Constrained Control and Online Safety Filtering for Autonomous Space Systems

by

Joseph M. Breeden

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Aerospace Engineering)
in the University of Michigan
2024

Doctoral Committee:

Professor Dimitra Panagou, Chair
Professor Dennis Bernstein
Professor Anouck Girard
Professor Necmiye Ozay

Joseph M. Breeden
jbreeden@umich.edu
ORCID iD: 0000-0002-7822-3146

© Joseph M. Breeden 2024

DEDICATION

To everyone that helped me get this far.

ACKNOWLEDGEMENTS

To say that this dissertation was a marathon seems like an understatement. Needless to say, I did not make it to the finish line of this marathon without the aid of countless others. I will try to recognize as many of them as possible here.

First, I am obliged to thank the funding agencies that made my five years of curiosity and exploration at the University of Michigan possible. I thank the United States National Science Foundation Graduate Research Fellowship Program for funding the first three years of my studies. I also thank Dr. Francis DuVinage for making me aware of this program and recommending that I apply. I also thank Dr. Mary Bowden, Dr. Paul Mason, Dr. Gerardo Cruz Ortiz, and Dr. Robert Sanner for recommending me for this and other fellowships, and for everything they did during my undergraduate years to prepare me for graduate studies. Also on this list should be Dr. David Akin, who advised me and mentored various projects during my undergraduate years, and helped me decide whether to pursue graduate school. I also thank the François Xavier Bagnoud Foundation for funding the final two years of my studies, as well as Dr. Dimitra Panagou and Dr. Ilya Kolmanovsky for recommending me for this fellowship and the University of Michigan Department of Aerospace Engineering for selecting me to receive it. Between these two fellowships, I was essentially free to study any area that I wanted for the past five years, and thus chose to focus on trust in spacecraft controls, a topic that my prior experiences made me feel was important. This dissertation might look very different if I had been subject to more specific funding requirements, or if I had to pivot projects halfway through my PhD when the first fellowship expired. I cannot adequately convey my thanks in words for these five years of intellectual freedom.

Next, the person most essential to this dissertation was my advisor Dr. Dimitra Panagou. Thinking back to the beginning, I was torn choosing between two universities for graduate school, and though I did not know Dr. Panagou well at the time, her engaged attitude and kind demeanor during interviews and visit day were major factors in my choosing the University of Michigan over another school with otherwise similar technical credentials. Since arriving at Michigan, she helped steer me towards the area in which I ended up focusing, and kept me on track throughout my PhD, while still giving me great flexibility to choose the topics I found most meaningful. She helped fix my earliest papers, which frankly were not very good, and later let me know when she trusted me to submit papers without her review. Dr. Panagou is undoubtedly the person most responsible for my development into a professional researcher over the past few years. I also thank her

for recommending me for several awards and opportunities, among which I particularly note the François Xavier Bagnoud Fellowship and the Chateaubriand Fellowship (more on that one later). I also thank my labmates for their help as pseudo-advisors over the years, including senior lab members Dr. William Bentz, Dr. Kunal Garg, Dr. James Usevitch, and concurrent lab members Dr. Mitchell Black, Dr. Vishnu Chipade, and soon-to-be-doctors Devansh Agrawal, Hardik Parwana, Alia Gilbert, and Kaleb Naveed. I also thank my starting cohort "gang" Gillian McGlothin and Carleen McKenna, who made my first year in particular more welcoming. Concluding the list of Michigan contributors, I thank the rest of my PhD committee Dr. Anouck Girard, Dr. Dennis Bernstein, and Dr. Necmiye Ozay for their advice on this dissertation.

In addition to studying at Michigan, I had the great opportunity to spend five months of my last year of studies at the Laboratoire d'Analyse et d'Architecture des Systèmes in Toulouse, France. I thank my French advisor, Dr. Luca Zaccarian, for making this possible and for advising my research during the months there. None of that research is included in this document due to timing constraints, but may appear in upcoming academic publications. I also thank the Office for Science & Technology of the Embassy of France in the United States for funding this stay through the Chateaubriand Fellowship. I thank Dr. Jared Miller for making me aware of this fellowship, and again my advisor Dr. Panagou for encouraging me to pursue this fellowship and helping me figure out how to fit it in when timelines seemed tough. I thank all the other members of the Méthodes et Algorithmes en Commande team for making this exchange a fulfilling experience.

The dedication of this work is to everyone who helped me get here. That is a long list! I certainly thank my parents, Debbie and Mike Breeden, who supported me from my very first days, who pushed me and gave me the resources to excel, and who supported me when things got tough. I thank my high school robotics team, FIRST Robotics Team 2537, who pushed me even further than I knew I was capable and showed me my passion for engineering. I cannot list all the volunteer parent mentors that made that team possible, but I thank Sharon Brackett in particular for pushing me to excel and connecting me to some of the opportunities that led to my getting here. In addition to her, I thank Dr. Akin and Dr. David Lovell for helping encourage me to attend the University of Maryland, as well as all of my high school teachers. I thank Dr. Cruz Ortiz for taking a chance on hiring me for my first NASA internship after my first year of college. Dr. Cruz Ortiz also served as an advisor for my undergraduate thesis, a mentor during my early career progression, and another major figure in my decision to attend the University of Michigan. I also thank Dr. Jim O'Donnell for later hiring me as a NASA Pathways Intern. My experiences during these internships greatly shaped my choice of research topic and the directions pursued in this dissertation. I thank Dr. Sanner for first teaching me controls, and all my other Maryland professors and project-advisors—especially Dr. Akin and Dr. Bowden—for everything I learned there. I also thank senior lab members Steve Lentine, Camden Miller, Blaire Weinberg, and Daniil Gribok of the University

of Maryland Space Systems Lab for all their electronics and other practical knowledge, and all the graduate students within the Space Systems Lab who made the lab function. I suppose I should also thank the University of Maryland Department of Aerospace Engineering staff for forbidding me from graduating in three years, as otherwise I would not have double-majored in Mathematics and this dissertation would look very different.

Another less obvious group of people that contributed to this thesis is the coaches that have supported me over the years. In addition to rocket science and controls, I also actively practice figure skating. For the majority of my studies, school has been very easy for me, whereas figure skating is never easy. I can confidently say that without having pursued the at-times-seemingly-impossible challenge that is figure skating, I would be a very different person today. Thus, I thank my coaches Steven Pottenger, Melanie Bolhuis, Jacques Gilson, Bianca Marro, and Brianna Weissman for supporting me throughout my skating journey. I also thank the rest of the Ann Arbor Figure Skating Club, which has become like a second home for me these past few years, and the University of Michigan Figure Skating Club. Without these two groups, I doubt I would have made it through COVID lockdowns without going insane. I also thank my parents for supporting my participation in this sport. I know that getting to and paying for practices was not always easy, but it truly shaped who I have become.

My path these past few years had several bumps along the way, so I also want to thank another group that helped me remain healthy enough to conduct this research, which is doctors of the University of Michigan Medical System. I particularly thank Dr. Heather Vance, whose timely intervention likely saved me from what could have been a much worse case of appendicitis. Without her help, I likely would have spent much more time in the hospital and Chapter 7 of this dissertation might not have existed. I also thank Dr. Mary Byrnes and Dr. Dana Telem for helping me manage pain that arose some months after this operation. Without their timely help, my months in Toulouse likely would not have happened. I also thank all the medical providers that I saw during what has been some tumultuous years for the field of medicine in general; this work would not have happened without these people. And I thank my advisor Dr. Panagou for being flexible and concerned when these issues arose.

This dissertation certainly would not have happened without all these other groups to get me this far, so thank you all again for all that you do.

PREFACE

I begin by explaining a stylistic choice and some of the ideas that led to this particular dissertation topic. First, this dissertation makes the somewhat unconventional choice to use the first-person singular "I" for the voice of the author, and the third-person "one" to refer to the reader or anyone implementing these methods. Unlike many other technical works at time of writing, I generally avoid the first-person plural "we". I make this stylistic choice 1) to take clear ownership for my contributions and assertions, 2) to avoid excessive use of the passive voice or anthropomorphization of the document, and 3) to avoid the ambiguity of the more traditional first-person plural "we", which could refer to any combination of myself, my advisor, my lab members, my colleagues, the controls community, and/or the reader. That said, while this is a personal monograph, this work of course would not have been possible without the assistance of all of the above sources. I also note here that I interchangeably use the words "space system", "spacecraft", and "satellite" throughout this dissertation.

Next, I discuss some of the motivating problems that led me to this particular research area and the topics that follow. When I started at the University of Michigan, I was most interested in multi-agent spacecraft control. However, as I briefly explain in Chapter 8, "multi-agent" in space is a rather vague term. There are a variety of multi-agent space mission proposals, all with very different characteristics and control technology requirements. Many of these missions could be accomplished simply by individually commanding each agent to follow its own trajectory; in these cases, there exists no real need for closed-loop communication between agents, so a multi-agent mission is not necessarily a multi-agent control problem. Some of these mission concepts could be solved with existing technologies, and simply have not yet because of limited budgets and the current science priorities. In the few cases where the system could benefit from closed-loop communication, the conclusion I came to repeatedly was that a capable multi-agent system is a collection of capable and autonomous single agents. Thus, this dissertation focuses on technologies to make more autonomous single agents; multi-agent extensions are left entirely to future work.

Starting from this general goal of "improve spacecraft autonomy capabilities", I then extracted all the sub-problems presented in this dissertation. My thought process was to have some (more than one in the case of this whole document) ambitious, science-fiction-like, concept in mind, and then to think through all the other technologies that would be needed to make that happen. I encourage future researchers to use a similar approach of starting from a grand application idea rather

than just looking for gaps in existing theory. Along the way, I ran experiments until I got stuck, in which case I knew I arrived at a problem that needed further work. For instance, Chapter 5 focuses on sampled-data dynamics. From a high-level, this seems like a not-that-important technology, but it was actually an area where I repeatedly got stuck when creating simulations and where existing tools in the literature never got me unstuck. Thus, I studied sampled-data theory until I knew it well enough that I never got stuck due to controller-sampling problems again. The simulations then demonstrate the new control theory developed on more immediately practical problems rather than the much-more-complex originating ideas. As all of this was motivated by aerospace problems, this work lives between the aerospace and general control theory literature. In general, I present all the controls results as general control theory with abstract equations, applicable to any relevant system, and then I choose to specialize the results in simulation to spacecraft. Thus, future aerospace engineers will have evidence that these approaches already work for spacecraft.

Besides improving spacecraft autonomy, another influence in settling on the topic of safety filtering was my prior internships at NASA. There, I was several times asked to solve a difficult constrained—often over-constrained—control problem, which usually resulted in developing some sort of nonlinear control law. I would then demonstrate that this controller worked in a majority of the proposed use cases, and also identify its limitations. Each time this occurred, other engineers would then ask "how do we verify this control law?". Usually, the engineers sought both some sort of analytic method that showed that the method indeed worked, and Monte Carlo simulations. The following work is a response to that need for an analytic proof of algorithm performance.

Regarding organization, this dissertation is structured as a collection of papers on various topics relevant to control barrier function and safety filter theory. The introduction and conclusion are entirely original to this document. It is impossible to adequately summarize the entire history of and all possible future routes for constrained control theory, so these two chapters only present what I believe are the most important subtopics in this domain. Chapter 2 is an extension of an unsubmitted brief paper that I have been working on for a while and that tries to iron out some of the ambiguities and unnecessary assumptions that persist in the modern safety filter literature. Regarding terminology, I generally prefer the term "safety filter" in this work, because that is what this method actually does, but at present the preferred keyword in the literature is "control barrier function". Chapters 3-7 are then a concatenation of several published academic papers with many additional remarks added since publication, some corrections and clarifications, and more connections made to other chapters and to the overall goal of enhanced spacecraft autonomy.

TABLE OF CONTENTS

LIST OF FIGURES

FIGURE

LIST OF TABLES

TABLE

LIST OF DEFINITIONS

DEFINITION

ABSTRACT

This dissertation presents advances in the design of constrained control laws, specifically control barrier function (CBF) quadratic program (QP) based control laws, herein called safety filters, with a focus on space systems applications. At present, most complex space systems tasks are solved by computing trajectories on the ground and then uplinking these trajectories to the spacecraft to follow. As humans venture further into the Solar System, and launch an ever increasing number of Earth-orbiting satellites each year, there is a need for control laws that can perform more complex tasks with less ground-based supervision. At the same time, on the ground, people are surrounded by more and more autonomous systems, and there is a need for controls tools for the safety supervision of such systems. CBF-QP safety-filters provide a computationally efficient framework for both constructing constrained control laws for these ground and space applications, including nonlinear systems, and for provably demonstrating that systems satisfy given safety constraints without requiring an engineer to review every trajectory. However, constructing a CBF and safety filter for given system dynamics is still a nontrivial problem.

In brief, this dissertation presents 1) tools to constructively design CBFs that are applicable to common spacecraft dynamics, and 2) extensions to safety filters that account for realistic control law implementation in space. Though motivated by spacecraft, both of these topics are equally relevant to other robotic systems.

Following an introductory chapter reviewing constrained control methodologies, the second chapter provides a thorough technical overview of CBFs and safety filters. The third chapter presents the "high relative-degree problem", and constructive methodologies for deriving CBFs that solve this problem while respecting input constraints. This chapter assumes a deterministic system, so the fourth chapter extends safety filters to provably guarantee safety for systems with bounded uncertainties, and then modifies these constructions accordingly. Notably, these constructions still provably satisfy input constraints, whereas prior work usually relaxes input constraints if the model is uncertain. This chapter also presents conditions on when so-called "tight-tolerance objectives" are feasible with an uncertain model, and constructive tools for achieving such objectives with safety filters.

The fifth chapter considers the effect of sampled measurements and both zero-order-hold and impulsive actuators. These phenomena more closely model how real spacecraft hardware operates.

Emphasis is placed on quantifying and then minimizing conservatism. The sixth chapter presents results on how to modify the CBF construction when the safety filter is tasked with enforcing many constraints simultaneously.

The seventh chapter presents one more tool for constructing CBFs, specifically for systems with large time-scales and small actuators. This type of CBF relies on future predictions of the state trajectory, but is distinct from model predictive control (MPC) and incurs a fraction of the computational cost of MPC. The eighth chapter presents conclusions and application remarks. Rather than focusing on a single space system, case studies for various robotic and space systems are presented throughout. In summary, this dissertation presents an array of tools for designing CBFs for several space (weak gravity asteroid, high gravity asteroid, planetary orbits, attitude control, relative-motion/docking) and robotic (unicycle, double integrator, car intersection) domains and safety filters for several implementation scenarios (continuous, zero-order-hold, impulsive, perturbed), and thus enables space missions and other constrained control applications that would require higher degrees of autonomy than was previously achievable.

CHAPTER 1

Introduction

This dissertation presents a collection of tools for the design of constrained control laws for space systems. In this chapter, I discuss the motivation for studying constrained control in space and describe some of the common constraints that motivated this course of research. I also compare to the existing state of the art. In the next chapter, I cover the fundamentals of the class of tools to which this dissertation contributes. I then devote each of Chapters 3-7 to documenting a specific new design tool in detail. Finally, Chapter 8 summarizes what these new design tools accomplish, and what are the most pressing areas for future work.

1.1 Motivations

1.1.1 Space Systems Requirements

In just a few words, this dissertation tries to answer the question "how can engineers design and verify control laws for space systems that provably satisfy system requirements?". By nature, most space systems are massive projects, employing hundreds or thousands of engineers and consisting of dozens of subsystems. To keep track of all these subsystems, engineers and project managers organize space missions by writing out *requirements* that each subsystem must meet. The design of the mission is complete when every subsystem reaches a level of development at which it satisfies every requirement. For instance, a requirement of the Guidance, Navigation, and Control (GNC) system of a satellite inspection mission might read "the GNC system shall be capable of keeping the Inspector at all times at least 2 cm away from the Target at the satellites' closest point". Subrequirements might then specify the disturbance environment, measurement capabilities, and other factors that may affect this requirement in progressively greater levels of detail. Importantly, the GNC team must both 1) design a control law to address the requirement, and 2) rigorously demonstrate that the control law indeed satisfies all requirements.

In the language of control theory, one would refer to a requirement as a *constraint* on the system. One common type of constraint is state constraints, such as the requirement above that

the two satellites' states stay separated. Another common type is input constraints, which require that the control law only command the actuators to work within their physical limits. The need to ensure input constraint satisfaction also often leads to imposing additional state constraints. The science and art of designing control laws to meet state and input constraints, and other types of constraints, is *constrained control*. In space systems, all design is guided by the requirements, so space systems naturally lead to constrained control problems. Moreover, these constraints are usually *safety-critical*, meaning that violation of a constraint may lead to loss of mission and/or life.

In addition to the challenge of designing for these constraints, often for complex system dynamics, controls engineers are also constrained to designing control laws that can operate under the limited computational capabilities of radiation hardened spacecraft computers, often 20x slower than the processors for ground applications [1][1]. Thus, the final control laws must be relatively simple, and the use of onboard path-planning is limited. Computer chips manufacturers will likely eventually make radiation hardened versions of newer specialized computing modules, but these versions will follow several years after such modules become widespread in Earth-based applications.

Finally, once a spacecraft is launched, it is usually impossible to fix the spacecraft if an error occurs. Thus, engineers require very rigorous verification that all requirements are satisfied in the worst-case expected environments (or the 3σ environments for Gaussian disturbances/measurements). The need for such stringent verification further encourages controls engineers to design simple control laws with minimal internal logics. As a result, most spacecraft control laws are based on linearized trajectory tracking, where trajectories are generally computed on the ground so that 1) engineers have access to more powerful (larger, more energy-intensive, not radiation hardened) ground-based computers and 2) operators can verify by hand that computed trajectories indeed meet requirements before trajectories are uplinked to the spacecraft. This framework is inefficient and limits the types of requirements (i.e. types of space missions) that controls engineers can achieve, because spacecraft must wait for a response from the ground before proceeding with new mission phases.

As the number of satellites operating near the Earth increases, satellite operators seek control

[1]Note that "slower" is often difficult to precisely define, because space computers can be made to varying radiation tolerances, in varying architectures, and with varying mass and power consumption. However, generally speaking, space computers on the market today are rated in 100s or occasionally 1000s of DMIPs [2], while ground computers are usually rated in 1000s or 10000s of DMIPs (e.g., NXP Electronics S32G399A at 36300 DMIPs[2] and Renesas Electronics R-Car H3 at 40000 DMIPs[3]). Also note that DMIPs may or may not be representative of true performance depending on what tasks must be computed.

[2]https://www.nxp.com/company/blog/accelerating-software-defined-vehicles-and-safety-processing-with-nxp-s32g3-processors:BL-NXP-S32G3-PROCESSORS

[3]https://www.renesas.com/us/en/document/fly/vehicle-computer-vc3

laws with greater onboard autonomy to reduce ground operating costs. Moreover, as humans seek to explore more of the Solar System, the complexity of the missions required to collect scientific data or to support manned missions will increase. At the same time, these missions must contend with long communication delays and infrequent use of in-demand communication equipment. Thus, future exploration missions will require control laws that can satisfy the system requirements without constantly receiving trajectories from the ground. This dissertation is interested in a class of control laws called *safety filters* that I believe is a promising solution to meet these demands. Note that while this dissertation is motivated by space applications, what follows is mostly general control theory. Moreover, I do not focus on any specific space mission concept, but rather test the following strategies on various systems to highlight their generality.

1.1.2 Safety Filters

1.1.2.1 Two Layer Approach

In the proposed approach, a control law is divided into a two layer hierarchy, the *nominal control law* and the *safety filter*. The safety filter acts as a higher-level supervisor for the nominal control law (this is sometimes called a "simplex architecture" [3, 4]). Thus, responsibility for the satisfaction of some or all of the requirements is encoded into the safety filter, which is rigorously verified, while the nominal control law does not need to undergo the same strict verification process. Thus, the nominal control law may be arbitrarily complex (or arbitrarily simple) and may include internal decision logics, path planning with simplified models, learning-based control, and other methodologies that are difficult to verify. Instead, the safety filter will discard any proposed control inputs that may be unsafe. While this two-layer approach relaxes the challenge of verifying the nominal control law, this of course transfers that challenge to the safety filter. Thus, the majority of this dissertation presents strategies for the design of safety filters that are relevant for space systems.

1.1.2.2 Motivation for Safety Filters

The motivation for using safety filters is the need for greater trust in control laws with a high degree of autonomy, i.e. control laws that can make complicated decisions without consulting ground operators. In space systems context, "complicated decisions" often means any control law that is nonlinear.

At time of writing, one of the challenges with implementing nonlinear control laws onboard satellites is the need for stringent verification. With linear control techniques, there are widely accepted performance metrics in terms of gain and phase margins, e.g. [5, Ch. 13.26], [6]. NASA has even published its "Gold Rules" of space systems designs, including the proper choice of these

margins [7, Section 1.30]. It is possible to have such "Gold Rules" because linear control laws are inherently rather simple. However, it is difficult to define analogous rules for nonlinear systems. Instead, if a satellite application truly demands a nonlinear control law, controls engineers usually verify it by either Lyapunov perturbation analysis or Monte Carlo simulation, or frequently both. Both of these are often only possible for relatively simple nonlinear control laws. For instance, if the control law is a switched control law with more than a few cases, then Lyapunov analysis may be prohibitively complicated, and a sufficiently large set of Monte Carlo simulations may take a prohibitively long time to compute.

Instead, with the safety filter methodology, the filter is proven using Lyapunov-like analysis (e.g. Theorem 2.2) to ensure satisfaction of the requirements. That is, at every state, the filter outputs a set of verified control inputs, and filters out any nominal control inputs that are outside that set. Additionally, Monte Carlo simulation can be done with the safety filter alone (e.g. with a random nominal control law). In addition to reducing the need to verify the nominal control law, this two layer approach enables other complex behaviors previewed in Section 2.4 and described in the rest of this dissertation. Importantly, as the safety filter is an extra step of the control computation, it must be quick to compute, as is indeed the case for the class of filters described in Section 2.4.

1.2 Literature Review

In this section, I review several methodologies for the constrained control of spacecraft. While this literature review is expansive, it is certainly not complete, as that would require a literature review longer than this entire dissertation.

1.2.1 Uses and Limitations of Linear Control

First, I review how linear control is used in spacecraft. Many aspects of space missions evolve slowly in time and are amenable to linearization. For satellite rendezvous and docking in circular orbits, the Clohessy-Wiltshire equations of motion [8] in the Hill Frame [9] yield a linear time-invariant approximation of satellite motion. Various elliptical extensions [10–13] have also been proposed. Outside of these high-precision relative motion tasks, most spacecraft position control is achieved open loop. Methods for such open-loop maneuver planning are not addressed in this dissertation.

Because satellites float freely in space, they may also rotate freely in all three dimensions. Previous authors have proposed many different parameterizations of satellite attitude (i.e. orientation in 3D space), summarized for instance in [14]. Due to the wrap phenomenon of rotations, the dynamics of each of these parameterizations is inherently nonlinear. Nonetheless, the problems of

1) regulating the attitude to a single point or a continuous trajectory, or 2) reorienting the satellite from one attitude to another, are easily solved with linear control laws [14, Ch. 7]. An engineer may then conduct an analysis of the maximum expected perturbation to certify that a linear control law indeed satisfies required tolerances.

Such linear control laws are used extensively today for satellite attitude control, and to a lesser extent for satellite position control. This control methodology has a strong operational history, and is preferred for its simplicity. However, sometimes satellites encounter constraints which cannot be easily addressed by linear control. The most common reason for this is an exclusion constraint, e.g. that a docking satellite cannot enter certain sections of the state space, or that a satellite cannot point an instrument in a certain direction while reorienting. It is impossible to encode such exclusions into a linear control law. To address this, operators instead preplan trajectories on the ground (e.g. using nonlinear optimization), and only require the spacecraft to track such trajectories, sometimes using linearized closed-loop feedback, and other times entirely open loop. This strategy also has a strong operational history, and allows operators to compute fuel-optimal and/or time-optimal trajectories that would be difficult to compute onboard. However, this becomes cumbersome because of the heavy reliance on ground operators and communication equipment. By contrast, the safety filtering approach is rarely fuel-optimal or time-optimal, but allows for greater autonomy and less reliance on ground operators.

Besides such exclusion constraints, another reason for nonlinear control is the unique sensors and actuators with which spacecraft are designed. Linear feedback works well with continuously (or approximately continuously) operating sensors and actuators. However, it is often the case that measurements are available only infrequently. Due to computational constraints, the satellite computer may only be able to read sensors or send signals to actuators at slow sampling frequencies, i.e. 0.2 Hz to 5 Hz. Additionally, satellite actuators are often either low-thrust (i.e. less than 1 Newton) or impulsive, and often operate in an on/off fashion rather than with continuous variation. These are only a subset of the many possible sensor/actuator constraints that may be present on satellites. Often, controls engineers can still approximate these behaviors within linear control frameworks (e.g. including phase margin to account for controller sampling). However, many nonlinear control laws, including some in this dissertation, instead try to directly account for these sensor/actuator dynamics.

1.2.2 Review of Other Constrained Control Methodologies

Many techniques for control of satellites with constraints have already been proposed in the literature. Each of these methods has its own benefits and drawbacks, and some of these properties will also apply to safety filters. An incomplete discussion of some alternatives to safety filters is as

follows.

