approach to Collision Avoidance", *IEEE Robotics & Automation Magazine*, vol. 4, no. 1, pp. 23–33, 1997.

[29] P. Ogren and N. E. Leonard, "A Convergent Dynamic Window Approach to Obstacle Avoidance", *Robotics, IEEE Transactions on*, vol. 21, no. 2, pp. 188–195, 2005.

[30] J. van den Berg, J. Snape, S. J. Guy, and D. Manocha, "Reciprocal Collision Avoidance with Acceleration-Velocity Obstacles", in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, IEEE, 2011, pp. 3475–3482.

[31] J. Alonso-Mora, A. Breitenmoser, P. Beardsley, and R. Siegwart, "Reciprocal collision avoidance for multiple car-like robots", in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, IEEE, 2012, pp. 360–366.

[32] D. Mellinger, A. Kushleyev, and V. Kumar, "Mixed-integer quadratic program trajectory generation for heterogeneous quadrotor teams", in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, IEEE, 2012, pp. 477–483.

[33] B. J. Thibodeau, S. W. Hart, D. R. Karuppiah, J. D. Sweeney, and O. Brock, "Cascaded Filter Approach to Multi-objective Control", in *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, IEEE, vol. 4, 2004, pp. 3877–3882.

[34] M. M. Zavlanos, H. G. Tanner, A. Jadbabaie, and G. J. Pappas, "Hybrid Control for Connectivity Preserving Flocking", *Automatic Control, IEEE Transactions on*, vol. 54, no. 12, pp. 2869–2875, bee2009.

[35] S. Grammatico, F. Blanchini, and A. Caiti, "Control-sharing and merging control lyapunov functions", *IEEE Transactions on Automatic Control*, vol. 59, no. 1, pp. 107–119, 2014.

[36] C. Prieur, "Uniting local and global controllers with robustness to vanishing noise", *Mathematics of Control, Signals, and Systems (MCSS)*, vol. 14, no. 2, pp. 143–172, 2001.

[37] F. Clarke, "Lyapunov functions and discontinuous stabilizing feedback", *Annual reviews in control*, vol. 35, no. 1, pp. 13–33, 2011.

[38] A. Wills and W. Heath, "A recentred barrier for constrained receding horizon control", in *American Control Conference, 2002. Proceedings of the 2002*, IEEE, vol. 5, 2002, pp. 4177–4182.

[39] M. Z. Romdlony and B. Jayawardhana, "Uniting control lyapunov and control barrier functions", in *Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on*, IEEE, 2014, pp. 2293–2298.

[40] K. P. Tee, S. S. Ge, and E. H. Tay, "Barrier Lyapunov Functions for the Control of Output-Constrained Nonlinear Systems", *Automatica*, vol. 45, no. 4, pp. 918–927, 2009.

[41] A. Papachristodoulou and S. Prajna, "On the construction of lyapunov functions using the sum of squares decomposition", in *Decision and Control, 2002, Proceedings of the 41st IEEE Conference on*, IEEE, vol. 3, 2002, pp. 3482–3487.

[42] G. Chesi, *Domain of attraction: analysis and control via SOS programming*. Springer Science & Business Media, 2011, vol. 415.

[43] A. Majumdar, A. A. Ahmadi, and R. Tedrake, "Control design along trajectories with sums of squares programming", in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, IEEE, 2013, pp. 4054–4061.

[44] H. Fukushima, T.-H. Kim, and T. Sugie, "Adaptive model predictive control for a class of constrained linear systems based on the comparison model", *Automatica*, vol. 43, no. 2, pp. 301–308, 2007.

[45] A. Bemporad and M. Morari, "Robust model predictive control: a survey", *Robustness in identification and control*, pp. 207–226, 1999.

[46] M. P. Deisenroth, D. Fox, and C. E. Rasmussen, "Gaussian processes for data-efficient learning in robotics and control", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 2, pp. 408–423, 2015.

[47] Y. Pan and E. Theodorou, "Probabilistic differential dynamic programming", in *Advances in Neural Information Processing Systems*, 2014, pp. 1907–1915.

[48] J. Vinogradska, B. Bischoff, D. Nguyen-Tuong, H. Schmidt, A. Romer, and J. Peters, "Stability of controllers for gaussian process forward models", in *Proceedings of the 33rd International Conference on International Conference on Machine Learning*, 2016, pp. 545–554.