First, the simplest method of constrained control follows from Lyapunov-like methods, such as *Artificial Potential Fields* (APFs) and Logarithmic Barrier Functions (LBFs, also called Reciprocal Barrier Functions and other names). APFs originated as a method for robotic navigation [15, 16]. In this strategy, one assigns a potential value (i.e. an APF) to every state in the state space, frequently as the sum of several continuously differentiable potential functions. For instance, one might place a large positive potential around every excluded space, and a large negative potential around a target state so that this is the global minimum of the APF. One then designs a control law (often a linear control law) so that the velocity of an agent tracks the negative gradient of the APF. If the bandwidth of the tracking controller is sufficiently large in comparison to the second derivative of the APF, then one can also obtain strong constraint satisfaction guarantees. Alternatively, with LBFs, one instead chooses a potential function that approaches infinity near the boundary of the permissible set, e.g. [17–19]. The concept of LBFs originates with barrier functions in optimization, e.g. [20]. APFs/LBFs are a simple and powerful tool for constrained control design, and in fact, the authors in [21] showed that this technique works well with impulsive-type satellite actuators [21]. The authors in [22] further highlighted how this method of attraction to a target and repulsion from obstacles can be used to achieve complicated behaviors, and the approach has also been simulated for satellite swarm synchronization [23, 24], docking [25], and attitude control [26, 27]. However, this method still requires the design of a tracking control law, and any provable constraint satisfaction guarantees will depend on that tracking law. By comparison, safety filters work directly with the control input (usually the second derivative in space systems, though any number of derivatives is workable), so there is more literature on achieving strong guarantees with safety filters using a wide variety of actuators, including continuous [28, 29], discrete [30], zero-order-hold [31], impulsive [32, 33], underactuated [34], and mixed-order [35], among others. Indeed, the bulk of Chapters 3-7 details behaviors that can be provably accomplished with safety filters, but which I am unsure how to replicate with APFs. That said, the intuition of descending an APF often carries over to explain CBF phenomena as well. For instance, APFs frequently have local minimum in addition to the target state at the global minimum, and controllers will sometimes converge to the local minimum instead. Safety filters frequently experience the same deficiency, particularly when the nominal control law and the safety filter work against each other. Various authors have explored solutions to this for specific systems, e.g. [36, 37], but this remains an open problem for both APFs/LBFs and safety filters. Indeed, the work in [27] is of particular note because it developed a set of LBFs for satellite attitude reconfiguration that exhibits no local minimum (though this dissertation still improves upon that approach in Section 5.2.5.1). Finally, the interested reader may refer to [38] for additional comparisons between APFs and CBFs.

Second, another method similar to safety filters is the *Reference Governor* [39–41]. In this

method, one must create a pre-stabilized control input that satisfies all the system constraints. One then programs a supervisor that switches either continuously or discontinuously between the nominal control law, which generally is chosen to yield good performance of the closed-loop system, and the constraint-satisfying control law. There are many different varieties of reference governors [40], and a complete review of these is beyond the scope of this summary. One variety of particular interest is the explicit reference governor [42], as this yields explicit formulas that are computationally simple to implement. Reference governors and safety filters are closely related, as both are a "two-layer" approach, where safety and performance are decoupled. The concept of reference governors is most similar to the concept exploited by one type of safety-filter, termed the "backup CBF". In this approach, the safety filter continuously runs predictions to test whether switching to a safety-preserving control law would render the system inside the specified safe set, and if not, then the control law switches to the safety-preserving control law. By contrast, in the reference governor, the switching logic is usually computed a priori, but the concept of switching to a safety-preserving control law if necessary is similar. In fact, the variant of backup CBF in [43] could be alternately called an explicit reference governor. That said, in general, the most common difference between reference governors and safety filters as studied in this dissertation is that safety filters are generally concerned with a set of safety-preserving control inputs rather than a single safety-preserving control law (this is still true for the most versions of the backup CBF other than [43]), and thus require an optimization step to choose among this set of control inputs. That is, with safety filters the engineer never needs to actually compute a safety-preserving control law, though such a computation is in essence implicit in the construction of a controlled-invariant set as part of the safety-filter. The final safety filter as in Section 2.4 could choose any control input that results in trajectories that remain within this control invariant set, rather than a single safety-preserving control law. As a result of this abstraction and set-based approach, much of the safety filter literature (generally under the title "control barrier functions") is concerned with general tools to compute such sets, whereas the reference governor literature is often more specific to the studied system. Reference governors are also often treated as an "add-on" whereas safety filters are usually an essential part of the control design, though this perspective does not necessarily imply a mathematical difference. Reference governors have been successfully applied to satellite attitude control [44, 45] and orbit transfers [46], and remain an active area of research. Similar to safety filters, reference governors allow for provable satisfaction of constraints, but often yield inferior performance compared to optimization-based methods such as the following two approaches.

Third, to avoid the problem of local minima with the prior two methods, one might choose to implement a *Trajectory Planner* onboard a satellite. A complete review of trajectory planning methodologies is well beyond the scope of this dissertation, so instead I only highlight some relevant properties of trajectory planners. As discussed in Section 1.2.1, one common strategy is to

compute trajectories on the ground, though this limits the responsiveness of a satellite to the responsiveness of the ground station and speed-of-light delay for signals to reach the ground. Instead, trajectories can also be computed onboard the satellite, given sufficient computational resources. Usually, this means pausing the satellite in a safe-hold state while the computations for the next segment of the trajectory take place. In any case, trajectories can be 1) computed using the full system dynamics, or 2) computed using a reduced-order model. Moreover, once computed, trajectories can be followed either open-loop (usually requires knowledge of the full system dynamics with minimal error) or closed-loop (frequently used to compensate for errors in a reduced-order model, or for high-precision applications). Safety filters can be used to ensure that closed-loop trajectories stay within a prescribed tolerance of a preplanned trajectory in cases where the target trajectory is computed with a reduced-order model [47]. If the full dynamics are used for planning, then one can instead use linear perturbation analysis to describe the expected deviation from the computed trajectory. Many trajectory search and/or optimization routines exist to compute such trajectories (e.g. see [48–50]). The techniques of rapidly-exploring-randomized-trees [51–54] (and other randomized approaches) and pseudospectral trajectory optimization [55–58] in particular have been specialized to the space domain due to their comparative computational efficiency. One of the main advantages of trajectory planning routines over other methods discussed here is that, given a sufficiently accurate model, one can use trajectory planners to find fuel-optimal or time-optimal trajectories. All fuel in space must be launched on rockets, so there is a strong financial incentive to utilize fuel-optimal trajectories. By comparison, safety filters do not necessarily yield fuel-optimal trajectories, as will become apparent in the simulations throughout this dissertation. Or rather, the optimality of safety filters is limited by the optimality of the nominal control laws with which they are used (and in this dissertation, I usually choose very simple nominal control laws, or even poorly chosen nominal control laws, so as to demonstrate how the safety filter yields constraint satisfaction regardless of the nominal control law). This is one reason why ground-based trajectory planners continue to be used in practice. That said, one challenge with trajectory-planners is that it is difficult to guarantee that such planners will always converge to a good (i.e. feasible and/or globally optimal) trajectory in fixed time, which is one reason that these planners continue to be run mostly on the ground rather than onboard satellites. Additionally, a trajectory that is computed in 15 minutes on the ground may take several hours to compute onboard, and may require that the spacecraft be built with a larger computer and larger solar arrays. Thus, the computational simplicity of safety filters may be a major advantage.

Fourth, the most popular method of constrained control is *Model Predictive Control* (MPC) [59–62]. In this method, the system is almost always discretized, and often linearized. With this discretization, the controller then predicts a finite horizon of control inputs and a finite horizon of state samples. The controller then solves an optimization problem for some cost function of these

control input and state samples. Importantly, one can also add constraints to this optimization problem, and thus can encode requirements such as the exclusion constraint that was impossible to encode in a linear control law. MPC is powerful and widely used, though it too has some downsides. One limitation is that while MPC is intended as a real-time control methodology (unlike the trajectory planner, which runs offline), it is still rather computationally intensive. This computational cost is acceptable for most ground-based systems with modern computers, but may be prohibitive for satellites with radiation hardened computers depending on the control frequency (multi-rate MPC control schemes have also been suggested [63, 64]). By comparison, though safety filters also often require solving an optimization problem, this optimization problem is usually much simpler (i.e. lower dimensional and with fewer constraints) than MPC. Moreover, the computational cost of implementing a safety filter of the type in Section 2.4 is the same for both linear and nonlinear systems/constraints, whereas MPC is far more complex for nonlinear systems or constraints. This has resulted in authors finding various approximations to keep constraints linear, such as the half-space approximation in [65]. To get around this computational burden, sometimes MPC is implemented with a long duration between samples—usually with the assumption that a lower level control will also take over between MPC samples—though such a long discretization can reduce performance and/or cause a violation of constraints (e.g. as occurs for the MPC comparisons in Sections 5.2.5.1 and Section 7.1.4.2). Alternatively, a robustness margin for the MPC discretization may be added, and such a margin may be computed identically as in Chapter 5. Another challenge with MPC is that the types of exclusion constraints most common in this dissertation, e.g. obstacle avoidance, result in "holes" in the set of allowable states, thus making this set nonconvex, and making the MPC optimization a nonconvex program. By comparison, the safety filter approach ignores this nonconvexity (which also has some downsides; see Chapter 7) and thus can usually be computed as the solution to a convex program [66] or sometimes an explicit algebraic expression [67]. Next, compared to a trajectory planner, MPC-derived trajectories are usually sub-optimal. This is because 1) the horizon predicted with MPC is usually smaller than with a trajectory planner, and 2) because of the finite horizon, the cost function used with MPC is usually not the total fuel, but is some other cost function that leads to better numerical stability. That said, because MPC still allows for direct consideration of a cost function, this function provides an extra "knob" to tune to potentially achieve greater fuel-optimal than a safety filter. Finally, I note that one difference between MPC and safety filters arises from the sets with which these two methods work. As described in Chapter 3, usually one possesses both a *constraint set*, which can be read directly from the requirement, and a *controlled invariant subset* of the constraint set, which one usually has to derive. The safety filters derived in this dissertation always work with the *controlled invariant subset*. By contrast, in MPC formulations, one often constrains the MPC optimization so that the predicted states always lie within the *constraint set*. Unfortunately, this

common practice forfeits the provable guarantee of constraint satisfaction with MPC. Instead, one should additionally require that the last state in the predicted horizon also belongs to the *controlled invariant subset*. This is a minor correction, though I believe it is worth highlighting here. Unfortunately, this extra step also implies a need to a priori compute a *controlled invariant subset*, which is frequently described by nonlinear functions (whereas the *constraint set* more often can be approximated by a linear function), thus resulting in nonlinear MPC optimizations. In summary, MPC is an oft-used approach for constrained control, though it is limited by the explosion in computational cost when applied to nonlinear systems/constraints, and thus has limited applications on currently-orbiting satellites (though there exist many academic studies of MPC for satellites) in either linear or nonlinear regimes.

Fifth, one recent method for constrained control is machine learning-based control, such as reinforcement learning (both model-based and model-free) [68–70]. Similar to how with APFs one could assign a positive potential to bad states and a negative potential to target states, one can assign positive cost (equivalently, negative reward) to unallowable states and a negative cost (equivalently, positive reward) near the target state. Learning can even be done without a model, whereas safety filters and the four methods above all require knowing an accurate system model. Thus, constrained control is a natural application of machine learning. In fact, I hypothesize that machine learning will soon be one of the principal methods of developing fuel-efficient constrained control laws for aerospace systems. That said, it is difficult to derive provable guarantees about constraint satisfaction with learning-based control laws. Furthermore, learning-based control laws often work well for the majority of the state space, but may yield poorer performance or loss of safety near the edges of the training set (this can also be the case for other advanced nonlinear control laws). Thus, I hypothesize that the safety filters derived in this dissertation will be particularly relevant as supervisors for future learning-based controllers. Note that such a supervisory safety filter will still require knowledge of the system model. There is also substantial literature on learning controlled invariant sets, and associated control barrier functions for use in safety filters (discussed in Section 1.2.3.6), though learning these sets/functions is distinct from learning a complete constrained control policy. That is, one can train a control law to perform both convergence and safety simultaneously, and thus satisfy some optimality criterion, or one can focus only on learning a function for use in a typical two-layer safety filter. Both approaches are subject to the traditional limits of machine learning, e.g. computational cost, possibility of non-convergence of the learning, lack of performance guarantees for the learned object, etc..

1.2.3 Review of Safety Filters

The origins and technical details of safety filters are covered in detail in Chapter 2. Rather, this subsection describes some of the known problems with safety filters that motivated this dissertation. Each of Chapters 3-7 also begins with its own literature review.

As described in Section 1.1.2.1, the safety filter works using a two-layer control approach. Specifically, the filter works to keep a specified set forward invariant. In this work, this set is always the zero-sublevel set of a function, called a *Control Barrier Function (CBF)*. As such, I call the zero-sublevel set of this function a *CBF set*. One can approach safety filters from either a set-based perspective, or a function-based perspective, and both perspectives are used in this dissertation.

Recall that, in Lyapunov analysis, if one possesses a function meeting the conditions of a Lyapunov function, e.g. [71, Thm. 4.2], then one immediately knows that the minimizer of the Lyapunov function is stable. Likewise, if one possesses a function meeting the conditions of a Barrier Function (BF), e.g. [72, Prop. 3], then one immediately knows that the BF set is forward invariant. As extensions of these ideas, if one possesses a Control Lyapunov Function (CLF), then one knows that there exists a control input that renders the minimizer of this function asymptotically stable [73]. Similarly, if one possesses a CBF, then one knows that there exists a control input that renders the CBF set forward invariant [29]. The challenge in all of these cases is to find a function meeting the definition of a (control) barrier/Lyapunov function.

Barrier functions were studied extensively in [72, 74–77]. The generalization of barrier functions to control barrier functions was first suggested in [28], and the modern study of CBFs as in this dissertation most closely follows the formulation in [29]. In [29], the authors delineated the theory of safety filters with CBFs and showed how this theory could be applied to a specific system, in this case, adaptive cruise control and lane keeping for an autonomous vehicle. While these results are powerful and set the stage for the remainder of this dissertation, these results are also geared towards that particular system and are not immediately applicable to spacecraft dynamics. The following subsections detail some of the extensions necessary to make CBFs relevant to space systems.

1.2.3.1 Safety Filters for Systems with Inertia

As mentioned above, one challenge with using safety filters is the need to find a CBF for the given system dynamics. For simple systems, it is sometimes possible to simply write a constraint as the zero-sublevel set of a function, and then verify that said function is indeed a CBF. However, more often, further analysis is required. While the fundamentals of CBFs were established in [29], at the start of this research, there were only a few tools/strategies available to actually derive

CBFs, each for certain classes of systems. For instance, [36] developed a CBF for the double integrator system. [34] suggested a form of CBF for nonholonomic systems. The authors in [29, 78–83] each develop CBFs for specific systems of interest. The only general methods were the backstepping CBF [84], the exponential CBF [85], and preliminary work on the backup CBF [86]. These tools were each difficult to apply to spacecraft dynamics. The backstepping CBF does not allow for input constraints; the exponential CBF only allows for input constraints under certain circumstances (clarified in later works [87]) and can be difficult to tune; and the preliminary work on the backup CBF assumed explicitly integrable dynamics that do not often occur in practice. Thus, additional tools were needed.

This dissertation does not focus on any single type of space system, but rather studies miscellaneous dynamics, including navigation around a low-gravity asteroid, navigation around a high-gravity asteroid, satellite relative motion and docking, attitude control, and management of satellites in dissimilar planetary orbits, as well as the dynamics for simple robots and ground vehicles. One property that all these systems have in common is that they are second-order systems, i.e. there are states whose state derivatives are only functions of other states and not functions of the control inputs. Physically, this means that all these systems have inertia; the control input only impacts the acceleration. Most often, the safety requirements are given as constraints on the position state, and the CBF must be a strategically chosen function of both the position and velocity. Thus, this work seeks strategies to find CBFs for A) the class of systems with inertia for B) constraints placed on the position state.

A general form of CBF for this setup is the High-Order CBF (HOCBF) introduced in [87], and subsequently expanded in [88–91]. Like the exponential CBF, the HOCBF is often difficult to tune. That said, the HOCBF is indeed one of the most general possible CBF formulations, and a comparison to this approach is discussed in Section 4.3.2. A form of CBF intended for multi-link robotic manipulators, and which is suitable for the above class of systems, and constructive conditions for tuning such a CBF are described in [92, 93], though this form of CBF can be overly conservative. Importantly, the works [85, 87, 92, 93] all work best when the operating set of the system is compact. By contrast, spacecraft are often allowed to operate anywhere except a given set of collision states. One could add a constraint of a maximal keep-in radius to make the operating set compact, but this radius would be arbitrary, yet would have a large impact on the CBF parameters. Instead, by starting from the perspective of obstacle avoidance in unbounded domains, Chapters 3-4 are able to develop explicit formulas for less conservative CBFs and constructive tools to tune these CBFs.

Finally, the backup CBF formulation has appeared in many iterations, including non-exhaustively [86, 94–100], as well as this dissertation. Compared to prior works, this dissertation specializes the backup CBF to obstacle avoidance scenarios, explains how to practically compute

this CBF (as this is often not straightforward and not explained in prior works), and studies the robustness of this CBF.

The above summarizes the papers on how to derive CBFs that are most relevant to space systems and to subsequent chapters, but this is far from exhaustive. Papers on how to find CBFs for specific systems remain a prolific area of research.

1.2.3.2 Safety Filters for Systems with Disturbances

Another challenge with safety filters and CBFs is that these are inherently a model-based approach, and "all models are wrong". There is also work on finding CBFs without a full model [101–104] and on adapting CBFs to uncertain models [91, 105–108]. However, I ignore this work subsequently, because 1) as explained previously, learning-based methods are difficult to trust, and 2) in the space domain, engineers often possess very good models. Nonetheless, these models still have some uncertainty, e.g. from the density of an unexplored asteroid being unknown [109] or the particular solar conditions being difficult to predict [110, 111]. Though small compared to other terms in the system dynamics, these disturbances can still potentially cause the system trajectories to leave the CBF set, and thus should be considered in the CBF formulation. Here, I equivalently use the terms "disturbance", "perturbation", and "model uncertainty"; these disturbances can be either deterministic (e.g. unmodelled dynamics), or random (e.g. solar perturbations).

The principal results on set invariance for perturbed systems are in essence extensions of perturbed Lyapunov theory as in [71, Ch. 5]: 1) under some assumptions (namely input-to-state stability [112–115]), a small disturbance may cause a small excursion outside the CBF set [89, 116], and 2) if the disturbance has a known bound, then under some assumptions, an adjustment to the safety filter design will prevent trajectories from leaving the CBF set under even a worst-case disturbance [117–119]. As generally any amount of constraint violation is unacceptable for space missions, this dissertation focuses on the latter approach. The adjustment to the safety filter design made in [118, Def. 1] is made less conservative and extended to non-compact safe sets in Chapter 4 via Theorem 4.1.

Besides requiring a modification to the numerical condition that the safety filter enforces online, the presence of model uncertainty also might require adjustment of the CBF and CBF set. Otherwise, the modified safety filter condition may be infeasible. Similar to conventional CBFs [29], it is often easier to write out the conditions that the robust CBF must satisfy than to derive a function that meets those conditions. One of the main contributions of Chapter 4 is the development of constructive methods to derive CBFs that are valid in the presence of both input constraints and bounded model uncertainties. I am not aware of any other authors that have developed similar constructive methods to develop input-constrained CBFs for perturbed systems.

In addition to the above Lyapunov-like analysis of perturbations, there is also related literature

on probabilistic set invariance for systems with stochastic disturbances (i.e. disturbances sampled from distributions with infinite tails), commonly called "risk-bounded CBFs", e.g. [75, 120–123]. Many other system phenomena can also be treated as disturbances, and most often, robustness to these phenomena can also be addressed by small modifications to the safety filter design. For instance, other authors have studied adversarial robustness [124], sampling robustness [119, 125], and measurement robustness [126, 127] all using similar frameworks.

1.2.3.3 Safety Filters for Sampled Systems and Impulsive Systems

Indeed, this dissertation devotes substantial attention to robustness to sampling effects. The safety filters used in this work are generally based on quadratic programs (see Section 2.4). These quadratic programs take finite time to compute, especially on computationally limited satellite hardware, so their outputs are only updated infrequently. Generally, the actuators will choose a zero-order-hold control input between each new control computation. Limited sensor update frequencies may also affect the control update times. This dissertation does not study input delay, but this is also discussed in [31].

Sampling introduces two new problems: 1) the control law must ensure that the state is still inside the CBF set at the next sample, and 2) the control law must ensure that the state never leaves the CBF set between samples. The first problem in essence generates a set invariance problem in discrete time, which can be solved via discrete CBFs [30, 128, 129]. However, discrete CBFs generally lead to safety filters that are nonlinear programs rather than quadratic programs, so a conservative approximation is made instead, e.g. [31, 119, 124, 125]. The second problem is almost always solved by effectively shrinking the CBF set by some margin computed as a function of bounds on the system's first and/or second derivatives, so that the trajectory between samples provably remains within the CBF set. The definition of "effectively shrinking" will be made clearer via Definition 5.2 (relative-degree 1 case) and (5.90) (relative-degree 2 case). Depending on the conservatism of the approximation used above, additional shrinking of the CBF set might be unnecessary, as in some cases (e.g. Theorem 5.7), both problems are solved simultaneously.

Similar to robustness to disturbances, the challenges with robustness to sampling are 1) determining how to modify the safety filter, ideally with minimal conservatism, and 2) determining constructive conditions to design functions that still satisfy the definition of a CBF after this conservatism is added. As spacecraft tend to operate with slow control update cycles—almost never more than 10 Hz—the need to minimize conservatism is especially pronounced. Chapter 5.1 introduces methods to quantify and compare conservatism, and then develops an improved sampled-data safety filter that is multiple orders of magnitude less conservative than [119, 124] and that is more explicitly characterized (i.e. all formulas are given) than [31, 125, 130]. Chapter 5.2 then details constructive conditions to find CBFs that meet this revised definition of CBF.

Additionally, many spacecraft operate with impulsive thrusters, rather than continuous or even zero-order-hold thrusters. If the time between thruster applications is lower-bounded, then this can be treated very similarly to the sampled-data CBF problem, and so is covered in Chapter 5.3. To my knowledge, no other authors have considered the problem of impulsive actuators with a minimum time between actuator applications. The problem setup in this case is indeed similar to CBFs for hybrid systems [32, 33, 72, 131–135], but the safety filter design is more similar to the above sampled-data work due to the timing requirements.

1.2.3.4 Safety Filters for Multiple Constraints

Next, I should emphasize that nearly all of the above sources are for safety filters constructed from one CBF at a time. Since the early work on CBFs in [29], authors have been interested in using safety filters with multiple CBFs. However, for reasons elaborated upon in Chapter 6, CBFs do not stack together as easily as constraints stack in MPC optimizations. Or rather, whether using MPC or safety filters or some other constrained approach, if there are multiple constraints, it is easy for the control law to become infeasible. The difficulty in both cases arises from the fact that the intersection of two controlled-invariant sets is not necessarily itself a controlled-invariant set. Thus, Chapter 6 takes a set-based perspective to create constructive conditions under which one can recover a controlled-invariant subset. Other notable works that have noted the existence of this problem, though not provided such constructive conditions, include [67, 136, 137]. The authors in [138] also provided constructive conditions to check for feasibility of multiple CBFs without input constraints, and [139] developed a constructive method to identify (or rather adaptively find) a set where such a safety filter remains feasible, assuming no input constraints and that such a set exists. Given several CBFs, [140] proposes an algorithm for checking that the intersection of several CBF sets is also controlled-invariant in the presence of input constraints. This can also be accomplished with tools from the viability theory literature (see Chapter 6 for more details, though this is also a large area of literature that is not thoroughly covered in this dissertation), though such tools are currently limited for nonlinear systems and generally do not make use of certain simplifying properties of CBFs. Building on this verification work, Chapter 6 then proposes some constructive conditions to modify a set of CBFs so as to pass this verification. To my knowledge, [140] and Chapter 6 are the only works at time of writing to consider this problem in the presence of input constraints, though I predict that this will be a dense area of research in the near future. A special case of a safety filter with multiple CBFs is also presented in the case study in Section 4.7, and Chapter 6 explains further why this is a special case.

1.2.3.5 Safety Filters for Improved Performance

Next, I mentioned previously how safety-filter-based control laws tend to be suboptimal in time and/or fuel compared to offline path planners, or even online path planners like MPC. This will depend of course on the choice of nominal control law, so for the purposes of discussion, suppose that the nominal control law is agnostic to the state constraints, as this is the case for all the nominal control laws used in the subsequent chapters. Work on improving the nominal control law (usually at greater computational expense) is also considered in [128, 141–143], but here I focus instead on improving the safety filter itself. In this case, the safety-filter is suboptimal often because it only focuses on the current state and not on a horizon of states, so it can only compute control inputs that optimize a local (i.e. instantaneous) cost function, and not a cost function over a trajectory, which is usually more relevant for mission objectives like time and fuel minimization. For this reason, safety filters with CBFs are referred to as a "reactive" methodology, or sometimes "myopic". At this point, I emphasize that existence of a CBF implies that a set can be rendered forward invariant for all future time, so in some sense, a CBF does inherently encode possible future trajectories (e.g., how the CBFs described in Section 1.2.3.1 consider the system inertia). However, this property does not prevent myopic behaviors. It is also worth mentioning that a small modification of the safety-filters herein leads to the solution to a particular optimal control problem [144], though the cost function of this problem is rarely of physical importance.