[49] A. Aswani, H. Gonzalez, S. S. Sastry, and C. Tomlin, "Provably safe and robust learning-based model predictive control", *Automatica*, vol. 49, no. 5, pp. 1216–1226, 2013.

[50] J. Schreiter, D. Nguyen-Tuong, M. Eberts, B. Bischoff, H. Markert, and M. Toussaint, "Safe exploration for active learning with gaussian processes", in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, Springer, 2015, pp. 133–149.

[51] S. Shalev-Shwartz, S. Shammah, and A. Shashua, "Safe, multi-agent, reinforcement learning for autonomous driving", *arXiv preprint arXiv:1610.03295*, 2016.

[52] H. B. Ammar, R. Tutunov, and E. Eaton, "Safe policy search for lifelong reinforcement learning with sublinear regret", in *International Conference on Machine Learning*, 2015, pp. 2361–2369.

[53] B. Van Niekerk, A. Damianou, and B. S. Rosman, "Online constrained model-based reinforcement learning", 2017.

[54] J. Achiam, D. Held, A. Tamar, and P. Abbeel, "Constrained policy optimization", *arXiv preprint arXiv:1705.10528*, 2017.

[55] J. Garcıa and F. Fernández, "A comprehensive survey on safe reinforcement learning", *Journal of Machine Learning Research*, vol. 16, no. 1, pp. 1437–1480, 2015.

[56] M. Ohnishi, L. Wang, G. Notomista, and M. Egerstedt, "Safety-aware adaptive reinforcement learning with applications to brushbot navigation", *arXiv preprint arXiv:1801.09627*, 2018.

[57] S. M. Khansari-Zadeh and A. Billard, "Learning control lyapunov function to ensure stability of dynamical system-based robot reaching motions", *Robotics and Autonomous Systems*, vol. 62, no. 6, pp. 752–765, 2014.

[58] H. Ravanbakhsh and S. Sankaranarayanan, "Learning lyapunov (potential) functions from counterexamples and demonstrations", *arXiv preprint arXiv:1705.09619*, 2017.

[59] Y. Ito, K. Fujimoto, and Y. Tadokoro, "Second-order bounds of gaussian kernel-based functions and its application to nonlinear optimal control with stability", *arXiv preprint arXiv:1707.06240*, 2017.

[60] F. Berkenkamp, M. Turchetta, A. P. Schoellig, and A. Krause, "Safe model-based reinforcement learning with stability guarantees", *arXiv preprint arXiv:1705.08551*, 2017.

[61] A. K. Akametalu, J. F. Fisac, J. H. Gillula, S. Kaynama, M. N. Zeilinger, and C. J. Tomlin, "Reachability-based safe learning with gaussian processes", in *Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on*, IEEE, 2014, pp. 1424–1431.

[62] X. Xu, P. Tabuada, J. W. Grizzle, and A. D. Ames, "Robustness of control barrier functions for safety critical contro", in *Analysis and Design of Hybrid Systems, 2015 IFAC Conference on*, IEEE, 2015.

[63] X. Xu, J. W. Grizzle, P. Tabuada, and A. D. Ames, "Correctness guarantees for the composition of lane keeping and adaptive cruise control", *arXiv preprint arXiv:1609.06807*, 2016.

[64] L. Wang, A. D. Ames, and M. Egerstedt, "Safety Barrier Certificates for Heterogeneous Multi-robot Systems", in *2016 American Control Conference (ACC)*, 2016, pp. 5213–5218.

[65] D. Pickem, P. Glotfelter, L. Wang, Y. Diaz-Mercado, M. Mote, A. Ames, E. Feron, and M. Egerstedt, "The robotarium: a remotely accessible swarm robotics research testbed", in *IEEE International Conference on Robotics and Automation (ICRA)*, 2017, to appear.

[66] U. Borrmann, L. Wang, A. D. Ames, and M. Egerstedt, "Control barrier certificates for safe swarm behavior", in *Analysis and Design of Hybrid Systems, 2015 IFAC Conference on*, IEEE, 2015.

[67] B. Morris, M. J. Powell, and A. D. Ames, "Sufficient conditions for the lipschitz continuity of qp-based multi-objective control of humanoid robots", in *52nd IEEE Conference on Decision and Control*, IEEE, 2013, pp. 2920–2926.

[68] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[69] L. Perera, J. Carvalho, and C. G. Soares, "Fuzzy logic based decision making system for collision avoidance of ocean navigation under critical collision conditions", *Journal of marine science and technology*, vol. 16, no. 1, pp. 84–99, 2011.