Two possible reasons a safety filter may be overly conservative include 1) the CBF is overly conservative (i.e. the CBF set is smaller than the largest possible controlled invariant set) and causes a reaction sooner than would occur along an optimal trajectory, and 2) the CBF set is not very conservative and causes a reaction later than would occur along a nominal trajectory. Regarding the first problem, just as there was interest in making robust and sampled-data CBFs less conservative, there is also interest in making the CBF sets as large as possible, either by finding better algebraic expressions (e.g. Chapters 3-4) or using optimal control approaches to derive better CBFs, e.g. [98, 99, 145–147]. Deriving less conservative CBFs for specific systems remains an open area of research. Regarding the latter problem, this is a less-studied problem, and is usually addressed by modifying the nominal control law [142, 148]. Instead, in Chapter 7, I introduce a new form of CBF that directly encodes future trajectory predictions and which can be used to solve both of the above problems. Firstly, by encoding the future trajectories into the CBF (with no changes to the nominal control law), this method makes the CBF less conservative than most other approaches. Of course, this approach is still more conservative than the approaches in [98, 99] for deriving the largest possible CBF sets (assuming a sufficiently large time horizon), but this approach is computationally lightweight enough to be used with much larger time horizons than [98, 99] and thus might yield larger CBF sets anyways. Secondly, the encoding of these predictions allows for a way to tune how far in advance the CBF reacts to obstacles predicted in the future. In

the case of obstacle avoidance, reacting earlier often results in trajectories that accelerate around an obstacle rather than away from an obstacle. Note that this tuning is done in the CBF itself, resulting in a different CBF set, rather than in the rate-limit within the safety filter, as was done in [112, 115, 149].

1.2.3.6 Other Topics in Safety Filters

Sections 1.2.3.1-1.2.3.5 discuss the literature most directly relevant to Chapters 2-7. I now summarize a few other miscellaneous topics in safety filtering and CBFs that I believe the reader may find of value. First, most CBF literature, and thus most of this dissertation, works with continuously differentiable CBFs, thus yielding continuous control laws. Minor extensions to functions differentiable almost everywhere arise in Chapter 3 and Chapter 7. More thorough treatments of nonsmooth CBFs are studied in [150–154], and helpful tools not specific to safety filtering are presented in [155]. Generalizations of CBFs to hybrid systems are provided in [132], among others (see references in Chapter 5.3).

Next, in addition to using machine learning to develop complete constrained control laws, many researchers have also investigated using machine learning to design CBFs for use in safety filters, e.g. [101–104, 156–159]. In these cases, verification of the CBF must be done with some other method, since this is no longer a model-based approach, but nonetheless such learning approaches can be useful for design. There is also a small community interested in polynomial optimization to verify and/or generate a CBF [160–162], assuming the system has polynomial dynamics and the CBF can be expressed as a polynomial. These assumptions may not hold for certain space systems (because of the $\frac{\mu}{r^2}$ gravity term) and may be computationally intensive, but in the cases where such methods do apply, they yield strong set invariance guarantees. Even in other cases, polynomial optimization may be a useful early-phase design tool. Extensions to nonpolynomial systems, specifically to neural networks with linear rectifier activation functions, also recently appeared in [163].

Another relevant research area is using CBFs for designing control laws that satisfy certain time specifications, that is, CBFs for spatiotemporal logics [79, 164–167]. This is generally done with non-Lipschitz continuous control barrier functions that effect finite-time or fixed-time convergence to a target set (instead of finite-time stability to a target point [168]), though this can also be achieved with time-varying control barrier functions [152, 169]. Also related to this, researchers have investigated the concurrent design of high-level planning control laws for task satisfaction and low-level safety-preserving control laws [141, 170]. Generally, provably safe control in such a multi-rate setup is achieved via well-chosen robustness margins added to the safety filter, similar to the papers listed in Sections 1.2.3.2-1.2.3.3. Between the domains of robustness and prediction, the work in [171] recently suggested a form of CBF that takes into account a finite prediction of

the future disturbances (e.g. road gradient, expected loads) to make robust CBFs less conservative; this may also have interesting applications to multi-rate control. Though not utilized in this thesis, these approaches for autonomously reaching desired states (instead of avoiding undesired states as in this dissertation) also have the potential to improve the level autonomy of robotic and satellite control laws. Such logics have already been proposed as an alternative approach for verifying spacecraft software [172].

1.3 Contributions and Outline

The previous sections have already discussed many of the contributions of this dissertation in the context of the prior literature. Here, I provide a more concise statement of this dissertation's contributions chapter-by-chapter.

- Chapter 2 presents an overview of control barrier functions implemented in quadratic-program-based safety filters. This chapter identifies connections between the most important of the prior works and emphasizes the varying technicalities and regularity conditions employed in prior works. The chapter then formally proves three set invariance theorems (Theorem 2.2, Theorem 2.6, and Theorem 2.8) for particular regularity conditions. This chapter serves as technical preliminaries for the rest of the dissertation.

- Chapter 3 presents two new and constructive strategies for synthesizing control barrier functions for higher-order systems in the presence of input constraints. A minor generalization of the CBF regularity conditions in Chapter 2 is also presented in Theorem 3.2. This chapter focuses on the author's work in [173].

- Chapter 4 presents results on robustness of safety filters to disturbances. Specifically Theorem 4.1, Theorem 4.3, and Theorem 4.4, generalize the main set invariance theorems in Chapter 2 to systems with bounded disturbances. The chapter then presents three strategies for constructively designing control barrier functions for second-order systems (the third strategy is also applicable to higher-order systems) in the presence of input constraints and bounded disturbances—two strategies are extensions of Chapter 3 while one strategy is unique to this chapter. The chapter continues by introducing a switching approach aimed at decreasing the conservatism of robust CBFs, and a specific choice of rate-limit for the safety filter implementation that allows further tuning of this conservatism. The chapter concludes with a case study, Section 4.7, that shows how this tuning can be used to ensure that robustness does not interfere with mission objectives. This chapter focuses on the author's work in [174, 175].

- Chapter 5 presents results on implementing safety-filters in sampled-data control laws. The chapter defines two notions of conservatism, and then presents less conservative conditions that still yield provable set invariance guarantees under sampled-data control laws. The chapter then presents a constructive method for developing one form of CBF applicable to second order systems subject to sampled-data control, bounded disturbances, and input constraints (building upon the first method in Chapter 4). Theorem 5.16 and Theorem 5.17 generalize the main set invariance theorems in Chapter 2 to sampled-data control laws. The chapter concludes by deriving similar conditions for sampled-data control with impulsive actuators with larger sampling times, and presents both a set invariance theorem (Theorem 5.19) and a stability theorem (Corollary 5.22) for this domain. This chapter focuses on the author's work in [176–178].

- Chapter 6 presents constructive conditions for designing safety filters that render several control barrier functions nonpositive. The chapter identifies how the intersection of two or more controlled-invariant sets is not necessarily controlled-invariant, and proposes geometric conditions to generate a smaller controlled-invariant subset. This chapter focuses on the author's work in [179].

- Chapter 7 presents a fourth strategy for constructively designing control barrier functions for higher-order (or first-order) systems. This strategy is based on predicting the system's future trajectory under the nominal control law, and if the nominal control law is unsafe, the safety filter chooses a different control input that encourages future safety. This reduces the myopic property of the safety filter in simulation. Unlike Chapters 3-4, this method does not constructively consider input constraints. This chapter is focused on the author's work in [180].

Parts of this dissertation were also included in the tutorial paper [181].

In summary, this dissertation presents several constructive strategies for designing control barrier functions, and extends safety filtering conditions to domains not previously studied, or where existing conditions were too conservative for practical application to space systems. Importantly, all the set invariance theorems presented in Chapters 2-5 are proven or re-proven within this dissertation rather than referencing external proofs. This is done to ensure that the assumed regularity conditions in these theorems are consistent throughout.

The above are all contributions to general control theory. As this dissertation is focused on space systems, the areas of control theory I chose to study are primarily areas that I believed needed additional work to be relevant to space systems. Also, while this theory is general, the simulations that demonstrate this theory in practice are generally space systems. A list of simulations is presented in Table 1.1. Note that this dissertation does not focus on any particular space system or

Dynamics	Used in
Double Integrator	Section 3.4.1, Section 7.1.4.1
Unicycle	Section 5.1.4
Attitude Dynamics	Section 5.1.4, Section 5.2.5, Section 6.4
Asteroid (High-Thrust)	Section 3.4.2, Section 4.5.4
Asteroid (Low-Thrust)	Section 4.5.3
Low-Earth Orbit	Section 7.1.4.2
Relative Motion ([8])	Section 4.7.4, Section 5.3.4

Table 1.1: List of Simulations. List of the types of spacecraft and robot dynamics used as simulation case studies in this dissertation

space mission application, but rather develops general methods, and demonstrates these methods on various space systems throughout.

1.3.1 Three Minute Thesis

As part of my studies, I participated in a "Three Minute Thesis" challenge to communicate my research to a non-technical audience in only three minutes. I include a summary of that project here. A visual summary is also shown in Fig. 1.1.

The setup is thus: imagine you are in Paris and you want to reach the Eiffel Tower. Suppose you have no map, no access to the internet, and cannot ask others for directions (as all of these

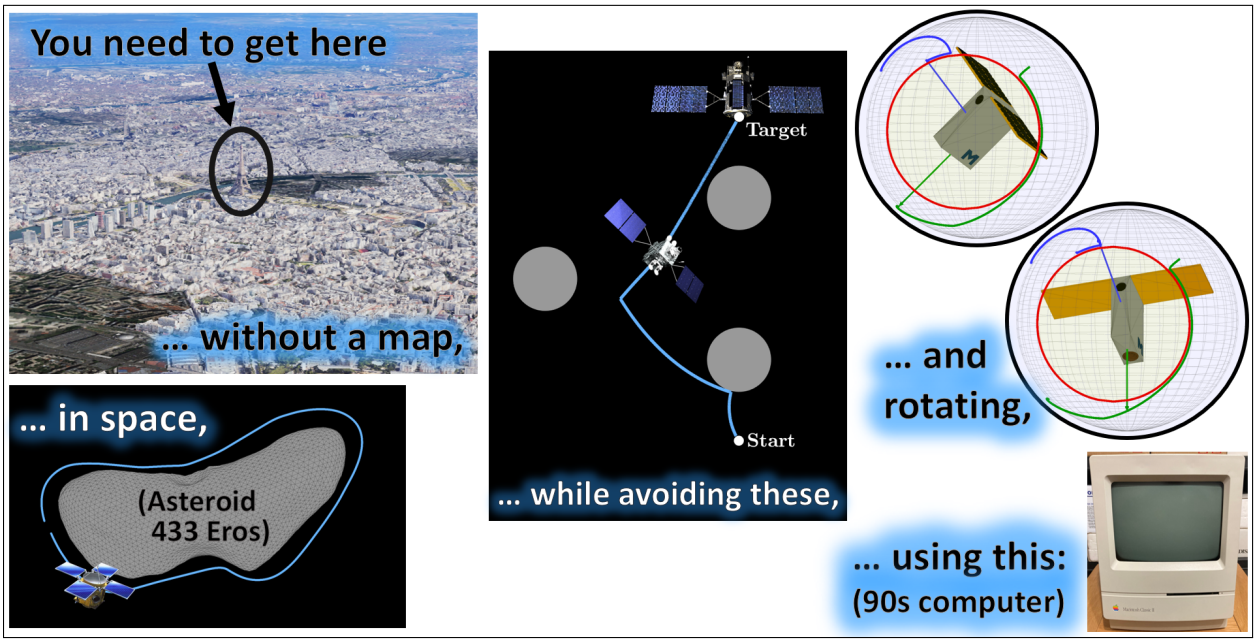

Figure 1.1: Non-Technical Explanation of Research Objectives

are computationally expensive or simply unavailable in space). However, you are close enough that you can see the Eiffel tower. Therefore you try to move generally in that direction. This is analogous to the nominal control law in the safety-filter framework. This control law can be simple or can make use of accumulated knowledge as you explore the city.

While your objective is to move generally towards the Eiffel Tower, you are at street-level, so there are many buildings in the way. You cannot move in a straight line towards the target, but rather skim along the edges of these buildings. This is analogous to the effect of the safety filter, which modifies the nominal control law to keep you away from places you cannot enter. This two-layer framework is extremely simple, but protects you from running into buildings, and if the streets of Paris are sufficiently well-connected, then you still reach the Eiffel Tower. You most likely do not take the most efficient route, which you could do if you had a map and planned out a trajectory, and it is possible that you get stuck at a dead-end road. However, this control law is much simpler to compute than a path planner. The objective of this dissertation is to advance the state-of-the-art enough to be able to implement this same simple two-layer logic on autonomous spacecraft.

CHAPTER 2

Preliminaries

In this chapter, I present a general form of the type of system analyzed throughout the remainder of the dissertation and discuss what it means for such a system to be *safe*. I also present a general definition of Control Barrier Function (CBF), three general theorems discussing how CBFs can be used to ensure safety, and a common form of control law used with CBFs. The following chapters then present more specific system models and other definitions of CBF to solve more specific problems. This chapter is intended to give a general overview of CBFs for the unfamiliar reader.

2.1 Notations

In this dissertation, calligraphic letters, e.g. $\mathcal{A}$, $\mathcal{B}$, and capital or bold Greek letters, e.g. Ξ, Ω, $\boldsymbol{\mu}$, will generally represent sets. Lower case Greek letters will generally represent scalars and scalar-valued functions. Latin characters will generally represent vectors and vector-valued functions. Note that these conventions are not followed strictly. When the need arises for quantities with multiple subscripts or superscripts, I will either define a new character to eliminate the need for multiple sub/superscripts, or else will separate the sub/superscripts with commas, e.g. $f_{a,b,c}$.

Given two objects a and b, let (a, b) represent the tuple of a and b. If a and b are both scalars or vectors, (a, b) also represents the concatenation/stacking of a and b, sometimes written as $c = [a^{\mathrm{T}}\ b^{\mathrm{T}}]^{\mathrm{T}} \equiv (a, b)$. The notations $\triangleq$ and $\equiv$ are used for emphasis when defining a new quantity and when declaring that two quantities are identical, respectively, but mathematically these characters are equivalent to the plain $=$ equality symbol. When a and b are vectors of equal dimension, the expressions $a = b$, $a \leq b$, $a < b$, $a \geq b$, and $a > b$ are interpreted elementwise. Let $\mathbb{R}$ denote the set of real numbers, $\mathbb{R}_{\geq a}$ the set of real numbers greater than or equal to a, and $\mathbb{R}_{>a}$ the set of real numbers strictly greater than a. Let $\mathbb{R}^n$ denote the set of n-dimensional vectors of real numbers, and $\mathbb{R}^{m\times n}$ the set of matrices of real numbers with m rows and n columns. Let $\mathbb{Z}$ denote the set of integers. For a number $N \in \mathbb{Z}_{\geq 1}$, let $[N] \triangleq \{1, 2, \cdots, N-1, N\}$.

Let $\|\cdot\|$ denote the two-norm, $\|\cdot\|_1$ the one-norm, and $\|\cdot\|_\infty$ the infinity-norm. Let $a \cdot b$ denote

the inner product of compatible column vectors a and b, also written as $a^{\mathrm{T}}b$. Let $a \times b$ denote the vector cross product for vectors $a, b \in \mathbb{R}^3$, and let $a^{\times}$ denote the skew-symmetric cross product matrix, such that $a^{\times}b \equiv a \times b$. Let I denote the identity matrix of appropriate dimension. The character $\exists$ is shorthand for "there exists" and the character $\forall$ is shorthand for "for all" or "for any".

Given a set $\mathcal{X}$, let $2^{\mathcal{X}}$ denote the power set of $\mathcal{X}$, that is, the set of all possible subsets of $\mathcal{X}$. Given sets $\mathcal{A}$ and $\mathcal{B}$, let $\mathcal{A} \subseteq \mathcal{B}$ denote that $\mathcal{A}$ is a subset of $\mathcal{B}$. In this dissertation, the notation $\mathcal{A} \subset \mathcal{B}$ is mathematically equivalent to $\mathcal{A} \subseteq \mathcal{B}$, but is used to further indicate that the set $\mathcal{A}$ in such a context is usually strictly smaller than the set $\mathcal{B}$, such as when $\mathcal{A}$ is assumed compact and $\mathcal{B}$ is $\mathbb{R}^n$. Let $\mathcal{A} \times \mathcal{B}$ denote the Cartesian product of $\mathcal{A}$ and $\mathcal{B}$. Let $|\mathcal{A}|$ denote the cardinality (i.e. number of elements) of the set $\mathcal{A}$. Let $\partial\mathcal{A}$ denote the boundary of a set $\mathcal{A}$, where $\mathcal{A}$ may be closed, open, or neither. Let $\mathrm{int}(\mathcal{A})$ denote the interior of $\mathcal{A}$, and $\mathrm{cl}(\mathcal{A})$ the closure of $\mathcal{A}$. Let $\mathcal{A} \cap \mathcal{B}$ denote the intersection of $\mathcal{A}$ and $\mathcal{B}$, and let $\mathcal{A} \cup \mathcal{B}$ denote the union of $\mathcal{A}$ and $\mathcal{B}$. Let $\emptyset$ denote the empty set. This dissertation will frequently utilize time-varying sets, which will be introduced as $\mathcal{A} : \mathcal{T} \to 2^{\mathcal{X}}$, where $\mathcal{T}$ is a time domain and $\mathcal{A}(t) \subseteq \mathcal{X}$. After the first introduction, such a time-varying set will be referred to only by the character $\mathcal{A}$, unless the value of the set $\mathcal{A}(t)$ at a specific time t is required. I also shorthand $\mathcal{A}^{\mathcal{T}} \triangleq \{(t, x) \in \mathcal{T} \times \mathcal{X} \mid x \in \mathcal{A}(t)\}$.

A function f that maps domain $\mathcal{A}$ to co-domain $\mathcal{B}$ will be introduced as $f : \mathcal{A} \to \mathcal{B}$. The image of $\mathcal{A}$ through f is $f(\mathcal{A})$. The expression $\frac{\partial f(t,x)}{\partial t}$ denotes the derivative (if t is scalar) or gradient row vector (if t is a vector) of the function f with respect to the function's first argument evaluated at the point (t, x). Likewise, $\frac{\partial f(t,x)}{\partial x}$ denotes the derivative or gradient with respect to the second argument. Let $\left.\frac{\partial f(x)}{\partial x}\right|_{x=y}$ denote that the derivative is evaluated at y; if such a y is not specified, then the derivative is assumed to be evaluated at the argument(s) in the numerator of the fraction. In later chapters, additional derivative notation will be introduced to clearly describe more complex arrangements. For instance, given two functions f and g with compatible domains and co-domains, the expression $\frac{\partial f(t,g(x))}{\partial x}$ is unclear using the above notation, and therefore will be clarified in the chapters where such an expression appears.

Let $\lim_{x \to a} f(x)$ denote the limit of f as the function argument approaches a, if the limit exists. Let $\lim_{x \to a^+}$ denote the one-sided limit from the right, i.e. from values greater than a. Let $\lim_{x \to a^-}$ denote the one-sided limit from the left, i.e. from values less than a. The character x is used for the state of a dynamical system throughout this dissertation, and $\dot{x}$ denotes the derivative of x in forward time, $\dot{x}(t) = \lim_{\delta \to 0^+} \frac{x(t+\delta) - x(t)}{\delta}$. For $\mathcal{T} \subset \mathbb{R}$, the expression $x(\mathcal{T})$ denotes a state trajectory curve with time domain $\mathcal{T}$; the initial condition and control law for such a curve may also be indicated or left unspecified.

Additional notations will be introduced in later chapters as needed for clarity.

2.2 Model and Assumptions

Throughout this work, consider a dynamical system of the form

$$\dot{x} = F(t, x, u) \tag{2.1}$$

where t is the time, x is the system state, and u is the control input. This work makes the following assumptions:

2.2-A1 The time t belongs to the set of considered times $\mathcal{T} \triangleq [t_0, t_f) \subseteq \mathbb{R}$, where t_f is possibly ∞;

2.2-A2 The state x belongs to the set of possible states $\mathcal{X} \subseteq \mathbb{R}^n$ (e.g. set of unit quaternions); trajectories of (2.1) starting in $\mathcal{X}$ always remain in $\mathcal{X}$ for any control input;

2.2-A3 The control input belongs to the set of allowable control inputs $\mathcal{U} \subseteq \mathbb{R}^m$, where $\mathcal{U}$ may be compact or may be unbounded depending on the chapter;

2.2-A4 The control input u is a function $u : \mathcal{T} \times \mathcal{X} \to \mathcal{U}$;

2.2-A5 The dynamics F is a function $F : \mathcal{T} \times \mathcal{X} \times \mathcal{U} \to \mathbb{R}^n$;

2.2-A6 Assume that F and u are sufficiently regular so as to always admit unique solutions to (2.1) for all times in the domain $\mathcal{T}$ and from all initial times and initial states in $\mathcal{T} \times \mathcal{X}$.

The regularity Assumption 2.2-A6 can be guaranteed using any of the sufficient conditions in [71, Ch. 3] or other works. Many works in the CBF literature further assume that F and u are locally Lipschitz continuous for this reason, but for the sake of generality, I only make such an assumption in specific chapters, not throughout the entire dissertation. Many of the subsequent chapters will redefine (2.1) to be control-affine, i.e. $F(t, x, u) = f(t, x) + g(t, x)u$, as is most common in the CBF literature.

2.3 Defining Safety and Control Barrier Functions

While humans have various intuitive notions of safety, in control theory, safety is defined precisely in terms of sets. Denote the set of all allowable states as $\mathcal{S} \subset \mathcal{X}$. For the sake of generality, suppose $\mathcal{S}$ is a time varying map $\mathcal{S} : \mathcal{T} \to 2^{\mathcal{X}}$, and denote $\mathcal{S}^{\mathcal{T}} = \{(t, x) \in \mathcal{T} \times \mathcal{X} \mid x \in \mathcal{S}(t)\}$. When working with time-varying sets $\mathcal{S}$, a "state" refers to a combination of a time and system state. I now present some definitions.

Definition 2.1 (Instantaneously Safe State)**.** *A state* $(t, x) \in \mathcal{T} \times \mathcal{X}$ *is* instantaneously safe *if* $x \in \mathcal{S}(t)$.

Definition 2.2 (Safe Trajectory)**.** *A trajectory* $x(\mathcal{T})$ *is* safe *if every state* $x(t)$ *along the trajectory is instantaneously safe for all* $t \in \mathcal{T}$.

Definition 2.3 (Perpetually Safe State)**.** *A state* (t, x) *is* perpetually safe *if there exists a control input* $u(\mathcal{T}_{\geq t})$ *such that* $u(\tau) \in \mathcal{U}$ *for all* $\tau \in \mathcal{T}_{\geq t}$ *and the trajectory* $x(\mathcal{T}_{\geq t})$ *originating from* (t, x) *under* u *is safe.*

Definition 2.4 (Viability Kernel [182, Def. 2.1])**.** *The viability kernel is the set* $\mathcal{A}^{\mathcal{T}} \subseteq \mathcal{S}^{\mathcal{T}}$ *of all perpetually safe states* (t, x).

2.3.1 Barrier Functions and Control Barrier Functions

The history of CBFs begins with Barrier Functions (BFs) for uncontrolled systems (equivalently, for systems wherein the control law is already specified). Prajna [72, 74–77] introduced the notion of *barrier certificates*, later called Barrier Functions (BFs), to certify whether states of an uncontrolled system are perpetually safe. A modified definition is as follows.

Definition 2.5 (Barrier Function [75, Thm. 1])**.** *Let* $\dot{x} = F(x)$ *and let* $\mathcal{S}$ *be time-invariant. A continuously differentiable function* $B : \mathcal{X} \to \mathbb{R}$ *is a* Barrier Function (BF) *if 1)* $B(x) > 0$ *for all* $x \in \mathcal{X} \setminus \mathcal{S}$*, 2)* $B(x) < 0$ *for some nonempty open subset of* $\mathcal{S}$*, and 3)* $\frac{\partial B(x)}{\partial x} F(x) \leq 0$ *for all* x *such that* $B(x) = 0$.

Since there is no control input, it is straightforward to test whether a function $B : \mathcal{X} \to \mathbb{R}$ satisfies Definition 2.5 or not. One can also show that any state that starts in the sublevel set $\mathcal{B} = \{x \in \mathcal{X} \mid B(x) \leq 0\}$ will stay in $\mathcal{B}$ for all future time [75, Thm. 1]. That is, $\mathcal{B}$ is forward invariant.

Definition 2.6 (Forward Invariant Set)**.** *Let* $\dot{x} = F(t, x)$*. A set* $\mathcal{A}^{\mathcal{T}} \subset \mathcal{T} \times \mathcal{X}$ *is* forward invariant *if for any* $(t, x) \in \mathcal{A}^{\mathcal{T}}$*, the trajectory* $x(\mathcal{T}_{\geq t})$ *satisfies* $(\tau, x(\tau)) \in \mathcal{A}^{\mathcal{T}}$ *for all* $\tau \in \mathcal{T}_{\geq t}$.

I emphasize that $\mathcal{B}$ as above is a subset of $\mathcal{S}$, and usually a strict subset, as much of this dissertation is concerned with identifying similar subsets and designing B so that these subsets are as large as possible. The controlled equivalent of a forward invariant set is a controlled invariant set.

Definition 2.7 (Controlled Invariant Set)**.** *A set* $\mathcal{A}^{\mathcal{T}} \subset \mathcal{T} \times \mathcal{X}$ *is* controlled invariant *if for any* $(t, x) \in \mathcal{A}^{\mathcal{T}}$*, there exists a control curve* $u(\mathcal{T}_{\geq t})$ *such that* $u(\tau) \in \mathcal{U}$ *for all* $\tau \in \mathcal{T}_{\geq t}$ *and the trajectory* $x(\mathcal{T}_{\geq t})$ *under* u *satisfies* $(\tau, x(\tau)) \in \mathcal{A}^{\mathcal{T}}$ *for all* $\tau \in \mathcal{T}_{\geq t}$.

Note that the set $\mathcal{T} \times \emptyset$ also trivially satisfies Definitions 2.6-2.7. Some papers choose to present Definitions 2.6-2.7 in terms of subsets $\mathcal{A} \subset \mathcal{X}$ rather than in terms of subsets $\mathcal{A}^{\mathcal{T}} \subset \mathcal{T} \times \mathcal{X}$; here I choose the $\mathcal{A}^{\mathcal{T}}$ formulation because this formulation applies more readily with Nagumo's theorem (repeated below as Lemma 2.7). Given a controlled invariant set, one then seeks a specific control law $u : \mathcal{T} \times \mathcal{X} \to \mathcal{U}$ such that the system $\dot{x} = F(t, x, u(t, x))$ is forward invariant. However, if a set is not controlled invariant, then no such control law will exist.

In addition to being forward invariant, $\mathcal{B}$ as above is also a *viable set*, defined as follows.

Definition 2.8 (Viable Set)**.** *A set $\mathcal{A}^{\mathcal{T}} \subseteq \mathcal{S}^{\mathcal{T}}$ is a* viable set *(also called a* viability domain*) if every state $(t, x) \in \mathcal{A}^{\mathcal{T}}$ is perpetually safe.*

Note that a viable set is not necessarily a forward invariant or controlled invariant set. Given a state (t, x) in a set $\mathcal{A}^{\mathcal{T}}$, a trajectory $x(\mathcal{T}_{\geq t})$ used to test whether $\mathcal{A}^{\mathcal{T}}$ is a viable set is permitted to leave $\mathcal{A}^{\mathcal{T}}$ as long as the trajectory does not leave $\mathcal{S}^{\mathcal{T}}$. However, by definition, any such trajectory must not leave the viability kernel, which is the largest viable set and which must be a controlled invariant set. The related field of Viability Theory is concerned with the computation of the viability kernel (e.g. [182–187]). By contrast, such computations are not studied in this dissertation, because they are often expensive even for linear systems [187, 188]. It will suffice to work with sets that are both controlled invariant and viable, but not necessarily equivalent to the viability kernel.

From the concept of BFs, Wieland and Allgöwer [28] proposed the concept of CBFs, which Ames later put into its modern form [29, 66, 189]. Note that Ames' early papers discuss what he calls a *reciprocal CBF* (a CBF that approaches ∞ at the boundary of the viable set) while Ames' later papers introduce what he calls a *zeroing CBF* (a CBF that is approaches zero at the boundary of the viable set, similar to Definition 2.5). The relation between the two forms is detailed in [29]. This dissertation works exclusively with zeroing CBFs. Moreover, at time of writing, nearly all new papers on CBF theory work with zeroing CBFs, so I drop the term "zeroing" and simply call these functions CBFs. Finally, note that BFs and CBFs have been defined in the literature such that the viable set is either the zero sublevel set (as in Definition 2.5) or the zero superlevel set (as in [29, 66, 189]) of the CBF, depending on the author. In this dissertation, all BF and CBF viable sets are zero sublevel sets of the corresponding BF/CBF. This formulation aligns with the most common convention in Viability Theory and Control Lyapunov Function (CLF) Theory (CBFs are essentially a variant of CLF), but this convention has become less popular in the most recent CBF literature.

Next, some definitions:

Definition 2.9 (Class-$\mathcal{K}$)**.** *A continuous function $\alpha : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is said to belong to class-$\mathcal{K}$, denoted $\alpha \in \mathcal{K}$, if $\alpha(0) = 0$ and α is strictly increasing.*

Definition 2.10 (Class-$\mathcal{K}_\infty$)**.** *A continuous function $\alpha : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is said to belong to class-$\mathcal{K}_\infty$, denoted $\alpha \in \mathcal{K}_\infty$, if $\alpha \in \mathcal{K}$ and $\lim_{\lambda \to \infty} \alpha(\lambda) = \infty$.*

Definition 2.11 (Class-$\mathcal{K}_e$)**.** *A continuous function $\alpha : \mathbb{R} \to \mathbb{R}$ is said to belong to extended class-$\mathcal{K}$, denoted $\alpha \in \mathcal{K}_e$, if $\alpha(0) = 0$ and α is strictly increasing.*

Definition 2.12 (Class-$\mathcal{K}_0$)**.** *A continuous function $\alpha : \mathbb{R} \to \mathbb{R}$ is said to belong to class-$\mathcal{K}$-zero, denoted $\alpha \in \mathcal{K}_0$, if $\alpha(0) = 0$, $\alpha(\lambda) \leq 0$ for all $\lambda < 0$, and $\alpha(\lambda)$ is strictly increasing for all $\lambda > 0$.*

The first step in going from BFs to CBFs involves noting a property of the trajectories of systems for which there exists a BF.

Lemma 2.1 ([29, Prop. 3])**.** *Let $\dot{x} = F(x)$, let $\mathcal{S}$ be time-invariant, and let $B : \mathcal{X} \to \mathbb{R}$ be a BF as in Definition 2.5. Let $\mathcal{B} = \{x \in \mathcal{X} \mid B(x) \leq 0\}$ be compact. Then there exists a set $\mathcal{D} \subseteq \mathcal{X}$ such that $\mathcal{B} \subseteq \mathcal{D}$ and a function $\alpha \in \mathcal{K}_e$ such that $\frac{\partial B(x)}{\partial x} F(x) \leq \alpha(-B(x))$ for all $x \in \mathcal{D}$.*

That is, the rate of increase of the BF is upper bounded by a class-$\mathcal{K}_e$ function. In the remainder of this section, I provide a definition of CBF, and then examine several of the peculiarities of some of the most common CBF theorems in the literature. Already, one notes two peculiarities of Lemma 2.1 that arise in certain papers: 1) the assumption that $\mathcal{B}$ is compact, and 2) the introduction of a larger set of definition $\mathcal{D}$. In Lemma 2.1, and often in other theorems, compactness of $\mathcal{B}$ is a critical assumption used in the proof in [29, Prop. 3]. In other papers in the CBF literature, the compactness assumption is sometimes made only as a means to guarantee existence of solutions of (2.1) for all time via [71, Thm 3.3], and thus may not be critical to all theoretical results. Therefore, when a compactness assumption is made in this dissertation, I will make clear the reasoning. The importance of the larger set $\mathcal{D}$ will also become apparent in the proof of Theorem 2.2 and Theorem 2.6 (note that one such $\mathcal{D}$ in Lemma 2.1 is simply $\mathcal{D} = \mathcal{B}$, whereas Theorem 2.2 requires $\mathcal{D}$ to be larger than $\mathcal{B}$ (or rather $\mathcal{H}$)).

From Lemma 2.1, Ames redefines BFs as follows (note that I have switched Ames' original definition to the zero sublevel set convention used in this dissertation).

Definition 2.13 (Barrier Function as in [29, Def. 3])**.** *Let $\dot{x} = F(x)$. Let $B : \mathcal{X} \to \mathbb{R}$ be continuously differentiable and let $\mathcal{B} = \{x \in \mathcal{X} \mid B(x) \leq 0\}$. The function B is a barrier function if there exists a set $\mathcal{D} \subseteq \mathcal{X}$ and a function $\alpha \in \mathcal{K}_e$ such that $\mathcal{B} \subseteq \mathcal{D}$ and $\frac{\partial B(x)}{\partial x} F(x) \leq \alpha(-B(x))$ for all $x \in \mathcal{D}$.*

Note that Definition 2.13 does not require $\mathcal{B}$ to be compact, but the equivalence between Definitions 2.5 and 2.13 only holds if $\mathcal{B}$ is compact and is a subset of $\mathcal{S}$. I am now ready to present a definition of CBF, in which I switch to using the character h as is more common in the CBF literature.

Definition 2.14 (Control Barrier Function)**.** *Let $h : \mathcal{T} \times \mathcal{X} \to \mathbb{R}$ be continuously differentiable, and let*

$$\mathcal{H}(t) \triangleq \{x \in \mathcal{X} \mid h(t,x) \leq 0\}, \tag{2.2}$$

$$\mathcal{H}^{\mathcal{T}} \triangleq \{(t,x) \in \mathcal{T} \times \mathcal{X} \mid h(t,x) \leq 0\}. \tag{2.3}$$

The function h is a control barrier function (CBF) *for the system* (2.1) *if there exists a set $\mathcal{D} : \mathcal{T} \to 2^{\mathcal{X}}$ and a function $\alpha \in \mathcal{K}_e$ such that $\mathcal{H}(\tau) \subset \mathcal{D}(\tau)$ for all $\tau \in \mathcal{T}$ and*

$$\inf_{u \in \mathcal{U}} \left[\frac{\partial h(t,x)}{\partial t} + \frac{\partial h(t,x)}{\partial x} F(t,x,u) \right] \leq \alpha(-h(t,x)), \forall (t,x) \in \mathcal{H}^{\mathcal{T}}. \tag{2.4}$$

Given a CBF, I refer to $\mathcal{H}$ and/or $\mathcal{H}^{\mathcal{T}}$ as the *CBF viable set*, or simply the *CBF set*. By construction, $\mathcal{H}(t)$ must be a closed set for every $t \in \mathcal{T}$ [190, Cor. 11.13]. Note that unlike Definition 2.5, Definitions 2.13-2.14 do not assume that $\mathcal{H}(t) \subseteq \mathcal{S}(t)$ for all $t \in \mathcal{T}$ or that $\mathcal{H}(t)$ is nonempty for all $t \in \mathcal{T}$, though these assumptions are usually required for the CBF to be of practical utility. Similarly, note that the formulation in [169], to which Definition 2.14 and Theorem 2.2 are most similar, also requires that $\mathcal{H}(t)$ possess no vanishing disconnected subsets. This dissertation does not explicitly make this assumption, because it is not necessary to prove Theorem 2.2, but in practice this property usually holds.

2.3.2 Set Invariance Theorems

Given a CBF, one has the following controlled invariance theorem.

Theorem 2.2. *Given a CBF for the system* (2.1) *as in Definition 2.14, let $\mathcal{D} : \mathcal{T} \to 2^{\mathcal{X}}$ be an open set such that $\mathcal{H}(\tau) \subset \mathcal{D}(\tau)$ for all $\tau \in \mathcal{T}$, let $\mathcal{D}^{\mathcal{T}} = \{(t,x) \in \mathcal{T} \times \mathcal{X} \mid x \in \mathcal{D}(t)\}$, and let $\alpha \in \mathcal{K}_e$. Then any control law $u : \mathcal{T} \times \mathcal{X}$ satisfying*

$$\frac{\partial h(t,x)}{\partial t} + \frac{\partial h(t,x)}{\partial x} F(t,x,u(t,x)) \leq \alpha(-h(t,x)), \forall (t,x) \in \mathcal{D}^{\mathcal{T}} \tag{2.5}$$

will render $\mathcal{H}^{\mathcal{T}}$ in (2.2) *forward invariant.*

Lemma 2.3. *Suppose $z : [t_0, \infty) \to \mathbb{R}$ is absolutely continuous and satisfies $\dot{z}(t) = \alpha(-z(t))$ for almost every $t \in [t_0, \infty)$, where $\alpha \in \mathcal{K}_e$. If $z(t_0) \leq 0$, then $z(t) \leq 0$ for all $t \geq t_0$.*

Proof. Suppose otherwise; that is, suppose $z(t_0) \leq 0$ and that there exists $t > t_0$ such that $z(t) > 0$. Since z is absolutely continuous, this implies there exists an open set $\mathbf{\Omega}$ where $z(t) > 0$ for all $t \in \mathbf{\Omega}$. Thus, without loss of generality, assume that z is differentiable at t. This implies that $\dot{z}(t) = \alpha(-z(t)) < 0$. It follows that, at an infinitesimal time $t - \delta$ preceding t, it must be that

$z(t-\delta) > z(t)$. This further implies that $z(\tau) > z(t)$ for all $\tau < t$, including $\tau = t_0$. This implies that $z(t_0) > z(t) > 0$, which contradicts the assumption that $z(t_0) \leq 0$. ■

Proof of Theorem 2.2. By assumption 2.2-A6, solutions to (2.1) are unique. Let $x(\mathcal{T})$ be any solution to (2.1) and define $\theta(t) = h(t, x(t))$. Controlled invariance is only concerned with states originating from $x(t_0) \in \mathcal{H}(t_0)$ for some $t_0 \in \mathcal{T}$, so $\theta(t_0) = h(t_0, x(t_0)) \leq 0$. The choice $u(t, x)$ satisfying (2.5) ensures that $\dot{\theta}(t) \leq \alpha(-\theta(t))$ for almost every $t \in \mathcal{T}$. Thus, by the comparison lemma [191, Lemma IX.2.6], $\theta(t) \leq \bar{z}(t)$, where $\bar{z}(t)$ is the maximum solution of $\dot{z}(t) = \alpha(-z(t))$ with $z(t_0) = \theta(t_0)$. By Lemma 2.3, every solution of $\dot{z}(t) = \alpha(-z(t))$ starting from $z(t_0) = \theta(t_0) \leq 0$ satisfies $z(t) \leq 0$ for all $t \in \mathcal{T}$. Therefore, $\bar{z}(t) \leq 0$ for all $t \in \mathcal{T}$, so $h(t, x(t)) = \theta(t) \leq \bar{z}(t) \leq 0$ and thus $x(t) \in \mathcal{H}(t)$ for all $t \in \mathcal{T}$, which is equivalent to $\mathcal{H}^{\mathcal{T}}$ being forward invariant. ■

Note two peculiarities of Theorem 2.2. First, I use the comparison lemma to complete the proof. There are two principal ways of proving forward invariance with CBFs: using the comparison lemma and using Nagumo's theorem. In particular, the above proof used an extended version of the comparison lemma since the comparison equation $\dot{z} \leq \alpha(-z)$ was permitted to be non-Lipschitz, and therefore to possibly have multiple solutions. Second, the statement of Theorem 2.2 required (2.5) to apply not only inside $\mathcal{H}$, but also inside a neighborhood $\mathcal{D}$. This is a general feature of set invariance theorems in continuous-time systems: it is not sufficient to design a control law only on $\mathcal{H}$; the control law must also be defined within an (arbitrarily small) open region around $\mathcal{H}$. Practically, this detail matters because most real control laws are sampled, so small excursions outside $\mathcal{H}$ may occur. Likewise, numerical simulators with multi-step Runge-Kutta propagation may take steps outside $\mathcal{H}$ before averaging is applied. Thus, the system dynamics should be well-defined outside $\mathcal{H}$ and should not discontinuously change at $\partial\mathcal{H}$. This might occur, for instance, if $\partial\mathcal{H}$ represents a physical wall, where an agent might start to experience contact dynamics with the wall; such a CBF should be re-designed to have a small amount of margin before the dynamics change. These implementation realities are mitigated by the following corollary.

Corollary 2.4. *Given the assumptions of Theorem 2.2, let $\mathcal{H}^\epsilon(t) = \{x \in \mathcal{X} \mid h(t, x) \leq \epsilon\}$, and further assume that there exists some $\epsilon \in \mathbb{R}_{>0}$ such that $\mathcal{H}^\epsilon(t) \subseteq \mathcal{D}(t)$ for all $t \in \mathcal{T}$. Then the set $\mathcal{H}^{\mathcal{T}}$ is asymptotically stable from all points in $\mathcal{H}^{\epsilon,\mathcal{T}} = \{(t, x) \in \mathcal{T} \times \mathcal{X} \mid x \in \mathcal{H}^\epsilon(t)\}$.*

Corollary 2.5. *Theorem 2.2 also applies if α belongs to $\mathcal{K}_0$ instead of $\mathcal{K}_e$. However, in this case, Corollary 2.4 only implies that $\mathcal{H}^{\mathcal{T}}$ is stable, not necessarily asymptotically stable, from all points in $\mathcal{H}^{\epsilon,\mathcal{T}}$.*

That is, the assumptions of Theorem 2.2 also result in $\mathcal{H}$ being attractive from $\mathcal{H}^\epsilon$ (or simply from $\mathcal{D}$). On the other hand, Corollary 2.5 implies that the dynamics in $\mathcal{D} \setminus \mathcal{H}$ need only be non-divergent, not necessarily attractive.

Next, it may be the case that one wishes to avoiding enforcing condition (2.5) everywhere in an open set $\mathcal{D}$. In this case, the following theorem provides an alternate regularity condition.

Theorem 2.6. *Let $u : \mathcal{T} \times \mathcal{X} \to \mathcal{U}$ be a control law, let h be a CBF for the system (2.1) as in Definition 2.14, and let $\alpha \in \mathcal{K}$. Let $\mathcal{D} : \mathcal{T} \to 2^{\mathcal{X}}$ be an open set such that $\partial\mathcal{H}(\tau) \subset \mathcal{D}(\tau)$ for all $\tau \in \mathcal{T}$, and let $\mathcal{D}^{\mathcal{T}} = \{(t, x) \in \mathcal{T} \times \mathcal{X} \mid x \in \mathcal{D}(t)\}$. Suppose that F is locally Lipschitz continuous in its third argument for all $u \in \mathcal{U}$, and suppose that F, u, $\frac{\partial h}{\partial t}$, and $\frac{\partial h}{\partial x}$ are locally Lipschitz continuous for all $(t, x) \in \mathcal{D}^{\mathcal{T}}$. Then any control law satisfying the above conditions and*

$$\frac{\partial h(t,x)}{\partial t} + \frac{\partial h(t,x)}{\partial x} F(t, x, u(t, x)) \leq \alpha(-h(t, x)), \forall (t, x) \in \mathcal{H}^{\mathcal{T}} \tag{2.6}$$

will render $\mathcal{H}^{\mathcal{T}}$ forward invariant.

Proof. Let $x(\mathcal{T})$ be any solution to (2.1) and define $\theta(t) = h(t, x(t))$. Solutions to (2.1) must be absolutely continuous, so any solution starting at $\theta(t_0) = h(t_0, x(t_0)) \leq 0$ must pass through $\theta(t) = 0$ for some $t \in \mathcal{T}_{\geq t_0}$ to become unsafe. Therefore, I only focus on the behavior of $\dot{\theta}$ in a neighborhood $\theta \in (-\epsilon, \epsilon)$ of $\theta = 0$, where $\epsilon \in \mathbb{R}_{>0}$. By assumption, $\dot{\theta}(t) = \frac{\partial h(t,x(t))}{\partial t} + \frac{\partial h(t,x(t))}{\partial x} F(t, x, u(t, x))$ is locally Lipschitz continuous for states $(t, x) \in \mathcal{D}^{\mathcal{T}}$, or equivalently for $\theta \in (-\epsilon, \epsilon)$. In this neighborhood, there exists a bound $\beta : \mathbb{R} \to \mathbb{R}$ such that 1) $\dot{\theta}(t) \leq \beta(-\theta(t))$ for all $\theta(t) \in (-\epsilon, \epsilon)$, 2) $\beta(-\lambda) \leq \alpha(-\lambda)$ for $\lambda \in (-\epsilon, 0]$ (recall that α is not defined for negative arguments), and 3) $\beta(\lambda)$ is locally Lipschitz continuous for $\lambda \in (-\epsilon, \epsilon)$. Thus, it follows that $\beta(0) \leq 0$, so one can linearly upper bound β as $\beta(\lambda) \leq L|\lambda|$ for some $L \in \mathbb{R}_{\geq 0}$ on the neighborhood $(-\epsilon, \epsilon)$. For any initial condition $z(t_0) \leq 0$, the system $\dot{z}(t) = L|z(t)|$ has solution $z(t) = z(t_0)e^{-L(t-t_0)}$, so it follows that $z(t) \leq 0$ for all $t \geq t_0$. By the comparison lemma [191, Lemma IX.2.6], any solution θ with $\theta(t_0) \leq 0$ also satisfies $\theta(t) \leq 0$ for all $t \in \mathcal{T}_{\geq t_0}$. This is equivalent to $\mathcal{H}^{\mathcal{T}}$ being forward invariant. ■

Compared to Theorem 2.2, Theorem 2.6 makes a stricter Lipschitz continuity assumption but only requires (2.6) to apply inside the CBF set, not on a larger open set. To see why this Lipschitz assumption is helpful, consider the system $\dot{z} = \sqrt{|z|}$ with $z(0) = 0$. This system has two solutions: $z(t) \equiv 0$ and $z(t) = \frac{1}{4}t^2$. One solution remains non-positive (i.e. safe) while the other solution becomes positive (i.e. unsafe). The additional Lipschitz assumption prevents this possibility of multiple solutions. Compared to Corollary 2.5, Theorem 2.6 now allows the dynamics on $\mathcal{D} \setminus \mathcal{H}$ to diverge from $\mathcal{H}$, as long as this divergence is locally Lipschitz in nature (as otherwise initial conditions with $h(t_0, x(t_0)) = 0$ might diverge as well).

On closer analysis of the proof of Theorem 2.6, one might conclude that Theorem 2.6 is a special case of Theorem 2.2, since the Lipschitz assumption implies that one can construct a bounding function on a larger set $\mathcal{D}$. That said, I present these two cases separately as they represent different

points of view, and to highlight some of the non-obvious ways Lipschitz assumptions are used in the broader CBF literature. Note that some papers instead assume that $\alpha \in \mathcal{K}$ is locally Lipschitz continuous for various reasons, but this is only a sufficient condition for forward invariance if $\alpha \in \mathcal{K}_e$ (not just $\mathcal{K}$) and the condition (2.5) holds for all (t, x) in the open set $\mathcal{D}^{\mathcal{T}}$ (not just on $\mathcal{H}^{\mathcal{T}}$).

Lastly, I introduce Nagumo's theorem, originally presented in [192], and show how the result in Theorem 2.2 and Theorem 2.6 can also be derived from that theorem. Recall the definition of tangent cone for a closed, possibly nonconvex set (first termed the *contingent cone* in [193, Page 66]).

Definition 2.15 (Tangent Cone [194, Def. 4.6])**.** *Given a closed set $\mathcal{A} \subseteq \mathbb{R}^n$, the tangent cone of $\mathcal{A}$ at $x \in \mathbb{R}^n$ is*

$$\mathbf{T}_{\mathcal{A}}(x) \triangleq \left\{ v \in \mathbb{R}^n \mid \liminf_{k \to 0} \frac{\inf_{w \in \mathcal{S}} \|x + kv - w\|}{k} = 0 \right\}. \tag{2.7}$$

Lemma 2.7 (Nagumo's Theorem [194, Cor. 4.8])**.** *Let $\dot{x} = F(x)$. A closed and time-invariant set $\mathcal{A} \subseteq \mathbb{R}^n$ is forward invariant if and only if $F(x) \in \mathbf{T}_{\mathcal{A}}(x)$ for all $x \in \mathcal{A}$.*

Note that Nagumo's theorem is a necessary and sufficient condition for set invariance, whereas Theorem 2.2 and Theorem 2.6 only stated sufficient conditions. Necessity of existence of a BF is studied in [76], so under the conditions of Lemma 2.1, existence of a CBF is guaranteed as well [28, Remark 11], [29, Page 3866]. However, in more complex "corner cases", Nagumo's theorem is often easier to apply to answer whether a set is forward invariant. Nagumo's theorem only applies to time-invariant sets, which is why I wrote Definitions 2.6-2.7 in terms of $\mathcal{A}^{\mathcal{T}} \subset \mathcal{T} \times \mathcal{X}$ instead of $\mathcal{A} : \mathcal{T} \to \mathcal{X}$. Also note that Nagumo's theorem depends heavily on the uniqueness assumption in Assumption 2.2-A6; if instead there exist multiple solutions, then Nagumo's theorem only guarantees that one of these solutions remains in $\mathcal{A}$ [195].

Ames then used Nagumo's theorem to state an alternate CBF forward invariance theorem [66, Thm. 2]. A time-varying extension of this theorem is presented as follows.