[70] K. Nagel, D. E. Wolf, P. Wagner, and P. Simon, "Two-lane traffic rules for cellular automata: a systematic approach", *Physical Review E*, vol. 58, no. 2, p. 1425, 1998.

[71] L. Wang, A. D. Ames, and M. Egerstedt, "Safety barrier certificates for collisions-free multi-robot systems", *Robotics, IEEE Transactions on*, 2017, to appear.

[72] Online, *Safety barrier certificates for collision-free multi-robot systems*, `http://tinyurl.com/htfsmnl`, 2016.

[73] D. Pickem, M. Lee, and M. Egerstedt, "The gritsbot in its natural habitat-a multi-robot testbed", in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, IEEE, 2015, pp. 4062–4067.

[74] Online, *Safety barrier certificates videos*, `https://liwanggt.github.io/`, 2017.

[75] H. Park and S. A. Hutchinson, "Fault-tolerant rendezvous of multirobot systems", *IEEE Transactions on Robotics*, vol. 33, no. 3, pp. 565–582, 2017.

[76] S. Mayya, P. Pierpaoli, G. Nair, and M. Egerstedt, "Collisions as information sources in densely packed multi-robot systems under mean-field approximations", in *Proc. Robot., Sci. Syst. Conf.*, 2017.

[77] E Xargay, I Kaminer, A Pascoal, N Hovakimyan, V Dobrokhodov, V Cichella, A. Aguiar, and R Ghabcheloo, "Time-critical cooperative path following of multiple unmanned aerial vehicles over time-varying networks", *Journal of Guidance, Control, and Dynamics*, vol. 36, no. 2, pp. 499–516, 2013.

[78] R. W. Beard, T. W. McLain, D. B. Nelson, D. Kingston, and D. Johanson, "Decentralized cooperative aerial surveillance using fixed-wing miniature uavs", *Proceedings of the IEEE*, vol. 94, no. 7, pp. 1306–1324, 2006.

[79] X. C. Ding, A. R. Rahmani, and M. Egerstedt, "Multi-uav convoy protection: an optimal approach to path planning and coordination", *Robotics, IEEE Transactions on*, vol. 26, no. 2, pp. 256–268, 2010.

[80] L. Wang, A. D. Ames, and M. Egerstedt, "Safe certificate-based maneuvers for teams of quadrotors using differential flatness", in *IEEE International Conference on Robotics and Automation (ICRA)*, 2017, to appear.

[81] *Bitcraze AB*, `https://www.bitcraze.io/`.

[82] D. Zhou and M. Schwager, "Vector field following for quadrotors using differential flatness", in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2014, pp. 6567–6572.

[83] J. Carsten, D. Ferguson, and A. Stentz, "3d field d: improved path planning and replanning in three dimensions", in *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, IEEE, 2006, pp. 3381–3386.

[84] S. Hrabar, "3d path planning and stereo-based obstacle avoidance for rotorcraft uavs", in *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, IEEE, 2008, pp. 807–814.

[85] A. H. Barr, "Superquadrics and angle-preserving transformations", *IEEE Computer graphics and Applications*, vol. 1, no. 1, pp. 11–23, 1981.

[86] M. Egerstedt, X. Hu, and A. Stotsky, "Control of mobile platforms using a virtual vehicle approach", *IEEE Transactions on Automatic Control*, vol. 46, no. 11, pp. 1777–1782, 2001.

[87] W. Hoenig, C. Milanes, L. Scaria, T. Phan, M. Bolas, and N. Ayanian, "Mixed reality for robotics", in *IEEE/RSJ Intl Conf. Intelligent Robots and Systems*, Hamburg, Germany, 2015, pp. 5382 –5387.

[88] Online, *Barrier certificates for safe quad swarm*, `https://www.youtube.com/watch?v=rK9oyqccMJw`, 2016.

[89] L. E. Parker, "Distributed intelligence: overview of the field and its application in multi-robot systems", *Journal of Physical Agents*, vol. 2, no. 1, pp. 5–14, 2008.

[90] P. Pirjanian and M. Mataric, "Multi-robot target acquisition using multiple objective behavior coordination", in *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*, IEEE, vol. 3, 2000, pp. 2696–2702.

[91] J. Ferber, *Multi-agent systems: an introduction to distributed artificial intelligence*. Addison-Wesley Reading, 1999, vol. 1.

[92] S. Scholtes, *Introduction to Piecewise Differentiable Equations*. Springer Science & Business Media, 2012.