Theorem 2.8. *Let $\mathcal{T}$ be a closed set, either $[t_0, t_f]$ or $[t_0, \infty)$, so that $\mathcal{H}^{\mathcal{T}}$ in (2.3) is a closed set. Given a CBF for the system (2.1), assume that $|\frac{\partial h(t,x)}{\partial t}| + \|\frac{\partial h(t,x)}{\partial x}\| \neq 0$ for all $(t, x) \in \partial\mathcal{H}^{\mathcal{T}}$, and let $\alpha \in \mathcal{K}$. Then any control law $u : \mathcal{T} \times \mathcal{X}$ satisfying (2.6) will render $\mathcal{H}^{\mathcal{T}}$ forward invariant.*

Proof. First, note that the tangent cone of a closed set $\mathcal{A} \subset \mathbb{R}^n$ at any point y in the interior of that set is $\mathbf{T}_{\mathcal{A}}(y) = \mathbb{R}^n$, and by assumption $\mathcal{H}^{\mathcal{T}}$ is a closed set. Therefore, Nagumo's condition is automatically satisfied for all $(t, x) \in \text{int}(\mathcal{H}^{\mathcal{T}})$. Next, under the assumption that $|\frac{\partial h(t,x)}{\partial t}| + \|\frac{\partial h(t,x)}{\partial x}\| \neq 0$, the tangent cone of $\mathcal{H}^{\mathcal{T}}$ in (2.3) for the boundary $(t, x) \in \partial\mathcal{H}^{\mathcal{T}}$ is $\mathbf{T}_{\mathcal{H}^{\mathcal{T}}}(t, x) = \{(\dot{t}, \dot{x}) \in \mathbb{R} \times \mathbb{R}^n \mid \frac{\partial h(t,x)}{\partial t}\dot{t} + \frac{\partial h(t,x)}{\partial x}\dot{x} \leq 0\}$. By definition, $\dot{t} \equiv 1$, so any control law satisfying

(2.6) causes $(\dot{t}, \dot{x}) = (1, F(t, x, u(t, x)))$ to always lie inside the tangent cone, so by Lemma 2.7, the set $\mathcal{H}^{\mathcal{T}}$ is forward invariant. ■

Thus, one has a third set of conditions under which a CBF can be used to render a set forward invariant. Instead of requiring a Lipschitz condition as in Theorem 2.6, Theorem 2.8 instead requires that the gradient of the CBF does not vanish on the boundary of the CBF set. As illustrated by Theorem 2.2 and Theorem 2.6, this condition is not always necessary. However, without this condition, the tangent cone of $\mathcal{H}^{\mathcal{T}}$ is no longer given by the formula above, so Nagumo's theorem is more difficult to apply [194, Ch. 4.2]. The scenario where the gradient of the CBF vanishes on $\partial\mathcal{H}^{\mathcal{T}}$ is common in the following chapters, which is why I also presented Theorem 2.2 and Theorem 2.6. The observation that Nagumo's theorem is effectively only a condition on flows along the boundary of $\mathcal{H}^{\mathcal{T}}$ will also be useful later. Further variations of Theorems 2.2, 2.6, 2.8 are likely possible as well.

2.4 Quadratic Programs for Safety Filtering

Now suppose that the system (2.1) is control-affine,

$$\dot{x} = F(t, x, u) = f(t, x) + g(t, x)u \tag{2.8}$$

for functions $f : \mathcal{T} \times \mathcal{X} \to \mathbb{R}^n$ and $g : \mathcal{T} \times \mathcal{X} \to \mathbb{R}^{n \times m}$, and suppose that either $\mathcal{U} = \mathbb{R}^m$ or $\mathcal{U}$ is a polytope. That is, there exists a matrix $A_u \in \mathbb{R}^{q \times m}$ and a vector $b_u \in \mathbb{R}^q$ such that $\mathcal{U} \equiv \{u \in \mathbb{R}^m \mid A_u u \leq b_u\}$.

Given a system as above and a CBF h as in Definition 2.14, it is common (see also [66, Sec. II-C]) to write control laws of the form

$$\begin{aligned} u(t, x) = \underset{u \in \mathbb{R}^m}{\arg\min} \; & \|u - u_{\text{nom}}(t, x)\| \\ \text{such that } & \frac{\partial h(t, x)}{\partial t} + \frac{\partial h(t, x)}{\partial x}\Big[f(t, x) + g(t, x)u\Big] \leq \alpha(-h(t, x)) \\ & A_u u \leq b_u \end{aligned} \tag{2.9}$$

where $u_{\text{nom}} : \mathcal{T} \times \mathcal{X} \to \mathcal{U}$ is some other control law. I refer to (2.9) as a *safety filter*, or as a *QP-based control law*. This type of control law allows one to separately design a "performance control law" u_{nom}, also called the "nominal control law", without considering the safety constraints encoded in $\mathcal{S}$ and $\mathcal{H}$. The optimization (2.9) then modifies the nominal control law to create a control signal that is provably safe according to Theorems 2.2, 2.6, 2.8. Multiple CBFs can be combined by adding constraints to (2.9), as is done in Chapter 6. The control law (2.9) is often

called a "minimally modifying safety filter" or "minimally invasive safety filter", as it pointwise chooses the control input u satisfying (2.5),(2.6) that is closest (measured by the 2-norm) to the nominal control input u_{nom}. An alternative formulation of (2.9) sets $u_{\text{nom}}(t,x) \equiv 0$ and instead adds a Control Lyapunov Function (CLF) as an additional constraint [29], though in my experience, this type of control law is more difficult to design and tune.

2.4.1 Optimizer Choice

Note that the optimization (2.9) is a Quadratic Program (QP), i.e. an optimization problem with a strictly convex quadratic cost function and constraints that are affine in the optimization variable. One can rewrite (2.9) as

$$\begin{aligned} u(t,x) = \underset{u\in\mathbb{R}^m}{\arg\min}\ & \frac{1}{2}u^{\mathrm{T}}Hu + c^{\mathrm{T}}u \\ \text{such that}\ \ & A_c u \leq b_c \\ & A_u u \leq b_u \end{aligned} \tag{2.10}$$

where

$$H = I, \qquad c(t,x) = u_{\text{nom}}(t,x), \qquad A_c(t,x) = \frac{\partial h(t,x)}{\partial x} g(t,x),$$
$$b_c(t,x) = \alpha(-h(t,x)) - \frac{\partial h(t,x)}{\partial t} - \frac{\partial h(t,x)}{\partial x} f(t,x).$$

QPs can be solved extremely fast, on the order of 50 microseconds on a personal computer for a 4-dimensional QP, and thus are practical for real-time control. One can also compute explicit solutions to (2.9), though the explicit formulas can be very long. Most of the results in this dissertation were derived using either MATLAB's `quadprog` optimizer, or the Operator Splitting QP optimizer (OSQP) [196], though many other off-the-shelf optimizers exist. Compared to Model Predictive Control (MPC), one advantage of the control law (2.9) is that it can be solved so simply and quickly that one can use almost any convex optimization algorithm, whereas when using MPC, one often needs to devote more attention to the choice of optimizer. That said, when solving a problem with many CBFs (i.e. hundreds or thousands of CBFs), note that some algorithms take almost no extra time to handle lots of constraints (generally because few constraints are active simultaneously) whereas other algorithms increase in computation time each time a constraint is added. Note that if (2.1) is not control-affine, then one can still construct a safety filter similar to (2.9), but the filter will generally have nonlinear constraints that may require more specialized optimizers and take more time to compute.

While almost any convex optimization algorithm is capable of quickly solving (2.9), these algo-

rithms output only approximate solutions within some numerical tolerance, so the choice of optimizer and optimizer settings can still affect results. In particular, if $\boldsymbol{\mu}(t,x) = \{u \in \mathcal{U} \mid A_c(t,x)u \leq b_c(t,x)\}$ is the set of guaranteed-safe control inputs (recall that Theorems 2.2, 2.6, 2.8 are sufficient, not necessary, conditions), then whether the optimizer guarantees that $u(t,x) \in \boldsymbol{\mu}(t,x)$ exactly, or within some small tolerance, may impact results. Additionally, if $\boldsymbol{\mu}(t,x)$ is a very small set, then `quadprog` may return an error that the problem is infeasible. In this case, a helpful debugging tool is that MATLAB's `linprog` optimizer can provide a *certificate of feasibility/infeasibility*. In practice, it is best to construct CBFs with margin at all steps (i.e. so that small disturbances and computational errors are not catastrophic (see Chapter 4), and so that $\boldsymbol{\mu}$ is never too small).

2.4.2 Class-$\mathcal{K}_e$ Functions

One potential source of margin is the class-$\mathcal{K}_e$ function in (2.9). Note the following two observations:

1. Nagumo's theorem (Lemma 2.7) says that the only region that matters for safety is the boundary of the CBF set, yet the class-$\mathcal{K}_e$ function in (2.9) applies everywhere in the CBF set.

2. The class-$\mathcal{K}_e$ function α in Definition 2.14 is not unique. In particular, if Definition 2.14 holds for some $\alpha_1 \in \mathcal{K}_e$, then the definition also holds for any $\alpha_2 \in \mathcal{K}_e$ satisfying $\alpha_2(\lambda) \geq \alpha_1(\lambda)$ for all $\lambda \in \mathbb{R}$.

Thus, the engineer has freedom to choose the class-$\mathcal{K}_e$ function in (2.9), and that this function can be a useful tuning parameter. In my opinion, this is the greatest practical difference between BFs as in Definition 2.5 and CBFs as in Definition 2.14. When using BFs, Definition 2.5 only provides a flow condition on the boundary of $\mathcal{B}$. The existence of a class-$\mathcal{K}_e$ function as in Lemma 2.1 is a side-effect of the definition. When using CBFs, to create a continuous control law that achieves the flow condition specified in Lemma 2.7, one instead chooses among the infinite number of allowable class-$\mathcal{K}_e$ functions.

To see one effect of this parameter, suppose that $\alpha(\lambda) = \gamma\lambda$ for some $\gamma \in \mathbb{R}_{>0}$. Then any solution to (2.1) satisfying (2.5) obeys $h(t,x(t)) \leq h(t_0,x(t_0))e^{-\gamma(t-t_0)}$. Thus, there is a relation between the choice of $\alpha \in \mathcal{K}_e$ used in the control law (2.9) and the rate of change of the system, analogous to the bandwidth of a linear control law. As in linear control systems, the intuition that a high-bandwidth control law (high slope $\alpha \in \mathcal{K}_e$) will require a higher controller sampling frequency holds for CBFs as well. On the other hand, even if a function α as above with a small γ satisfies Definition 2.14, one may wish to choose a larger γ in order to permit the system state to evolve more quickly.

This dissertation provides a few remarks about the class-$\mathcal{K}_e$ function in Definition 2.14, mostly oriented towards removing the function and thus developing methods closer to Nagumo's necessary condition, but this is not a main topic of this work. Analyzing the effect of the class-$\mathcal{K}_e$ function further is an open area for future work. Moreover, finding methods to tune or vary the choice of class-$\mathcal{K}_e$ function to achieve certain results (instead of attempting to remove it) is an area of research that has recently gained attention, including [36, 149].

2.4.3 Continuity of Quadratic Programs

Assumption 2.2-A6 assumes that u is sufficiently regular so as to guarantee unique solutions, so regularity of the control law (2.9) is not a major topic of this dissertation. Thus, I only provide brief remarks on the regularity of QPs in this subsection and in sections where I intentionally add control law discontinuities. For theoretical reasons, Lipschitz continuity of (2.9) is often desired, so that the theorems in [71, Ch. 3] apply.

One way to prove/disprove Lipschitz continuity is to solve for the explicit, piecewise solution to (2.9) according to which constraints are active and inactive, e.g. [116]. Note that the work in [116] and others rely on a constraint qualification condition, such as the Mangasarian Fromowitz Constraint Qualification (MFCQ) condition [197, 198]. For a small number of constraints and no input bounds, the MFCQ assumption is reasonable. However, if $\mathcal{U}$ is a compact polytope, then the MFCQ conditions will rarely hold globally. At any corner of a polytope $\mathcal{U} \subset \mathbb{R}^m$, there will already be m active inequality constraints, so a CBF constraint $A_c u \leq b_c$ coinciding with a corner will cause $m+1$ active constraints, in which case MFCQ cannot hold. MFCQ also cannot hold if the gradient of the CBF $\frac{\partial h(t,x)}{\partial x}$ vanishes, as occurs with some of the methods in Chapters 3-4. A more general regularity condition is as follows.

Lemma 2.9. *For the control law* (2.9)*, suppose that there exists some set $\mathcal{D} \subseteq \mathcal{T} \times \mathcal{X}$ such that 1) $A_c : \mathcal{T} \times \mathcal{X} \to \mathbb{R}^{q \times m}$ and $b_c : \mathcal{T} \times \mathcal{X} \to \mathbb{R}^q$ are twice continuously differentiable on $\mathcal{D}$, and 2) for every $(t, x) \in \mathcal{D}$, there exists v such that $A_c(t,x)v < b_c(t,x)$ and $A_u v < b_u$. Then the control law* (2.9) *is locally Lipschitz continuous on $\mathcal{D}$.*

Proof. The proof follows immediately from [199, Thm. 2.1]. Assumption A1 and A2 in [199] are stated above, and Assumption A3 in [199] holds because $H = I$ is positive definite. ■

Continuity without Lipschitz guarantees is also established under more general conditions in [200, Thm. 3.1]. Other conditions for more specific problems have been studied in [29, 136, 201] and the erroneous result in [202] (corrected in [201]). Finally, CBFs can also be used with explicit control laws such as [28, 67], though this is not done in this dissertation.

2.4.4 Why Use Safety Filters

One alternative to CBFs is to plan a safe trajectory through $\mathcal{S}$ for all times in $\mathcal{T}$ and then follow that trajectory. If such a trajectory can be found, then one can skip the step of finding a viable subset of $\mathcal{S}$, as occurs implicitly when finding a CBF. Alternatively, one can plan a safe trajectory for a receding finite horizon and require that the the endpoint of the trajectory lie within some viable set. This idea is also the motivation behind a type of CBF called the Backup CBF [95, 96]. There exist many variations of trajectory planners, including some combinations between planners and CBFs. For this comparison, I consider MPC to also be a variant of trajectory planner. The advantages of safety filters over trajectory planners are 1) the reduced computation time, and 2) the ability to naturally handle nonlinear dynamics without approximation.

To see the reason for reduced computation time, consider the common problem of controlling a group of agents navigating in a space with obstacles. Each obstacle creates a "hole" in the complete state space that the agents are not allowed to enter. This results in $\mathcal{S}$ often being nonconvex. Trajectory planning algorithms often struggle with nonconvex specifications, or require problem-specific heuristics (e.g. "always go right") to navigate the nonconvexities. The two layer safety filtering (2.9) of the nominal control law u_{nom} bypasses these nonconvexities. In the first layer, the nominal control law can be a simple linear control law, a control law based on an approximate model, and/or the result of a separate optimization problem that does not take into account the safety constraints, and thus which is hopefully convex. In the second layer, the optimization (2.9) is also convex. Thus, a nonconvex trajectory planning problem may be replaced by two convex optimization problems with exact safety guarantees (i.e. no approximations in the safety layer). Moreover, the safety filter (2.9) has only affine constraints even when the system (2.8) is nonlinear in the state (t, x), whereas a trajectory planner must either optimize over the full nonlinear dynamics or add margins for any approximations made.

A summary of the key differences between CBFs and trajectory planning is shown in Table 2.1. Note that this is only a general comparison, as I am not referring to any specific trajectory planning algorithm. In addition to the convexity observations, note that the safety filter (2.9) only considers the current control input, of dimension m. In contrast, a trajectory planner computes a horizon of of N control inputs, which leads to an optimization problem of dimension mN (or possibly $(m+n)N$ if the states are also optimization variables) that is more computationally expensive. Thus, the safety filtering approach (2.9) converts a high dimension optimization problem solved at the start of the trajectory (and possibly recomputed periodically throughout the trajectory as in MPC) to two convex optimization problems that are solved continuously throughout the trajectory. This sort of continuously running optimization fits naturally into a real-time control system, whereas trajectory planning is often implemented in a multi-rate architecture. That said, CBFs can also fit into a multi-rate control architecture [141], as the options for variations of filters and planners are

Trajectory Planning	Safety Filtering
Performance and safety are coupled	Performance and safety are decoupled
Requires exact model or margins for approximations	Requires exact model only in the safety layer
Optimization dimension mN or $(m+n)N$	Optimization dimension m
Generally nonconvex optimization with nonlinear constraints	A convex safety filter and a simple nominal control law
Yields true optimal solution	Trajectories may be inefficient
Runs once before start, possibly recomputed	Runs continuously
Can make decisions based on future predictions	Can only make decisions based on the present state and environment

Table 2.1: Comparison of Trajectory Planning and CBFs as Safety Filters

as infinite as the number of open control problems.

Compared to linear control laws, CBFs and safety filters may also simplify certain aspects of control design. For one, CBFs often provide a more natural language to express and verify requirements. Often, requirements are phrased as tolerances, which can be equivalently phrased as a permissible set of states. With linear trajectory tracking methods, an engineer has to find a nominal trajectory that is always inside the set of permissible states and then verify that the real trajectory under perturbations indeed remains within that set. By contrast, the methods that follow allow one to directly encode sets as constraints, and build CBFs that by construction render the state trajectories always in these sets, rather than iteratively designing a control law and checking for constraint satisfaction. This also enables one to use very simple nominal control laws in (2.9) rather than going to the effort of computing a safe trajectory a priori.

While CBFs as safety filters are a powerful control tool, this approach also has disadvantages. The most obvious at time of writing is the need for a highly educated engineer to implement such filters. The tuning of class-$\mathcal{K}_e$ functions is also less intuitive than tuning in linear control laws. Finding a CBF and CBF viable set may also be challenging, akin to the challenge of finding a Lyapunov function. Next, while a trajectory planner provides an optimal solution over its prediction horizon, safety filters are only pointwise-optimal and thus may yield inefficient trajectories. The authors in [144] showed that the control law (2.9) is the solution to a particular optimal control problem, but in general, this control law will not yield the trajectory that is optimal in fuel, time, or other common optimality metrics.

Because of the lack of prediction horizon, CBFs may also result in deadlocks, where either a

single agent gets stuck in front of an obstacle, or multiple agents all come to a stop and are unable to navigate around each other. Deadlocks may occur when using either centralized or decentralized control laws [36]. Such points are analogous to the phenomenon of local minima in Artificial Potential Fields (APFs), and a relation between APFs and CBFs is discussed in [38]. Also similar to APFs, trajectories under (2.9) may come to a near-stop near an obstacle before slowly navigating around the obstacle, whereas a trajectory planner with a sufficiently large time horizon should be able to navigate around an obstacle without stopping. That said, in a cluttered environment with many obstacles, a trajectory planner may be just as conservative as the safety filter (2.9) because of the need for the trajectory to end inside a (likely conservative) viable set.

As with trajectory planners and most other methodologies, there are variations of the core CBF method illustrated above that handle some of the well-known disadvantages of this approach. Overall, CBFs often provide a "middle-ground" between simplicity and performance. Some of the above phenomena also serve as motivations for the improvements in the following chapters.

CHAPTER 3

High Relative Degree Constraint Functions

In this chapter, I introduce the "high relative-degree problem", formalized as Problem 3.1. In this problem, the set of instantaneously safe states $\mathcal{S}$ takes the form of a sublevel set of a function, called the *constraint function*, that is of "high relative-degree with respect to the system dynamics", defined precisely in Definition 3.2. This chapter then proposes two general methods to find a CBF and a viable subset of $\mathcal{S}$ for arbitrary relative-degrees, and compares the methods in simulation using the controller (2.9). The preliminary analysis of the "high relative-degree problem" in this chapter then serves as an introduction to the "robust high relative-degree problem" in Chapter 4, where I will expand upon the two methods presented here and introduce a third related method. Note that this chapter constitutes earlier work within this dissertation and thus contains more references to older works, in particular works written before the HOCBF formulation (a related approach discussed further in Chapter 4) became standardized in [88, 89].

3.1 Introduction

Control Barrier Functions (CBFs) have recently gained popularity across disciplines for control synthesis in safety-critical systems. However, the problem of designing CBFs for high relative-degree ($r \geq 2$) constraint functions under control input constraints remains an open question except for specific systems [28, 29, 36, 82]. In this chapter, I develop two methods of designing CBFs that generalize the results for specific systems in [28, 29, 36, 82], and in particular the methods in [82, 96, 203], to a wider class of systems, namely systems that meet the assumptions of Theorems 3.4-3.5.

As in Chapter 2, the objective of this chapter is to design a control law that always renders systems inside some set of instantaneously safe states $\mathcal{S}$, henceforth termed the *safe set*. Here and in subsequent chapters, the safe set $\mathcal{S}$ is specified as the zero sublevel set of some *constraint function*. If the constraint function is of relative-degree $r = 1$ with respect to the system dynamics, then the set can be rendered forward invariant if the constraint function is also a CBF [29]. For

control-affine dynamics as in (2.8), this leads to an affine condition on the control input, which can be applied pointwise to yield either explicit control laws as in [28], or optimization-based control laws as in [29, 84, 85, 87] and (2.9) that render the safe set forward invariant. However, if the constraint function is of relative-degree $r \geq 2$ and the uncontrolled dynamics allow trajectories to leave the safe set (i.e. if $\mathcal{S}$ is not forward invariant under $u \equiv 0$), then the constraint function alone cannot be a CBF. To recover such a control-affine condition, this chapter constructs CBFs composed of both the constraint function and its derivatives, and designed such that the CBF viable set is a subset of the instantaneously safe set.

Several papers develop methods to convert high relative-degree constraint functions ($r \geq 2$) into CBFs, including using compositions with bounded monotonic functions [29], backstepping [84], feedback linearization and pole placement [85, 138], or by defining allowable sets for every order of derivative of the constraint function [87]. The approach in [203] bypasses the creation of a new CBF, but develops a condition on the lowest order controllable derivative that fills the same role as the conditions in [84, 85]. All these approaches lead to conditions on the control input that are potentially infeasible if the set of valid control inputs is bounded (i.e. (2.9) might become infeasible). In practice, input constraints may be satisfied within the frameworks of [29, 84, 85, 138] for certain trajectories by tuning (e.g. choosing different poles using the method in [85]), but these approaches only yield provably feasible control laws if the control set is $\mathbb{R}^m$. The work in [87] improves upon this by defining a subset of the safe set that is controlled invariant in the presence of input constraints. However, choosing appropriate class-$\mathcal{K}$ functions to satisfy the feasibility requirements in [87, Def. 7] may not be straightforward.

Conditions for safety under input constraints for the n-integrator system are introduced in [82]. For certain other systems, a second CBF that guarantees satisfaction of control input constraints for all future times can also be introduced [29, 36]. For more general systems, [94, 96] recently developed an approach (not specific to high relative-degree) wherein a small known *backup set* is expanded to the set of states which can reach the backup set in a finite time horizon under input constraints. The work in [86] is similar to [94, 96], but generalizes the approach to infinite time horizon.

Compared to [28, 29, 36, 82, 94, 96], this chapter addresses the problem of designing CBFs for high relative-degree constraint functions for a broader class of systems using two extensions to the approach in [86]. The first strategy employs a predefined nominal control law and tests whether the system would remain inside the safe set under this control law, similar to [86, 94, 96]. However, this strategy does not require a predefined backup set as in [94, 96], and does not require that the searched time horizon contain a unique maximizer of the constraint function as in [86]. The second strategy simplifies the first and generalizes [82] to any system for which there exists a minimum control authority over the rth derivative of the constraint function everywhere in the

safe set. Similar to [87], the resultant controlled invariant sets are subsets of the safe set, and may be smaller or larger than the corresponding sets obtained in [87]. However, unlike [87], these methods by construction respect input constraints without tuning other parameters of the CBF. After introducing the theory behind these methods, I then compare both strategies in simulation on an obstacle avoidance problem for a double-integrator system, and demonstrate the second strategy on a safety-critical spacecraft control problem for asteroid observation.

3.2 Preliminaries

3.2.1 Notations

In addition to the notations in Section 2.1, for a vector $v \in \mathbb{R}^n$, let $v^\perp = \{w \in \mathbb{R}^n : v^\mathrm{T} w = 0\}$. For a continuously differentiable function $\kappa : \mathbb{R}^n \to \mathbb{R}$, let $L_f\kappa(x)$ denote the Lie-derivative of κ with respect to a function $f : \mathbb{R}^n \to \mathbb{R}^{n\times m}$ at the point x, that is $L_f\kappa(x) \triangleq \frac{\partial \kappa(x)}{\partial x} f(x)$. For sufficiently differentiable κ and f, let $L_f^2\kappa(x) = L_f(L_f\kappa(x))$ and $L_f^r\kappa(x) = L_f(L_f^{r-1}\kappa(x))$. The Lie-derivative notation is an efficient way to denote high order derivatives and was common in the early CBF literature, but I use this notation limitedly in the rest of this dissertation because it is not as helpful when κ is a function of multiple variables. Let $\dot\kappa$ denote the total derivative of κ along some system dynamics, explained further in Section 3.2.2. Let $\ddot\kappa$ denote the second total derivative and $\kappa^{(r)}$ denote the rth total derivative. Define the directional derivative of a function $h : \mathbb{R}^n \to \mathbb{R}^m$ in direction $v \in \mathbb{R}^n$ as

$$\partial^v h(x) \triangleq \lim_{\delta \to 0^+} \frac{h(x + v\delta) - h(x)}{\delta}\,. \tag{3.1}$$

Definition 3.1 (Directionally Differentiable Function)**.** *A continuous function $h : \mathcal{X} \to \mathbb{R}^m$ with $\mathcal{X} \subseteq \mathbb{R}^n$ is* directionally differentiable *if the directional derivative* (3.1) *of h exists for all $x \in \mathcal{X}$ and all $v \in \mathbb{R}^n$.*

3.2.2 Model and Problem Formulation

This chapter works with the time-invariant system

$$\dot x = f(x) + g(x)u\,, \tag{3.2}$$

with state $x \in \mathcal{X} \subseteq \mathbb{R}^n$, control input $u \in \mathcal{U} \subset \mathbb{R}^m$ where $\mathcal{U}$ is compact, and functions $f : \mathcal{X} \to \mathbb{R}^n, g : \mathcal{X} \to \mathbb{R}^{n\times m}$ where f, g are r-times continuously differentiable. The first total derivative of a function $\kappa : \mathcal{X} \to \mathbb{R}$ along (3.2) is $\dot\kappa(x, u) \triangleq L_f\kappa(x) + L_g\kappa(x)u$.

Definition 3.2 (Relative Degree)**.** *A function $\kappa : \mathcal{X} \to \mathbb{R}$ is said to be of* relative-degree r *with respect to the dynamics* (3.2) *if*

1. *κ is r-times continuously differentiable,*
2. *$L_g L_f^k \kappa(x) \equiv 0,\ \forall x \in \mathcal{X},\ \forall k = 0, 1, \cdots r-2$, and*
3. *$\exists \mathcal{C} \subseteq \mathcal{X}, \mathcal{C} \neq \emptyset$ such that $L_g L_f^{r-1} \kappa(x) \neq 0, \forall x \in \mathcal{C}$.*

Denote the space of all functions of relative-degree r with respect to a given system as $\mathcal{G}^r$. For a function $\kappa \in \mathcal{G}^r$ for $r \geq 2$, it follows that $L_g \kappa(x) \equiv 0$, so $\dot{\kappa}(x, u) \equiv \dot{\kappa}(x) = L_f \kappa(x)$. Thus, even without specifying a control law, one can derive higher order derivatives such as $\ddot{\kappa}(x, u) = L_f^2 \kappa(x) + L_g L_f \kappa(x) u$. For readability, total derivatives above second order are denoted $\kappa^{(r)}$.

Let $\kappa : \mathcal{X} \to \mathbb{R}$ satisfy $\kappa \in \mathcal{G}^r$, and let $\mathcal{S}$ in this chapter be given as

$$\mathcal{S} \triangleq \{x \in \mathbb{R}^n \mid \kappa(x) \leq 0\} \tag{3.3}$$

where $\mathcal{S}$ is time-invariant. I refer to κ as the *constraint function* for set $\mathcal{S}$.

If $\kappa \in \mathcal{G}^1$, then a sufficient condition for controlled invariance of $\mathcal{S}$ is that κ is a CBF as in Definition 2.14, i.e. that there exists a set $\mathcal{D} \subseteq \mathcal{X}$ satisfying $\mathcal{S} \subseteq \mathcal{D}$ and, in the notation of this chapter, such that

$$\inf_{u \in \mathcal{U}} \left[L_f \kappa(x) + L_g \kappa(x) u \right] \leq \alpha(-\kappa(x)), \forall x \in \mathcal{D}\,. \tag{3.4}$$

Note that the infimum above can be replaced by a minimum since $\mathcal{U}$ is assumed compact in this chapter.

However, if instead $\kappa \in \mathcal{G}^r$ for $r \geq 2$, then $L_g \kappa(x) u \equiv 0$, so the infimum in (3.4) disappears and (3.4) becomes

$$L_f \kappa(x) \leq \alpha(-\kappa(x)), \forall x \in \mathcal{D}\,. \tag{3.5}$$

The condition (3.5) is not useful for control design, so going forward I assume that κ is not a CBF and instead seek to design a new CBF h. Thus, the objective of this chapter is as follows.

Problem 3.1. *Given a constraint function $\kappa \in \mathcal{G}^r$ for $r \geq 2$, such that there exists at least one $x_0 \in \partial\mathcal{S}$ where $L_f \kappa(x_0) > 0$, and a compact control set $\mathcal{U}$, develop functions $h : \mathcal{X} \to \mathbb{R}$ such that $\mathcal{H} = \{x \in \mathbb{R}^n \mid h(x) \leq 0\}$ is a subset of $\mathcal{S}$, and h is a CBF.*

A solution h to Problem 3.1 defines a subset $\mathcal{H}$ of the safe set that is controlled invariant. Problem 3.1 can be addressed by methods such as those in [87] with a proper selection of $\alpha_i \in \mathcal{K}, i = 1, 2, \cdots, r$ (or without input constraints for any $\alpha_i \in \mathcal{K}$). In contrast, the approaches in this chapter always satisfy the input constraints by construction.

Lastly, similar to [82, 86], define the flow operator $\psi_\kappa(t; x, u)$ for $t \geq 0$ as the value $\kappa(y(t))$ resulting from the initial value problem $\dot{y} = f(y) + g(y)u, y(0) = x$ under the control law u. Also, let $\psi_x(t; x, u)$ denote the value of the state $y(t)$ according to the same initial value problem.

3.2.3 Directionally Differentiable Control Barrier Functions

To address Problem 3.1, this chapter considers the possibility of a CBF that is continuous, but not necessarily continuously differentiable. This generalization of CBF takes inspiration from the relaxed differentiability conditions in [204], though the conditions used here differ slightly from [204] because the use of the open set V in [204, Def. 2.1] is too restrictive. Even more general conditions may be obtained from [150, 151]. Motivated by [204, Def. 2.3], consider the slightly generalized definition of CBF as follows.

Definition 3.3 (Control Barrier Function (Directionally Differentiable))**.** *Let $h : \mathcal{X} \to \mathbb{R}$ be directionally differentiable as in Definition 3.1 and let*

$$\mathcal{H} \triangleq \{x \in \mathcal{X} \mid h(x) \leq 0\}. \tag{3.6}$$

The function h is a Control Barrier Function (CBF) *for the system* (3.2) *if there exists a set $\mathcal{D} \subseteq \mathcal{X}$ such that $\mathcal{H} \subseteq \mathcal{D}$ and a function $\alpha \in \mathcal{K}_e$ such that*

$$\inf_{u \in \mathcal{U}} \partial^{f(x)+g(x)u} h(x) \leq \alpha(-h(x)), \forall x \in \mathcal{D}. \tag{3.7}$$

Definition 2.14 and Definition 3.3 are so similar that I do not introduce a new term for Definition 3.3. This definition is useful because of the following observation.

Lemma 3.1. *Let $\{h_i\}_{i=1}^N$ be a finite collection of continuously differentiable functions $h_i : \mathcal{X} \to \mathbb{R}$ and define $h : \mathcal{X} \to \mathbb{R}$ as $h(x) = \max_{i \in [N]} h_i(x)$. If there exists a set $\mathcal{D} \subseteq \mathcal{X}$ such that $\mathcal{H} \subseteq \mathcal{D}$ in* (3.6) *and a function $\alpha \in \mathcal{K}_e$ such that*

$$\inf_{u \in \mathcal{U}} \max_{i \in \mathcal{I}(x)} \left[L_f h_i(x) + L_g h_i(x) u\right] \leq \alpha(-h(x)), \forall x \in \mathcal{D}, \tag{3.8}$$

where $\mathcal{I}(x) \triangleq \{i \in [N] \mid h(x) = h_i(x)\}$, then h is a CBF as in Definition 3.3.

Proof. Since N is finite, the function h as above is directionally differentiable and its derivative is $\partial^v h(x) = \max_{i \in \mathcal{I}(x)} [\partial^v h_i(x)]$ by [205, Thm. 3.2]. It follows that $\partial^{f(x)+g(x)u} h(x) \equiv \max_{i \in \mathcal{I}(x)} \partial^{f(x)+g(x)u} h_i(x) \equiv \max_{i \in \mathcal{I}(x)} \left[L_f h_i(x) + L_g h_i(x) u\right]$, so (3.8) implies (3.7). ■

Forward invariance of $\mathcal{H}$ is then guaranteed as follows.

Theorem 3.2. *Given a CBF $h : \mathcal{X} \to \mathbb{R}$ for the system* (3.2) *as in Definition 3.3, let $\mathcal{D}$ be an open set satisfying $\mathcal{H} \subset \mathcal{D}$ and let $\alpha \in \mathcal{K}_e$. Then any control law $u : \mathcal{X} \to \mathcal{U}$ satisfying*

$$\partial^{f(x)+g(x)u(x)} h(x) \leq \alpha(-h(x)), \forall x \in \mathcal{D} \tag{3.9}$$

will render $\mathcal{H}$ in (3.6) *forward invariant as in Definition 2.6.*

Proof. The proof follows the same logic as Theorem 2.2. Note that the comparison lemma [191, Lemma IX.2.6] applies for any type of derivative. Let $\theta(t) = h(x(t))$ and let $\dot{\theta}(t) = \lim_{\delta \to 0^+} \frac{\delta(t+\delta)-\delta(t)}{\delta}$ denote the directional derivative in forward time. Then $\dot{\theta}(t) = \partial^{f(x(t))+g(x(t))u(x(t))} h(x(t)) \leq \alpha(-\theta(t))$, and the result follows by the same logic as Theorem 2.2. ■

Recall that Theorem 2.6 was shown to be a special case of Theorem 2.2, so it follows that one can extend Theorem 2.6 to CBFs as in Definition 3.3 as well. I also include an extension of Theorem 2.8 as follows, because the following result is illustrative of the usefulness of Nagumo's theorem.

Theorem 3.3. *Suppose that h is a CBF as constructed in Lemma 3.1, and suppose that $\| \frac{\partial h_i(x)}{\partial x} \| \neq 0$ for all $i \in \mathcal{I}(x)$ and all $x \in \partial\mathcal{H}$, and let $\alpha \in \mathcal{K}$. Then any control law $u : \mathcal{X} \to \mathcal{U}$ satisfying*

$$\max_{i \in \mathcal{I}(x)} \left[L_f h_i(x) + L_g h_i(x) u(x) \right] \leq \alpha(-h(x)), \forall x \in \mathcal{H} \tag{3.10}$$

will render $\mathcal{H}$ in (3.6) *forward invariant as in Definition 2.6.*

Proof. Since $h(x) = \max_{i \in [N]} h_i(x)$, it follows that $\mathcal{H} = \cap_{i=1}^N \mathcal{H}_i$ and the tangent cone of $\mathcal{H}$ is $\mathbf{T}_{\mathcal{H}}(x) = \cap_{i=1}^N \mathbf{T}_{\mathcal{H}_i}(x)$. Since $\frac{\partial h_i(x)}{\partial x}$ does not vanish for $x \in \partial\mathcal{H}$ for any active $i \in \mathcal{I}(x)$, it follows from the same argument as in Theorem 2.8 that any control law satisfying (3.10) will cause $\dot{x}$ to lie inside the tangent cone $\mathbf{T}_{\mathcal{H}}(x)$ and thus render $\mathcal{H}$ forward invariant. ■

Thus, the continuously differentiable notion of CBFs may be easily expanded to more complicated functions h. Again, recall the critical Assumption 2.2-A6 that solutions to (3.2) are unique (see [195] for a discussion of what may happen if there are multiple solutions). Note also that the maximum over a finite index set as in (3.10) can be encoded in a QP-based control law as in (2.9) by enforcing $L_f h_i(x) + L_g h_i(x) u \leq \alpha(-h(x))$ simultaneously as separate constraints on the QP for every $i \in \mathcal{I}(x)$.

3.3 Two Methods to Derive CBFs for High Relative-Degree Constraint Functions

3.3.1 General Case (Backup CBF)

For $\kappa \in \mathcal{G}^r$, I refer collectively to the derivatives $\dot{\kappa}, \ddot{\kappa}, \cdots \kappa^{(r-1)}$ which are not explicit functions of u as *generalized inertia*, by analogy to inertia in kinematic systems. To ensure safety when $r \geq 2$, a controller must be able to dissipate this generalized inertia before leaving the safe set, i.e. must be able to ensure that $\kappa^{(k)}(x) \leq 0, \forall k \leq r$ for all x such that $\kappa(x) = 0$.

The approach in this chapter is to examine the system response forward in time according to its generalized inertia. Suppose that $\kappa(x) < 0$ and $\kappa^{(k)}(x) > 0$ for one or more $k \in \{1, 2, \cdots r\}$. Then this chapter seeks to determine how large each $\kappa^{(k)}$ can be allowed to grow before there is no allowable control input under which the trajectory stays within the safe set at some future time. However, analyzing all possible future trajectories (i.e. all possible control laws) for safety is intractable, so instead suppose that one has a predefined control law $u^* : \mathcal{X} \to \mathcal{U}$ that attempts to drive the state towards the interior of $\mathcal{S}$ (called a "nominal evading maneuver" in [86]). For example, if $\mathcal{U}$ is a closed ball, one might choose

$$u^*_{\text{ball}}(x) \triangleq \arg\min_{u \in \mathcal{U}} L_g L_f^{r-1} \kappa(x) u \,, \tag{3.11}$$

which pointwise minimizes $\kappa^{(r)}$.

Assumption 3.1. *For any $x(0) \in \mathcal{S}$, assume that the system* (3.2) *admits a unique solution for all $t \geq 0$ under the control law $u^*(x)$.*

Denote the evolution of κ under u^* from initial condition $x(0)$ as $\psi_\kappa(t; x(0), u^*)$. Then there exists at least one safe and feasible trajectory from $x(0)$ if $\psi_\kappa(t; x(0), u^*) \leq 0, \forall t \geq 0$. Thus, I introduce a new function,

$$h^*(x) \triangleq \sup_{t \geq 0} \psi_\kappa(t; x, u^*) \,, \tag{3.12}$$

which I seek to render nonpositive (see also [86, Eq. 14]).

Assumption 3.2. *For all $x \in \mathcal{S}$, assume that $h^*(x)$ in* (3.12) *exists and is finite and directionally differentiable.*

That is, Assumption 3.2 assumes that the trajectories under u^* dissipate the generalized inertia so that ψ_κ is upper bounded (and not infinite), and that this upper bound is regular with respect to x.

The relationship between κ, ψ_κ, and h^* is visualized in Fig. 3.1. In this example, $\kappa(x(0)) < 0$, but there exist $t > 0$ such that $\kappa(x(t)) > \kappa(x(0))$ under the control law u^*. The point of interest in

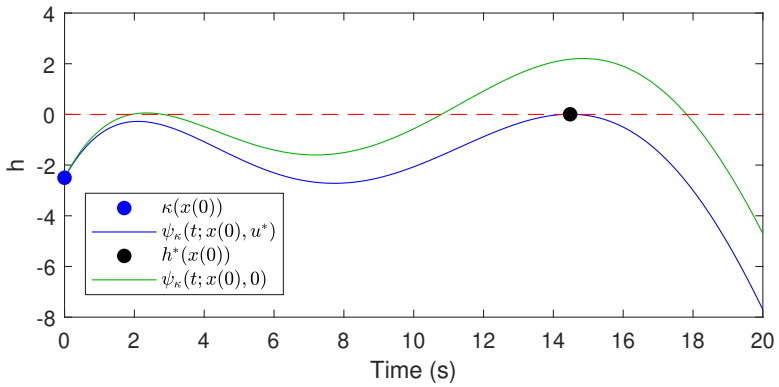

Figure 3.1: Explanation of Backup Control Law. A potential trajectory of $\psi_\kappa(t; x, u^*)$ under control input $u = u^*(x)$, and unforced evolution $\psi_\kappa(t; x, 0)$ under $u = 0$ for comparison. The method in this section is concerned with whether the point $h^*(x)$ exceeds 0, which would imply that under the proposed backup controller u^*, there exists t such that $\kappa(x(t)) > 0$.

Fig. 3.1 is the black dot $h^*(x(0))$, because $h^*(x(0)) > 0$ would imply that there exists $t > 0$ such that $\kappa(x(t)) > 0$ under u^*, which means u^* does not render the system safe from $x(0)$.

If the supremum in (3.12) is achieved for finite t, denote the set of time instances that maximize ψ_κ as

$$t_c^*(x) \triangleq \left\{ \arg\max_{t \geq 0} \psi_\kappa(t; x, u^*) \right\}, \tag{3.13}$$

so that (3.12) can be equivalently written as

$$h^*(x) = \psi_\kappa(t_{c,0}; x, u^*), \ \forall t_{c,0} \in t_c^*(x)\,. \tag{3.14}$$

Note that the set $t_c^*(x)$ may contain more than one element. One can see this visually for the example in Fig. 3.1, for which a slight perturbation might result in the blue line having two maximizers.

Assumption 3.3. *For all $x \in \mathcal{S}$, assume that $t_c^*(x)$ in* (3.13) *has finitely many elements.*

Finally, define the set rendered safe by u^* as

$$\mathcal{H}^* \triangleq \{x \in \mathbb{R}^n \mid h^*(x) \leq 0\}\,. \tag{3.15}$$

By definition, $h^*(x) \geq \kappa(x), \forall x \in \mathbb{R}^n$, so $\mathcal{H}^* \subseteq \mathcal{S}$.

For brevity, in this section let $\psi_\kappa(t) = \psi_\kappa(t; x, u^*)$ and $\psi_x(t) = \psi_x(t; x, u^*)$. I am now ready to state the first main result of this chapter.

Theorem 3.4. *The function h^* in* (3.12) *is a CBF for the control set $\mathcal{U}$, provided $\mathcal{H}^* \neq \emptyset$ in* (3.15).

Proof. The function h^* is a CBF if it meets the condition (3.4) or the extensions (3.7)-(3.8). This proof considers three cases depending on the elements of $t_c^*(x)$.

First, consider if $0 \in t_c^*(x)$. Then it must hold that $\dot{\kappa}(x) \leq 0$ (where $\dot{\kappa}$ is independent of u if $\kappa \in \mathcal{G}^r$ for $r \geq 2$) for 0 to be a maximizer of $\psi_\kappa(t)$. By definition, $\psi_\kappa(0) = \kappa(x)$ and thus $L_f\psi_\kappa(0) + L_g\psi_\kappa(0)u = L_f\kappa(x) + L_g\kappa(x)u = \dot{\kappa}(x) \leq 0$. Thus, condition (3.4) is satisfied for any u if $\kappa \in \mathcal{G}^r, r \geq 2$, and for the input u^* if $\kappa \in \mathcal{G}^1$.

Next, consider if $t_{c,0} \in t_c^*(x)$ where $t_{c,0} > 0$. Then $t_{c,0}$ is a maximizer on an open interval, so a necessary condition is $\frac{\partial \psi_\kappa(t)}{\partial t}\big|_{t=t_{c,0}} = 0$. Define $\omega \triangleq \psi_\kappa(t_{c,0}; x, u^*)$. It follows that

$$\omega = \psi_\kappa(t_{c,0}; x, u^*) = \psi_\kappa(t_{c,0} - \tau; \psi_x(\tau; x, u^*), u^*)\,, \tag{3.16}$$

for any $\tau \in [0, t_{c,0}]$. Since ω is constant with respect to τ, its derivative satisfies

$$\frac{d}{d\tau}[\omega] = \frac{\partial \psi_\kappa(t_{c,0} - \tau; \psi_x(\tau), u^*)}{\partial(t_{c,0} - \tau)} \frac{\partial(t_{c,0} - \tau)}{\partial \tau} + \frac{\partial \psi_\kappa(t_{c,0} - \tau; \psi_x(\tau), u^*)}{\partial \psi_x(\tau)} \frac{\partial \psi_x(\tau)}{\partial \tau} \tag{3.17}$$

$$\begin{aligned} \implies 0 = &-\frac{\partial \psi_\kappa(t_{c,0} - \tau; \psi_x(\tau), u^*)}{\partial(t_{c,0} - \tau)} \\ &+ \frac{\partial \psi_\kappa(t_{c,0} - \tau; \psi_x(\tau), u^*)}{\partial \psi_x(\tau)} (f(\psi_x(\tau)) + g(\psi_x(\tau))u^*(\psi_x(\tau))\,. \end{aligned} \tag{3.18}$$

At $\tau = 0$, it further holds that $\frac{\partial \psi_\kappa(t_{c,0}-\tau)}{\partial(t_{c,0}-\tau)}\big|_{\tau=0} = \frac{\partial \psi_\kappa(t)}{\partial t}\big|_{t=t_{c,0}} = 0$, and $\psi_x(0) = x$, so substituting $\tau = 0$ into (3.18) yields

$$0 = L_{f(x)}\psi_\kappa(t_{c,0}; x, u^*) + L_{g(x)}\psi_\kappa(t_{c,0}; x, u^*)u^*(x)\,. \tag{3.19}$$

Thus, the input u^*, which by definition is always in $\mathcal{U}$, renders $L_f\psi_\kappa(t_{c,0}) + L_g\psi_\kappa(t_{c,0})u = 0$, thereby satisfying condition (3.4).

Finally, if the supremum in (3.12) is not achieved for finite t (i.e. (3.13) does not exist), then choose $\omega = \lim_{t\to\infty} \psi_\kappa(t - \tau; \psi_x(\tau; x, u^*), u^*)$. By Assumption 3.2, the limit exists, so $\lim_{t\to\infty} \frac{\partial \psi_\kappa(t)}{\partial t} = 0$. Differentiating ω with respect to τ as in the prior case yields that (3.19) holds in the limit as $t_{c,0} \to \infty$, so under the input u^*, condition (3.4) is still satisfied. As an abuse of notation, I denote this case as $\infty \in t_c(x)$.

Since t_c may contain multiple elements (but finitely many by Assumption 3.3), the trajectory of h^* satisfies $\dot{h}^*(x, u) = \max_{t_{c,0} \in t_c^*(x)} [L_f\psi_\kappa(t_{c,0}) + L_g\psi_\kappa(t_{c,0})u]$ by [205, Thm. 3.2]. The control law u^* is independent of $t_{c,0}$, so the above cases show that there exists a single control law $u : \mathcal{X} \to \mathcal{U}$ given by $u(x) = u^*(x), \forall x \in \mathcal{X}$ that always maps to the set of allowable control inputs $\mathcal{U}$ and that renders $L_f\psi_\kappa(t_{c,0}) + L_g\psi_\kappa(t_{c,0})u$ nonpositive for every element $t_{c,0} \in t_c^*(x)$. Thus, the control law u^* will also render the maximum of these flows nonpositive, so that $\dot{h}^*(x, u(x)) \leq 0 \leq \alpha(-h^*(x))$ for all $x \in \mathcal{S}$ and for any choice of $\alpha \in \mathcal{K}_e$. By Lemma 3.1, h^* in (3.12) is a CBF as in Definition 3.3. ■

The immediate consequence of Theorem 3.4 is that if $x(0) \in \mathcal{H}^*$, then there is a controller such that $x(t) \in \mathcal{H}^*, \forall t \geq 0$, and by extension $x(t) \in \mathcal{S}, \forall t \geq 0$ since $\mathcal{H}^* \subseteq \mathcal{S}$. Specifically, Theorem 3.4 implies that if $x(0) \in \mathcal{H}^*$, then there exists at least one safe trajectory. Unlike in [86], Theorem 3.4 also allows for the possibility of ψ_κ having multiple maximizers. The following remark then provides a means to calculate $\dot{h}^*$ for general systems, which I then apply as a condition on the control input using (3.9)-(3.10).

Remark 3.1. *Suppose that the control law $u^* : \mathcal{X} \to \mathcal{U}$ is continuously differentiable. For a control input $v \in \mathcal{U}$, $\dot{h}^*$ is given by*

$$\dot{h}^*(x, v) = \max_{t_{c,0} \in t_c^*(x)} \left(\frac{\partial \psi_\kappa(t_{c,0}; x, u^*)}{\partial x} \left(f(x) + g(x)v \right) \right). \tag{3.20}$$

The gradient of $\psi_\kappa(t_{c,0}; x, u^*)$ with respect to x at a particular $t_{c,0}$ is given by

$$\frac{\partial \psi_\kappa(t_{c,0}; x, u^*)}{\partial x} = \left. \frac{\partial \kappa(x)}{\partial x} \right|_{x = \psi_x(t_{c,0}; x, u^*)} \Theta(t_{c,0}) \tag{3.21}$$

where matrix $\Theta(t)$ is the solution of the initial value problem

$$\begin{aligned} \dot{\Theta} &= \frac{\partial}{\partial y} [f(y) + g(y)u^*(y)] \big|_{y=z} \Theta, \qquad \Theta(0) = I \\ \dot{z} &= f(z) + g(z)u^*(z), \qquad\qquad z(0) = x \end{aligned} \tag{3.22}$$

where I is the identity matrix.

Note that the maximum of (3.20) is generally not known until v is selected. Thus, the consequence of Remark 3.1 is that a control law u of the form (2.9) should be encoded so as to satisfy the condition $L_f \psi_\kappa(t_{c,0}) + L_g \psi_\kappa(t_{c,0}) u \leq \alpha(-h^*(x))$ once for each element $t_{c,0} \in t_c^*(x)$ (i.e. one additional safety constraint in the QP for each maximizer in t_c^*). Note that u^* also must be differentiable for (3.22) to be well-defined, though the case study in Section 3.4 shows how differentiability of u^* almost everywhere is often sufficient in practice.

While Theorem 3.4 theoretically applies to any system for which a nonempty $\mathcal{H}^*$ exists, in practice, it may be limited by the requirement to propose a "good" u^* (i.e. one which yields a large set $\mathcal{H}^*$). Also, for most systems, h^* and $\frac{\partial \psi_\kappa}{\partial x}$ will not have explicit expressions. That said, computing h^* and $\frac{\partial \psi_\kappa}{\partial x}$ only requires propagating two Ordinary Differential Equations (ODEs), which can be done efficiently. On the other hand, the advantage of this approach is that if $h^*(x(0)) \leq 0$, then one immediately knows $\mathcal{H}^*$ can be rendered forward invariant under the input constraints.

Next, I present an alternative approach that avoids the complexity of (3.20)-(3.22) but yields a different, usually smaller, CBF set $\mathcal{H}$.