[93] L. Kuntz and S. Scholtes, "Qualitative Aspects of the Local Approximation of a Piecewise Differentiable Function", *Nonlinear Analysis: Theory, Methods & Applications*, vol. 25, no. 2, pp. 197–215, 1995.

[94] J. R. Kok, M. T. Spaan, N. Vlassis, *et al.*, "Multi-robot Decision Making using Coordination Graphs", in *Proceedings of the 11th International Conference on Advanced Robotics, ICAR*, vol. 3, 2003, pp. 1124–1129.

[95] Online, *Composable Safety and Connectivity Barrier Certificates for Multi-Robot Systems*, `https://www.youtube.com/watch?v=LXzgxCzZIsM`, 2015.

[96] B. Tibken, "Estimation of the domain of attraction for polynomial systems via LMIs", in *Proceedings of the Conference on Decision and Control*, vol. 4, 2000, pp. 3860–3864.

[97] G. Chesi, "Rational Lyapunov functions for estimating and controlling the robust domain of attraction", *Automatica*, vol. 49, no. 4, pp. 1051–1057, 2013.

[98] P. A. Parrilo, "Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization", PhD thesis, California Institute of Technology, 2000.

[99] D. Henrion and M. Korda, "Convex computation of the region of attraction of polynomial control systems", *IEEE Transactions on Automatic Control*, vol. 59, no. 2, pp. 297–312, 2014.

[100] M. Korda, D. Henrion, and C. Jones, "Inner approximations of the region of attraction for polynomial dynamical systems", in *Proceedings of the IFAC Symposium on Nonlinear Control Systems*, 2008, pp. 2291–2296.

[101] G. Valmorbida and J. Anderson, "Region of attraction analysis via invariant sets", in *Proceedings of the American Control Conference*, 2014, pp. 3591–3596.

[102] D. Han, A. El-Guindy, and M. Althoff, "Estimating the domain of attraction based on the invariance principle", in *Decision and Control (CDC), IEEE 55th Conference on*, IEEE, 2016, pp. 5569–5576.

[103] M. Putinar, "Positive polynomials on compact semi-algebraic sets", *Indiana University Mathematics Journal*, vol. 42, no. 3, pp. 969–984, 1993.

[104] J. F. Sturm, "Using sedumi 1.02, a matlab toolbox for optimization over symmetric cones", *Optimization methods and software*, vol. 11, no. 1-4, pp. 625–653, 1999.

[105] S. Prajna, A. Papachristodoulou, and P. A. Parrilo, "Introducing sostools: a general purpose sum of squares programming solver", in *Decision and Control, 2002, Proceedings of the 41st IEEE Conference on*, IEEE, vol. 1, 2002, pp. 741–746.

[106] J. Lofberg, "Yalmip: a toolbox for modeling and optimization in matlab", in *Computer Aided Control Systems Design, 2004 IEEE International Symposium on*, IEEE, 2005, pp. 284–289.

[107] C. E. Rasmussen, "Gaussian processes for machine learning", 2006.

[108] E. Snelson and Z. Ghahramani, "Sparse gaussian processes using pseudo-inputs", in *Advances in neural information processing systems*, 2006, pp. 1257–1264.

[109] H. Peng and Y. Qi, "Eigengp: gaussian process models with adaptive eigenfunctions.", in *IJCAI*, 2015, pp. 3763–3769.

[110] M. Lázaro-Gredilla, J. Quiñonero-Candela, C. E. Rasmussen, and A. R. Figueiras-Vidal, "Sparse spectrum gaussian process regression", *Journal of Machine Learning Research*, vol. 11, no. Jun, pp. 1865–1881, 2010.

[111] Y. Pan, X. Yan, E. A. Theodorou, and B. Boots, "Prediction under uncertainty in sparse spectrum gaussian processes with applications to filtering and control", in *International Conference on Machine Learning*, 2017, pp. 2760–2768.

[112] D. Nguyen-Tuong, J. R. Peters, and M. Seeger, "Local gaussian process regression for real time online model learning", in *Advances in Neural Information Processing Systems*, 2009, pp. 1193–1200.

[113] W. Chu and Z. Ghahramani, "Preference learning with gaussian processes", in *Proceedings of the 22nd international conference on Machine learning*, ACM, 2005, pp. 137–144.

[114] R. T. Rockafellar, *Convex analysis*. Princeton university press, 2015.

[115] H. K. Khalil, *Nonlinear systems*. Prentice hall, third edition, 2002.