3.3.2 Special Case of Constant Control Authority

Instead of allowing for any u^* satisfying Assumptions 1-2, which could make (3.12) difficult to compute, this section chooses the control input so as to regulate the system to a constant rate of generalized inertia dissipation, i.e., chooses any u such that $\kappa^{(r)}(x,u)$ is a predefined constant. Specifically, consider a control law $u' : \mathcal{X} \to \mathcal{U}$ such that Assumption 3.1 holds and $u'(x) \in \boldsymbol{\mu}(x) \subset \mathcal{U}, \forall x \in \mathbb{R}^n$, where I define:

$$\boldsymbol{\mu}(x) \triangleq \left\{ u \in \mathcal{U} \;\middle|\; \kappa^{(r)}(x,u) = -a_{\max} \right\}, \tag{3.23}$$

where $a_{\max}$ is a precomputed constant rate of generalized inertia dissipation,

$$a_{\max} \triangleq \max \left(\left\{ a \in \mathbb{R} \;\middle|\; \forall x \in \mathcal{S}, \exists v \in (L_g L_f^{r-1} \kappa(x))^{\perp} : - \frac{(a + L_f^r \kappa(x))(L_g L_f^{r-1} \kappa(x))}{||L_g L_f^{r-1} \kappa(x)||^2} + v \in \mathcal{U} \right\} \right), \tag{3.24}$$

assuming an $a_{\max} > 0$ exists. This choice of u is reasonable if $\min_{u \in \mathcal{U}} \kappa^{(r)}(x,u)$ does not vary much with x, but may be overly conservative in other cases, where the system may be able to dissipate generalized inertia at a rate higher than $a_{\max}$ except within a small subset of $\mathcal{S}$.

Under any $u \in \boldsymbol{\mu}(x)$, it follows that $\kappa^{(r)}(x,u) = -a_{\max}$, so ψ_κ has the Taylor expansion:

$$\psi_\kappa(t; x, u') = \sum_{i=0}^{r-1} \frac{1}{i!} \kappa^{(i)}(x) t^i - \frac{1}{r!} a_{\max} t^r \,. \tag{3.25}$$

Next, define

$$t_c'(x) \triangleq \left\{ \arg\max_{t \geq 0} \psi_\kappa(t; x, u') \right\}, \tag{3.26}$$

$$h'(x) \triangleq \psi_\kappa(t_{c,0}; x, u'), \;\; \forall t_{c,0} \in t_c'(x) \,, \tag{3.27}$$

$$\mathcal{H}' \triangleq \{ x \in \mathbb{R} \mid h'(x) \leq 0 \} \,. \tag{3.28}$$

Remark 3.2. *The functions $t_c'(x)$ and $h'(x)$ always exist because $\psi_\kappa(t; x, u')$ is a polynomial with strictly negative highest coefficient $-a_{max}$. That is, Assumptions 3.2-3.3 are always satisfied for u'.*

By definition, $h'(x) \geq \kappa(x), \forall x \in \mathbb{R}^n$, so $\mathcal{H}' \subseteq \mathcal{S}$ as well. Lastly, define the set $\mathcal{U}'$ as

$$\mathcal{U}' \triangleq \left\{ u \in \mathcal{U} \;\middle|\; \exists x \in \mathcal{S} : u \in \boldsymbol{\mu}(x) \right\} \subseteq \mathcal{U} \,, \tag{3.29}$$

which represents the set of control inputs such that $\kappa^{(r)} = -a_{\max}$. For brevity, in this section let

$\psi_\kappa(t) = \psi_\kappa(t; x, u')$. I now state the second main result of this chapter.

Theorem 3.5. *The function h' in* (3.27) *is a CBF for the control set $\mathcal{U}'$ (or $\mathcal{U}$), provided $\mathcal{H}' \neq \emptyset$ in* (3.28).

Proof. As in Theorem 3.4, $t_c'(x)$ is not necessarily unique, so this proof considers two cases depending on the elements of $t_c'(x)$. If $0 \in t_c'(x)$, then $\psi_\kappa(0) = \kappa(x)$ and condition (3.4) is satisfied by the same argument as Theorem 3.4.

If $t_{c,0} \in t_c'(x)$ for $t_{c,0} > 0$, then $t_{c,0}$ is a maximizer on an open interval so $\frac{\partial \psi_\kappa(t)}{\partial t}\big|_{t=t_{c,0}} = 0$, where

$$0 = \left.\frac{\partial \psi_\kappa(t)}{\partial t}\right|_{t=t_{c,0}} = \sum_{i=0}^{r-2} \frac{(t_{c,0})^i}{i!}\kappa^{(i+1)}(x) - \frac{(t_{c,0})^{r-1}}{(r-1)!}a_{\max}\,. \tag{3.30}$$

Then one has

$$\begin{aligned} L_f\psi_\kappa(t_{c,0}) + L_g\psi_\kappa(t_{c,0})u &\overset{(3.25)}{=} \sum_{i=0}^{r-2}\frac{1}{i!}\kappa^{(i+1)}(x)(t_{c,0})^i + \frac{1}{(r-1)!}\left[L_f^r\kappa(x) + L_gL_f^{r-1}\kappa(x)u\right](t_{c,0})^{r-1} \\ &\overset{(3.30)}{=} \frac{(t_{c,0})^{r-1}}{(r-1)!}\left[a_{\max} + L_f^r\kappa(x) + L_gL_f^{r-1}\kappa(x)u\right]\,. \end{aligned} \tag{3.31}$$

By definition of $a_{\max}$, the right hand side of (3.31) can be rendered nonpositive by a $u \in \mathcal{U}' \subseteq \mathcal{U}$ independent of $t_{c,0}$, so by the same argument as Theorem 3.4, h' satisfies the definition of a CBF. ■

Similar to Theorem 3.4, Theorem 3.5 provides a guarantee of at least one safe trajectory, but $t_c'(x)$ is computed via polynomial root-finding rather than ODE propagation, making (3.31) easier to compute than (3.20)-(3.22) when implementing conditions (3.9)-(3.10). Note the polynomial degree depends only on the relative-degree of κ (usually $r \leq 4$ [82]), and not on the state dimension. Similar to $\dot{h}^*$, note that if $t_c'(x)$ ever has multiple elements, then $\dot{h}'$ is given by (3.20) with t_c' in place of t_c^* and where $\frac{\partial \psi_\kappa(t_{c,0};x,u')}{\partial x}$ is easily derived from (3.25). See also [82] for some restrictions on what t_c' can be when $r = 3, 4$. When $r = 2$, t_c' always contains only one element, which allows one to construct the following explicit form for h'.

Example 3.1. *In the case where $\kappa \in \mathcal{G}^2$, h' takes the form:*

$$h'(x) = \begin{cases} \kappa(x) & \dot{\kappa}(x) < 0 \\ \kappa(x) + \frac{\dot{\kappa}(x)^2}{2a_{max}} & \dot{\kappa}(x) \geq 0 \end{cases}. \tag{3.32}$$

3.4 Case Study for Spacecraft Application

In this section, I present two use cases for the CBFs in Section 3.3 for a spacecraft in weak gravity, in which the spacecraft must navigate around an object under observation using control inputs calculated online via a CBF.

In each case, the spacecraft state is $x = [r,\ v] \in \mathbb{R}^6$ with

$$\dot{x} = \begin{bmatrix} \dot{r} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} v \\ f_\mu(r) + u \end{bmatrix}, \tag{3.33}$$

where r is the vector from the center of the object under observation to the spacecraft, v is the rate of change of r as measured in an inertial frame (where the center of the object under observation is assumed to be non-accelerating in an inertial frame), f_μ is the local gravitational force, and u is the control thrust. I construct a preplanned path $r_p : [0, 1] \to \mathbb{R}^3$ of desired observations which circumnavigate the observed object (the interested reader is invited to examine the simulation code below to see the curve fitting algorithm used to construct r_p). This path is on the surface of the object, which is outside the safe set $\mathcal{S}$, so the spacecraft must get close to r_p while staying within the safe set. The spacecraft is driven to track the target using the Control Lyapunov Function (CLF) [85]:

$$V(x) = \frac{1}{2}||r - r_p||^2 + \frac{1}{2}k_2||v - k_1(r - r_p)||^2 . \tag{3.34}$$

The spacecraft control input is then calculated as:

$$\begin{aligned} u(x) = \underset{u\in\mathcal{U},\delta\in\mathbb{R}}{\arg\min}\ & u^{\mathrm{T}}u + J\delta^2 \quad \text{such that} \\ & L_f h(x) + L_g h(x)u \le \alpha(-h(x)) \\ & L_f V(x) + L_g V(x)u + \delta \le -k_3 V(x) \end{aligned} \tag{3.35}$$

where δ is a slack variable for the CLF to ensure feasibility, J is a constant slack penalty, k_1, k_2, k_3 are constants, and h is a CBF. By construction, the QP (3.35) will always be feasible in $\mathcal{H}^*$ and $\mathcal{H}'$ and I assume that (3.35) is sufficiently regular to satisfy Assumption 2.2-A6. For simplicity, let $\alpha(\lambda) = \lambda$ in (3.35). Note that (3.35) utilizes a CLF and a slack penalty to establish convergence rather than a nominal control law u_{nom} as in (2.9).

3.4.1 Case 1: Spherical Target

I first consider a spherical object of radius ρ_t with fixed position r_s, and negligible gravity $f_\mu \equiv 0$, so (3.33) reduces to a double integrator. Maintaining a distance of ρ_s from the object is

equivalent to maintaining the relative-degree 2 constraint function $\kappa_a(x) \leq 0$:

$$\kappa_a(x) = \rho_a - ||r - r_s||\,, \tag{3.36}$$

where $\rho_a = \rho_s + \rho_t > 0$. Note that κ_a is defined specifically so that $\ddot{\kappa}_a$ represents physical acceleration, since if fuel is consumed slowly enough, then the spacecraft peak acceleration in any direction is a known constant, so a_{max} is easy to compute. Suppose the spacecraft has six identical and orthogonal thrusters, so $\mathcal{U} = \{u \in \mathbb{R}^3 : \|u\|_\infty \leq u_{\text{max}}\}$ for some $u_{\text{max}} \in \mathbb{R}_{>0}$.

I then construct a CBF h^* as in Section 3.3.1 using the assumed control law u^*_{ball} in (3.11). Specifically, to compute h^*, the controller numerically propagates the dynamics (3.33) for some amount of time T_{prop}, yielding an array of states $\{\psi_x(t_k; x, u^*_{\text{ball}})\}_{k=1}^N$, and an array of κ values $\{\psi_\kappa(t_k; x, u^*_{\text{ball}})\}_{k=1}^N$. For this particular combination of dynamics (3.33) and control law (3.11), there is always a unique maximizer time $t_{c,0}$. Moreover, $t_{c,0}$ can be upper bounded, and I use this bound to choose the propagation time T_{prop}. Unfortunately, depending on the integration method, the array $\{\psi_\kappa(t_k; x, u^*_{\text{ball}})\}_{k=1}^N$ may not include the true maximizer, since this array is only a sampling of a continuous curve. Thus, the controller selects the three highest points $\{\psi_\kappa(t_{k_l}; x, u^*_{\text{ball}})\}_{l=1}^3$ and fits a quadratic curve $q(\tau) = a\tau^2 + b\tau + c$ to these three points (where τ is the propagated time minus the current time, i.e. $q(0) = \kappa(x)$). Assuming the maximum is not simply $h^*(x) = \kappa(x)$ (i.e. that $t_{c,0} \neq 0$), the code then chooses $h(x) = \max_{\tau \in \mathbb{R}_{\geq 0}} q(\tau) = c - \frac{b^2}{4a}$, which is associated with a unique maximizer $t_q = \arg\max_{\tau \in \mathbb{R}_{\geq 0}} q(\tau) = \frac{-b}{2a}$. From here, the gradient of h^* can be computed as in Remark 3.1 using $t_{c,0} = t_q$. Alternatively, for these dynamics specifically, it holds that $\Theta(t_{c,0}) = \begin{bmatrix} I_{3\times 3} & t_{c,0} I_{3\times 3} \\ 0_{3\times 3} & I_{3\times 3} \end{bmatrix}$ in (3.21) (note that u^*_{ball} is not continuously differentiable here because of the ∞-norm, but the ODE (3.22) still has a solution because u^*_{ball} is differentiable almost everywhere; proving that Remark 3.1 still holds in this case is an open problem not considered here; see also [206, Thm. 6.1]). For more details, see the simulation code below. Note, for more complex problems where $t_c(x)$ may have more than one element, if there are multiple local maximizers t_k such that the values $\psi_\kappa(t_k; x, u^*)$ are close to each other (but not necessarily identical), then I recommend constructing $q(\tau)$ for each local maximizer.

Next, for the same system, I construct h' as in Section 3.3.2. It follows that h' is as given in (3.32) with κ_a in place of κ and $a_{\text{max}} = u_{\text{max}}$. This leads to $\mathcal{U}' = \{u \in \mathbb{R}^3 : \|u\| \leq u_{\text{max}}\} \subset \mathcal{U}$. Finally, for comparison, consider the CBF

$$h_o(x) = \left(\arctan(\dot{\kappa}_a(x)) + \frac{\pi}{2}\right) \kappa_a(x)\,, \tag{3.37}$$

derived using the rules in the prior work [29]. As discussed in [29], and similar to [84, 85, 138], this CBF is only guaranteed to be valid over $\mathcal{U} = \mathbb{R}^3$.

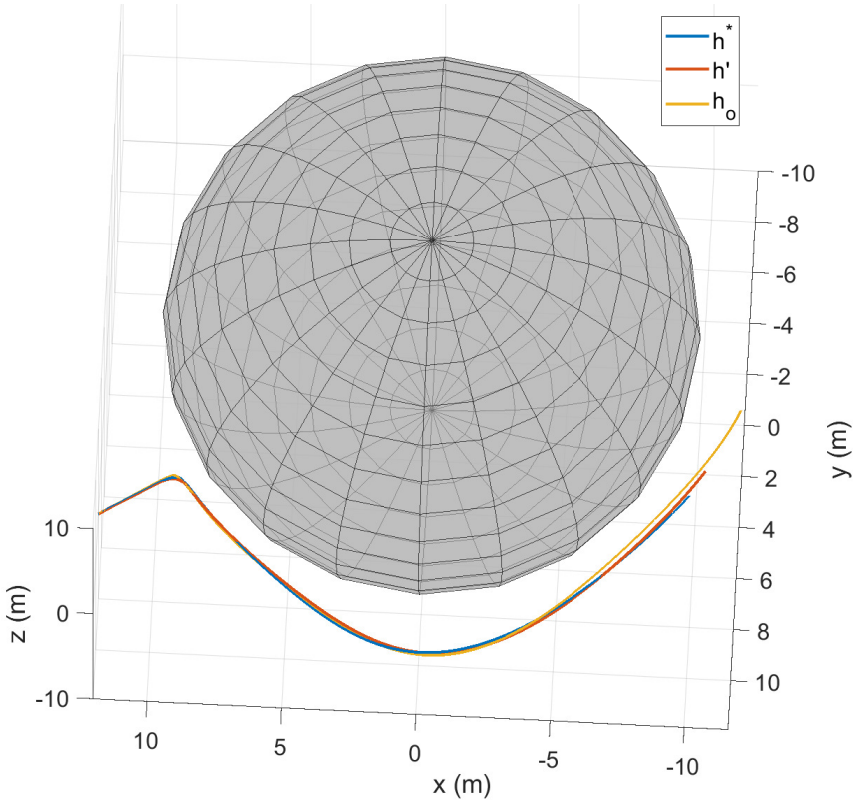

Figure 3.2: Simulation and Comparison for Spherical Obstacle. The paths around the spherical obstacle under the three CBFs considered $(h^*(x), h'(x), h_o(x))$

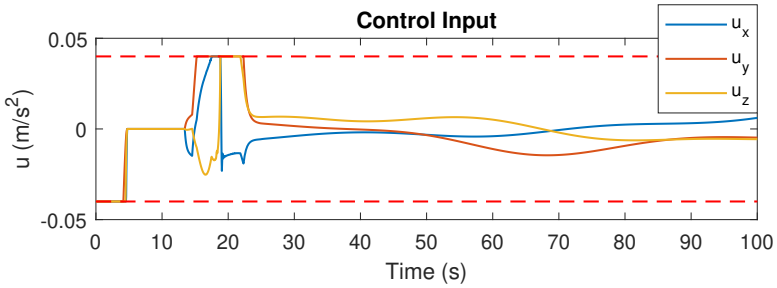

Figure 3.3: Control Inputs Using Backup CBF. The control input using h^* as the CBF

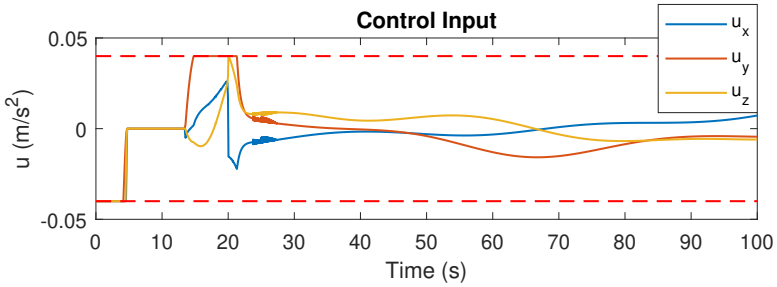

Figure 3.4: Control Inputs Using Polynomial-Based CBF. The control input using h' as the CBF

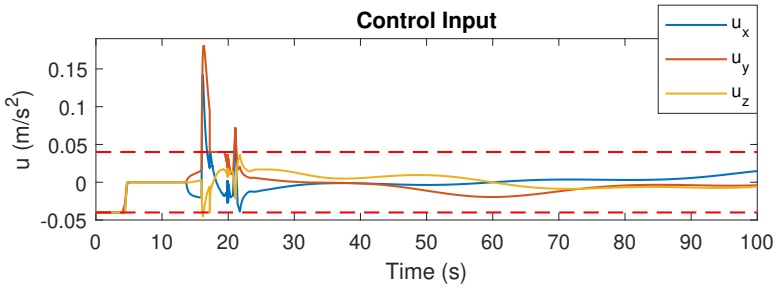

Figure 3.5: Control Inputs Using Comparison CBF. The control input using h_o as the CBF, which necessitates using control inputs outside the prescribed bounds (dashed red lines) for (3.35) to have a solution

I then simulated trajectories under all three CBFs (h^*, h', h_o) in MATLAB (the simulation code for this chapter can be found at https://github.com/jbreeden-um/phd-code/tree/main/2021/CDC%20High%20Relative%20Degree%20CBFs%20Input%20Constraints). The three simulated paths are shown in Fig. 3.2, and the corresponding control inputs are shown in Figs. 3.3-3.5. All three trajectories in Fig. 3.2 remained safe and generally followed similar paths around the obstacle. However, the control input only stayed within the designated control set in Figs. 3.3-3.4. For the trajectory under $h_o(x)$, the QP in (3.35) became infeasible at $t = 14$ (see Fig. 3.5), so I had to expand the control set $\mathcal{U}$ to compute a control input satisfying the safety constraints (3.9)-(3.10). Thus, the proposed methods always yield trajectories that are safe in the presence of control input constraints, whereas earlier methods might not.

Additionally, note that the lines for u_x, u_z for $t \in [15, 25]$ are slightly closer to the horizontal axis in Fig. 3.4 than in Fig. 3.3. This occurred because h' only makes use of $u \in \mathcal{U}' \subset \mathcal{U}$ even though the QP was calculated over the complete control set $\mathcal{U}$.

3.4.2 Case 2: Asteroid Target

Next, I consider a spacecraft avoiding an object more complicated than a sphere, in this case an asteroid, but still in weak gravity, with gravity modeled by spherical harmonics [207]. If the asteroid is convex, one could simply use $\kappa_a(x)$ as the constraint function, setting r_s as the instantaneous closest point. However, if the asteroid is nonconvex (as in this example), the spacecraft could obtain a large velocity with respect to a point other than the closest point, so that strategy is no longer sufficient. Instead, I consider a discrete point-cloud model of the entire asteroid. The simplest response is to construct a CBF for every point in the model, similar to [136], though this could be computationally demanding (computational costs will also depend on the optimizer used). Using the result in [151, Thm. 3] one can reduce the number of constraints to only those constraints that could be violated within some finite time horizon $\Delta t > 0$ to reduce complexity. For this case study, I chose asteroid 433 Eros. The point cloud is a shape model of Eros with $N = 7790$ plates [208], and gravity f_μ modeled using the 16th order spherical harmonics in [209]. For this simulation, I assume that the spacecraft knows the truth gravity model. I construct a CBF with the form of h' for every point, yielding $\{h'_i\}_{i=1}^N$. I then apply a QP control law similar to (3.35) with multiple CBF constraints $\dot{h}'_i(x, u) \leq \alpha(-h'_i)$; in this chapter, I assume that all these CBF constraints are simultaneously feasible (which turns out to be true along the simulation trajectory), but I also note that simultaneous feasibility of several CBFs is not necessarily guaranteed unless one follows the analysis in Chapter 6, which potentially requires additional margins.

To make the spacecraft traverse around the asteroid, I construct a preplanned path $r_p : [0, 1] \to$

$\mathbb{R}^3$ around the asteroid. I then let $s \in [0, 1]$ be an additional state with single integrator dynamics $\dot{s} = u_{\text{ground}}$ for $u_{\text{ground}} \in \mathbb{R}$, and I enforce the constraint $h_{\text{ground}}(x, s) = \rho_{\text{ground}} - \|r - r_p(s)\|$ remain nonpositive, where $\rho_{\text{ground}} > \rho_a$. Thus, as the spacecraft tries to get closer to the instantaneous target r_p, the position of r_p also advances. Note that h_{ground} is already a CBF, because I let the artificial state s have first-order dynamics with no input constraints.

The trajectory, control inputs, and CBF values for the asteroid simulation are shown in Figs. 3.6-3.8 and a video of the scenario can be found at https://youtu.be/JKj3PUrYnEg. As expected, the trajectory is always safe and u satisfies the input constraints. Since the spacecraft generally moves tangentially to the asteroid, h' is only slightly larger than κ_a in Fig. 3.8 for most of the simulation.

3.5 Conclusions

In this chapter, I introduced two general methods to determine CBFs for high relative-degree constraint functions. Both approaches are derived from control policies known to meet the input constraints, and thus are guaranteed safe under such constraints without further tuning. The two strategies were demonstrated safe under input constraints on a spacecraft obstacle-avoidance example, whereas most prior techniques would have failed to meet the input constraints [29, 84, 85, 203] or required clever selection of bounding functions [87]. The second strategy was further verified on an asteroid exploration example. Both strategies benefited from a relaxed definition of CBF in Definition 3.3. While these two methods are fairly generally applicable, there will always be a need for future researchers to develop additional methods of finding CBFs for systems not yet covered by these methods. This need is demonstrated most clearly by the problem posed in Chapter 7, for which none of these methods are applicable.

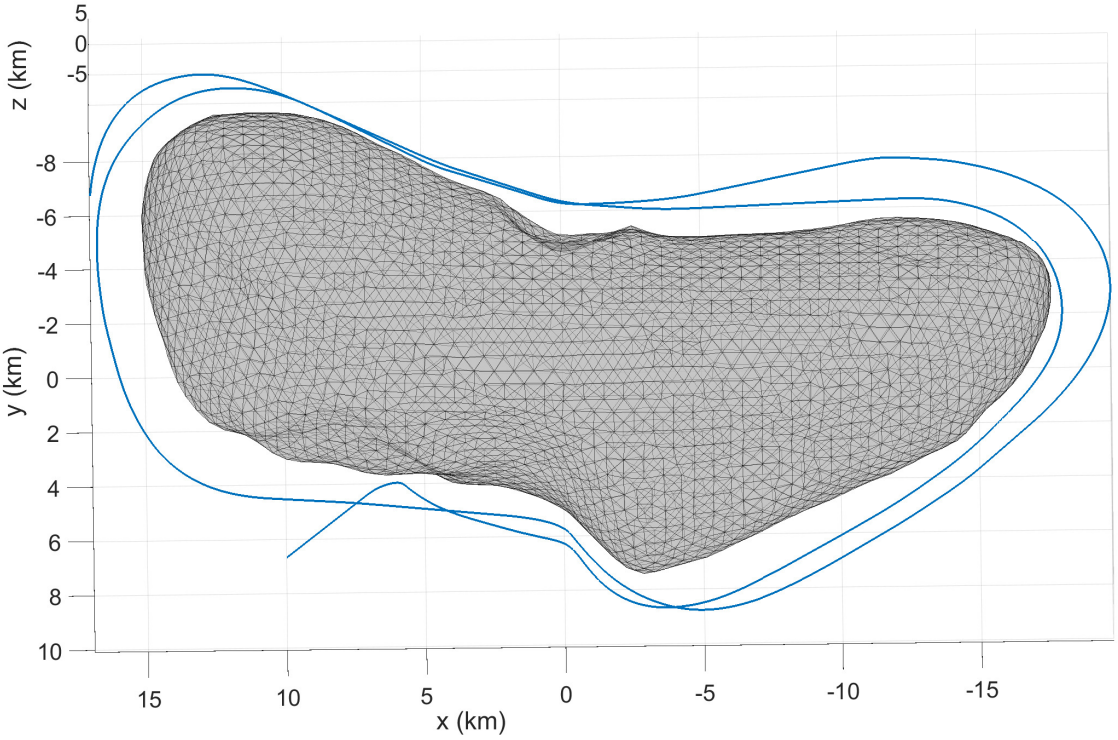

Figure 3.6: Eros Circumnavigation Trajectory (Non-Rotating). The trajectory around Eros (top view)

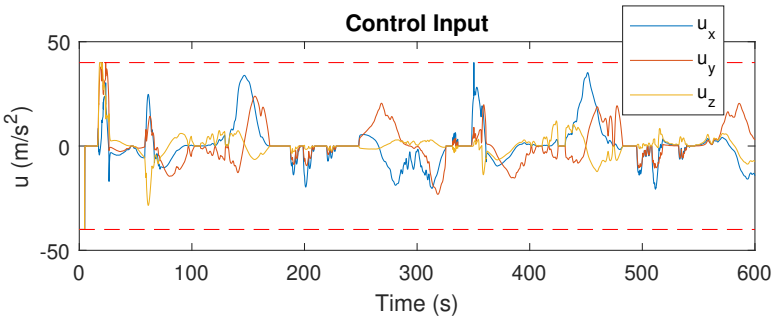

Figure 3.7: Eros Circumnavigation Control Inputs. The control input over time for Fig. 3.6, which stays between the prescribed bounds (dashed red lines)

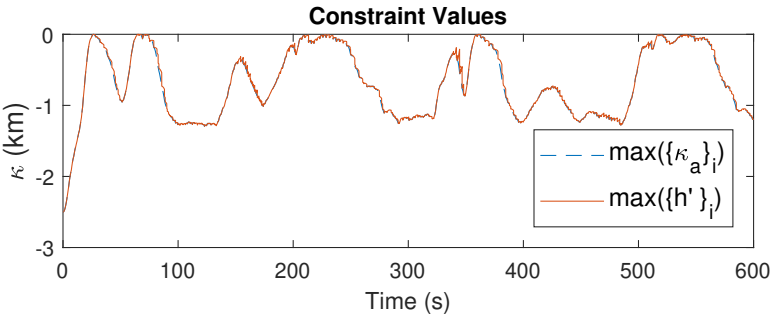

Figure 3.8: Eros Circumnavigation Constraint Function and CBF Values. The value of the largest $\kappa_a(x)$ and $h'(x)$ over time

CHAPTER 4

Robust Control Barrier Functions

This chapter is concerned with the "robust high relative-degree problem", formalized in Section 4.2.2. This is an extension of the problem considered in Chapter 3, wherein now the system may experience disturbances. These disturbances this could represent model uncertainty, measurement error, sampling error, or others. Regardless of the cause, I assume that these disturbances are bounded. I then describe how to design CBFs as in the previous chapter, now for this revised system model and with more emphasis specifically on second-order systems rather than on arbitrary relative-degrees as in Chapter 3. After describing these "Robust CBFs" in detail, I present a special application to what I call a "tight-tolerance objective", and show how the Robust CBF approach can be used not just for safety (i.e. avoidance) but also for accomplishing objectives defined as convergence (i.e. attraction) to a set, or equivalently, convergence to a specific state within some tolerance.

4.1 Introduction

This chapter advances the recent theory of Control Barrier Functions (CBFs) to systems with high relative-degree under input constraints and disturbances, and applies the results to control design for satellite trajectories. Currently, satellite trajectory design is generally the product of extensive optimizations for fuel and/or time consumption, completed long before a satellite is deployed. As spacecraft venture further away from the Earth and attempt more complex mission objectives, there is a need for greater autonomy, and a subsequent need to ensure that autonomous trajectories meet various requirements, herein termed *safety*. While the present focus is on spacecraft, the following results are broadly applicable to systems with constraints of high relative-degree.

System safety is often formulated as an invariance problem for a set of safe states, referred to as the *safe set*. As in Chapter 3, the safe set is defined as the zero sublevel set of some *constraint function*, which is assumed to be of high relative-degree with respect to the system dynamics, as is the case for satellite problems. This chapter then designs a CBF whose zero sublevel set, called

the *CBF set*, is a subset of the safe set. Existing CBF theory then provides a sufficient condition (e.g. (2.5),(2.6)), which I call the *CBF condition*, on the control input that establishes forward invariance of the CBF set. However, one limitation of this approach is that finding a valid CBF may be challenging in general. Thus, the main questions of this chapter are: given a constraint function with relative-degree 2 (or higher in Section 4.3.3) and specified input constraints, 1) how to determine a CBF whose CBF set is a subset of the safe set and which is valid with respect to the input constraints, and 2) how to ensure invariance of the CBF set in the presence of bounded disturbances.

Given this setup, several papers develop methods on constructing CBFs whose CBF sets are equivalent to the safe set. Prior methods include constructing CBFs via compositions of the constraint function and its derivatives [29], backstepping [84], or feedback linearization [85, 102, 138, 203]. However, the aforementioned papers all require that the set of allowable control inputs is $\mathbb{R}^m$. Since the constraint function is of high relative-degree, there may exist states in the safe set from which arbitrarily large control inputs are needed to render the system trajectories within the safe set. Therefore, if the set of allowable control inputs is bounded, there may be no admissible control input that keeps the system trajectories within the safe set, and hence the system could become unsafe (e.g., the satellite might fail to decelerate before colliding with an obstacle and/or (2.9) might become infeasible). The work in [87] fixes this issue by specifying the CBF set as a viability domain for the given system dynamics—i.e. a set inside the safe set that may be rendered forward invariant under input constraints—and provides parameters (class-$\mathcal{K}$ functions) that can be tuned to meet various input constraints. However, finding such parameters is a non-trivial challenge, akin to finding a Lyapunov function. Recently, such parameters have been found via learning from expert demonstration [101] and reinforcement learning [157]. This chapter will similarly specify a subset of the safe set that is controlled forward invariant but, unlike [87], here I use first-order CBFs [66] and I provide three principled methodologies to select the analogue of the parameters in [87] for certain classes of systems (namely, systems satisfying the theorem assumptions in Section 4.3). These methodologies are based on feedback linearization (distinct from [85] and similar to Section 3.3.2), potential energy functions (the completely new method in this Chapter), and model-predictive safety (similar to Section 3.3.1), respectively. The significance of these results and those in [87, 101, 157] is that if a valid CBF can be found via one of these methods, then existence of a safe trajectory is guaranteed from any point in the CBF set.

Other studies have taken advantage of special features of certain systems to satisfy input constraints when the constraint function is of high relative-degree. The work in [36] develops a CBF specific to the double integrator system. Similarly, [82] develops a technique applicable to the n-integrator system, which was generalized in Section 3.3.2. The method in Section 3.3.2 is then further extended in Section 4.3.1 to be robust to disturbances while maintaining provable safety

under input constraints, and in Section 4.3.2 to work with more general dynamics for which $a_{\max}$ in (3.24) is possibly zero. Alternatively, one could use an additional CBF to limit the agent velocity to a certain domain, as in [29, 36]. The example in [28] instead takes advantage of the damping of the considered system to meet input constraints. Finally, the work in [86, 96] and in Section 3.3.1 considers CBFs that examine a system's trajectory forward in time under various assumptions to determine safety. The authors of [96] consider the set of states reachable in fixed time from a pre-designated *backup set*, but do not consider whether the predicted trajectories from the current state to the backup set are everywhere safe (i.e. whether the backup trajectories become unsafe in the process of reaching the backup set). The work in [86] improves upon this by examining the minimizer of a *performance function* applied along the predicted trajectories (thus testing every point along the backup trajectory for safety), and Section 3.3.1 extends [86] to work when the minimizer is not unique. Compared to these works, Section 4.3.3 extends this strategy to be robust to disturbances while maintaining provable safety under input constraints, and provides tools for computing the CBF and its derivatives when analytic solutions are unavailable (see also [96, Sec. V-A]).

Several papers have considered CBF robustness to disturbances in various senses. Neglecting input constraints, an early result on CBF robustness in [116] shows that a bounded disturbance causes a bounded excursion outside the CBF zero sublevel set, and later authors showed that this excursion can be tuned [112]. Recently, [89] extended this result to higher-order CBFs as in [87]. Safety under a bounded worst-case disturbance, as is considered in this chapter, is studied in [117–119], while probability of safety using a similar approach with a stochastic disturbance is studied in [72]. In multi-agent systems, dynamic couplings between agents that act independently of each other can also be considered disturbances, and robustness to such effects are treated similarly in [80, 124]. In all of these papers, it is assumed that the system has sufficient control authority to counteract these disturbances. However, satisfying this assumption is nontrivial, and is a requirement of the methods in Section 4.3.

Finally, one objective of this chapter is to place fewer restrictions on closed-loop trajectories by applying safety criteria only near the boundary of the CBF set. This was accomplished in [89] by designating strict subsets of the safe set termed "performance-critical regions" where safety was guaranteed without enforcing the CBF condition. However, this relaxation required expanding the control set to $\mathbb{R}^m$. For systems subject to many CBFs simultaneously, the work in [36, 203] simplifies control input calculation in a similar manner by breaking the state space into regions where only a few CBFs are actively applied, though this introduces potential issues with non-uniqueness of system solutions. These issues are fixed in [151] by relaxing the system to a differential inclusion. A similar approach using products of CBFs is described in [204]. Expanding upon these approaches, this chapter introduces a hysteresis-switching approach inspired by [151] and [28] that

relaxes the CBF condition in the interior of the CBF set while still provably guaranteeing safety in the presence of input constraints. Such hysteresis-switching removes the need for differential inclusions, and thus prevents chattering control inputs that may not be feasible on real actuators. This switching approach also motivates a special choice for the class-$\mathcal{K}_e$ function that is left as a free tuning parameter in all of the theorems in Section 4.3 (as was also the case for Theorem 3.4 and Theorem 3.5). I then show in simulation how the proposed choice allows one to directly tune how closely trajectories approach the boundary of the CBF set.

In summary, this chapter is divided into:

1. three strategies for generating CBFs from high relative-degree constraint functions in the presence of input constraints and bounded matched and unmatched disturbances simultaneously (Section 4.3);

2. a switching method for relaxing the CBF condition in the interior of the CBF set and for tuning how closely trajectories approach the boundary of the CBF set (Section 4.4);

3. specializations of the above CBFs to deep-space trajectory applications (Section 4.5); and

4. an extension and case study of these methods to "tight-tolerance specifications", introduced later in Section 4.7.

4.2 Preliminaries

4.2.1 Notation

In addition to the notations in Section 2.1, for a continuously differentiable function $\phi : \mathbb{R} \to \mathbb{R}$, let $\phi' : \mathbb{R} \to \mathbb{R}$ denote the first derivative of ϕ. Denote the inverse, if it exists, as $\phi^{-1} : \mathbb{R} \to \mathbb{R}$. For a continuously differentiable function of both time and state, $\varphi : \mathcal{T} \times \mathcal{X}$, denote the partial derivative with respect to the first argument as $\partial_t \varphi$. Denote the gradient row vector with respect to the second argument as $\nabla \varphi$. That is, $\partial_t \varphi(s, x) \triangleq \left.\frac{\partial \varphi(t,x)}{\partial t}\right|_{t=s}$ and $\nabla \varphi(t, y) \triangleq \left.\frac{\partial \varphi(t,x)}{\partial x}\right|_{x=y}$. Let $\dot{\varphi}$ denote the total derivative of φ along the dynamics (4.1), $\dot{\varphi}(t, x, u, w_u, w_x) = \partial_t \varphi(t, x) + \nabla \varphi(t, x) F(t, x, u, w_u, w_x)$, where it may be the case that $\varphi(t, x, u, w_u, w_x) = \varphi(t, x, w_x)$ if φ is of high relative-degree. For the purpose of illustrating concepts, I may also reference the second derivative, $\ddot{\varphi}$, or rth derivative, $\varphi^{(r)}$, along some dynamics, though note that derivatives beyond the first order are generally not well-defined in this chapter (and thus should not be used in equations) unless $w_x \equiv w_u \equiv 0$, because the disturbances w_u and w_x are not assumed to be differentiable. The exception to this remark is $\ddot{\kappa}_w$, which is well-defined in (4.20). Additional derivative notation is introduced in Section 4.3.3 to prevent confusion in that section.

4.2.2 Model and Problem

Consider the time-varying control-affine model

$$\dot{x} = \underbrace{f(t,x) + g(t,x)(u + w_u) + w_x}_{=F(t,x,u,w_u,w_x)}, \tag{4.1}$$

with time $t \in \mathcal{T} = [t_0, \infty)$, state $x \in \mathcal{X} \subseteq \mathbb{R}^n$, control input $u \in \mathcal{U} \subset \mathbb{R}^m$ where $\mathcal{U}$ is compact, unknown disturbances $w_u \in \mathbb{R}^m$ and $w_x \in \mathbb{R}^n$ that are continuous in time, and functions $f : \mathcal{T} \times \mathcal{X} \to \mathbb{R}^n$ and $g : \mathcal{T} \times \mathcal{X} \to \mathbb{R}^{n \times m}$. Let w_u and w_x be bounded as $\|w_u\| \leq w_{u,\max}$ and $\|w_x\| \leq w_{x,\max}$ for some $w_{u,\max}, w_{x,\max} \in \mathbb{R}_{\geq 0}$, and define the set of allowable disturbances $\mathcal{W} \triangleq \{w_u \in \mathbb{R}^m \mid \|w_u\| \leq w_{u,\max}\} \times \{w_x \in \mathbb{R}^n \mid \|w_x\| \leq w_{x,\max}\}$. Assume a unique solution to (4.1) exists for all $t \in \mathcal{T}$. Given dynamics (4.1) with f and g sufficiently differentiable, a function $\varphi : \mathcal{T} \times \mathcal{X} \to \mathbb{R}$ is said to be of relative-degree r if it is r-times total differentiable along (4.1) for $w_x \equiv w_u \equiv 0$, and if $\varphi^{(r)}$ is the lowest order derivative in which u and w_u appear explicitly. Denote the set of all relative-degree r functions as $\mathcal{G}^r$, similar to Definition 3.2 in Chapter 3.

Remark 4.1. *Instead of viewing w_x as a disturbance, one could alternatively view x as a filtered estimate of the true state of the system (which in practice is never known exactly), and w_x as the corrections to that estimate (provided that one can approximate a bound $w_{x,\max}$).*

Let $\kappa : \mathcal{T} \times \mathcal{X} \to \mathbb{R}$, $\kappa \in \mathcal{G}^r$, denote the *constraint function*, and define a time-varying safe set $\mathcal{S} : \mathcal{T} \to 2^{\mathcal{X}}$ as

$$\mathcal{S}(t) \triangleq \{x \in \mathcal{X} \mid \kappa(t,x) \leq 0\}. \tag{4.2}$$

For compactness, denote the safe set across time as $\mathcal{S}^{\mathcal{T}} \triangleq \{(t,x) \in \mathcal{T} \times \mathcal{X} \mid x \in \mathcal{S}(t)\}$.

This chapter is devoted to developing methods for rendering the state trajectory always inside the safe set $\mathcal{S}$ in the presence of any allowable disturbances $(w_u, w_x) \in \mathcal{W}$. I will do this by constructing functions $h : \mathcal{T} \times \mathcal{X} \to \mathbb{R}$ that generate sets of the form

$$\mathcal{H}(t) \triangleq \{x \in \mathcal{X} \mid h(t,x) \leq 0\}, \tag{4.3}$$

$$\mathcal{H}_{\text{res}}(t) \triangleq \{x \in \mathcal{X} \mid h(t,x) \leq 0 \text{ and } \kappa(t,x) \leq 0\}, \tag{4.4}$$

visualized in Fig. 4.1. I refer to the set $\mathcal{H}$ as a *CBF set*, and to the set $\mathcal{H}_{\text{res}}$ as a *restricted CBF set*. Note that if $h(t,x) \geq \kappa(t,x)$ for all $(t,x) \in \mathcal{T} \times \mathcal{X}$, then $\mathcal{H} \equiv \mathcal{H}_{\text{res}}$. A controller is said to render $\mathcal{H}_{\text{res}}$ forward invariant, if given any $x(t_0) \in \mathcal{H}_{\text{res}}(t_0)$, the closed-loop trajectory satisfies $x(t) \in \mathcal{H}_{\text{res}}(t), \forall t \in \mathcal{T}$, similar to Definition 2.6. In general, there may exist points $x(t_0) \in \mathcal{S}(t_0)$, from which one will not be able to render $\mathcal{S}$ forward invariant under (4.1). That is, $\mathcal{S}$ (or $\mathcal{S}^{\mathcal{T}}$) is generally not a controlled invariant set as in Definition 2.7. Nevertheless, if one can render the

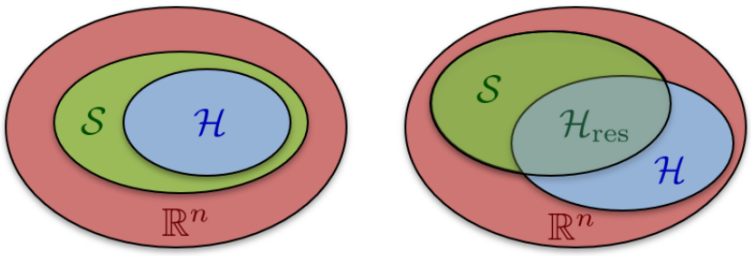

Figure 4.1: Illustration of CBF Set and Restricted CBF Set. Given a safe set $\mathcal{S} \subset \mathcal{X}$ as in (4.2) and input constraints $\mathcal{U}$, this chapter presents methods of finding viability domains $\mathcal{H}$ as in (4.3) that are subsets of the safe set (left). In certain cases (Theorem 4.10 and Theorem 4.11), the presented forms of κ cause the set $\mathcal{H}$ to include unsafe states (see also Fig. 4.2), so I introduce the set $\mathcal{H}_{\text{res}}$ in (4.4) (right). If κ is an RCBF as in Definition 4.2, then $\mathcal{H}$ (or $\mathcal{H}_{\text{res}}$ in Theorem 4.10 and Theorem 4.11) can be rendered forward invariant under any disturbances $(w_u, w_x) \in \mathcal{W}$ while always satisfying the input constraints $\mathcal{U}$.

subset $\mathcal{H}_{\text{res}} \subseteq \mathcal{S}$ forward invariant, then one can ensure that the closed loop trajectories of (4.1) are safe (i.e. never exit $\mathcal{S}$) for initial conditions lying in the set $\mathcal{H}_{\text{res}}$. Thus, a crucial requirement is that $x(t_0) \in \mathcal{H}_{\text{res}}(t_0)$, where κ is chosen from the strategies in Section 4.3. Also define the domains $\mathcal{H}^{\mathcal{T}} \triangleq \{(t,x) \in \mathcal{T} \times \mathcal{X} \mid x \in \mathcal{H}(t)\}$ and $\mathcal{H}^{\mathcal{T}}_{\text{res}} \triangleq \{(t,x) \in \mathcal{T} \times \mathcal{X} \mid x \in \mathcal{H}_{\text{res}}(t)\}$ similar to $\mathcal{S}^{\mathcal{T}}$.

At this point, I emphasize that the purpose of this chapter is to derive conditions that ensure that state trajectories never leave $\mathcal{H}_{\text{res}}$ even in the presence of disturbances. There is a related school-of-thought that is instead interested in designing CBFs such that $\mathcal{H}_{\text{res}}$ is asymptotically stable, e.g. [29, Prop. 2], and thus a bounded disturbance will cause a bounded excursion outside $\mathcal{H}_{\text{res}}$. Since I am interested in spacecraft, I assume that any excursion outside the safe set could lead to catastrophic damage and is therefore unacceptable. Note however that, mathematically, these two notions of robustness are very similar, and the relation will become clearer in the context of the asymptotically stable manifolds discussed in Section 4.4 and Section 4.7.

4.2.3 Mathematical Background

Recall the definition of a CBF without disturbances in Definition 2.14. Now, to account for disturbances, I introduce the following definitions, inspired by [117, 118].

Definition 4.1 (Control Barrier Function on a Set)**.** *For the system* (2.1)*, a continuously differentiable function* $h : \mathcal{T} \times \mathcal{X} \to \mathbb{R}$ *is a* control barrier function (CBF) on a time-varying set $\mathcal{D} : \mathcal{T} \to 2^{\mathcal{X}}$ *if there exists a function* $\alpha \in \mathcal{K}_e$ *such that*

$$\inf_{u \in \mathcal{U}} \Big[\partial_t h(t,x) + \nabla h(t,x)\big[f(t,x) + g(t,x)u\big]\Big] \leq \alpha(-h(t,x)), \forall x \in \mathcal{D}(t), t \in \mathcal{T}\,. \tag{4.5}$$

Definition 4.2 (Robust Control Barrier Function on a Set)**.** *For the system* (4.1)*, a continuously differentiable function* $h : \mathcal{T} \times \mathcal{X} \to \mathbb{R}$ *is a* robust control barrier function (RCBF) on a time-varying set $\mathcal{D} : \mathcal{T} \to 2^{\mathcal{X}}$ *if there exists a function* $\alpha \in \mathcal{K}_e$ *such that*

$$\max_{(w_u,w_x)\in\mathcal{W}} \Big[\inf_{u\in\mathcal{U}} \Big(\partial_t h(t,x) + \nabla h(t,x)\big[f(t,x) + g(t,x)(u+w_u) + w_x\big] \Big) \Big] \leq \alpha(-h(t,x)), \\ \forall x \in \mathcal{D}(t), t \in \mathcal{T}\,. \quad (4.6)$$

Note the added language "CBF (or RCBF) on a set $\mathcal{D}$" compared to Definition 2.14 and Definition 3.3. In prior chapters, a function was a CBF if it was a CBF on $\mathcal{H}$ or any superset $\mathcal{D}$ of $\mathcal{H}$. By contrast, in this chapter, the set $\mathcal{D} : \mathcal{T} \to 2^{\mathcal{X}}$ may be any set, not necessarily a superset of $\mathcal{H}$, so I require that this set is specified whenever Definition 4.1 or Definition 4.2 is invoked. Usually, $\mathcal{D}$ will be either $\mathcal{H}$ or $\mathcal{H}_{\text{res}}$, though other sets will be possible later in Chapter 6.

Based on Definition 4.2, define

$$\begin{aligned} W(t,x) &\triangleq \max_{(w_u,w_x)\in\mathcal{W}} \nabla h(t,x)(g(t,x)w_u + w_x) \quad (4.7)\\ &\equiv \|\nabla h(t,x)g(t,x)\| w_{u,\max} + \|\nabla h(t,x)\| w_{x,\max}\,, \end{aligned}$$

where I will use both equivalent forms of $W(t,x)$ in (4.7) depending on the setting. The set of control inputs such that (4.6) is satisfied is then

$$\mathcal{U}_{\text{rcbf}}(t,x) \triangleq \{u \in \mathcal{U} \mid \partial_t h(t,x) + \nabla h(t,x)(f(t,x) + g(t,x)u) \leq \alpha(-h(t,x)) - W(t,x)\} \quad (4.8)$$

for some function $\alpha \in \mathcal{K}_e$. Note that since Definition 4.2 considers the allowable control set $\mathcal{U}$, if h is an RCBF on $\mathcal{D}$, then $\mathcal{U}_{\text{rcbf}}(t,x)$ is nonempty for all $x \in \mathcal{D}(t), t \in \mathcal{T}$. Given an RCBF on $\mathcal{H}$, one can establish forward invariance of $\mathcal{H}$ using the following theorem.

Theorem 4.1 (Extension of Theorem 2.2 to RCBFs)**.** *For the system* (4.1)*, suppose* $h : \mathcal{T}\times\mathcal{X} \to \mathbb{R}$ *is an RCBF as in Definition 4.2 on an open set* $\mathcal{D} : \mathcal{T} \to 2^{\mathcal{X}}$ *satisfying* $\mathcal{H}(\tau) \subset \mathcal{D}(\tau)$ *for all* $\tau \in \mathcal{T}$ *with* $\mathcal{H}$ *as in* (4.3)*. Let* $\mathcal{D}^{\mathcal{T}} = \{(t,x) \in \mathcal{T} \times \mathcal{X} \mid x \in \mathcal{D}(t)\}$ *and let* $\alpha \in \mathcal{K}_e$*. Then any control law* $u : \mathcal{T} \times \mathcal{X} \to \mathcal{U}$ *satisfying* $u(t,x) \in \mathcal{U}_{\text{rcbf}}(t,x), \forall (t,x) \in \mathcal{D}^{\mathcal{T}}$ *will render* $\mathcal{H}^{\mathcal{T}}$ *forward invariant.*

The proof of Theorem 4.1 follows identical logic to Theorem 2.2 and is similar to that of [118, Lemma 4]. A version for time-invariant sets is also found in [117, Thm. 2].

Note that the methods in Section 4.3 only output functions that are guaranteed to be RCBFs on $\mathcal{H}$ (or $\mathcal{H}_{\text{res}}$), not necessarily on a larger open set $\mathcal{D}$ as required in Theorem 4.1. Therefore, one should verify that the dynamics do not drastically change right at the boundary $\partial\mathcal{H}$ (i.e. change so as to make $\dot{h}$ non-Lipschitz and positive in a neighborhood $\{(t,x) \in \mathcal{T} \times \mathcal{X} \mid h(t,x) \in [0,\epsilon]\}$ for

$\epsilon \in \mathbb{R}_{>0}$), as was discussed following Theorem 2.2. Alternatively, one can use one of the following theorems that only require $u(t,x) \in \mathcal{U}_{\text{rcbf}}(t,x)$ for $(t,x) \in \mathcal{H}^{\mathcal{T}}$ instead of on a larger open set $\mathcal{D}^{\mathcal{T}}$.

Corollary 4.2. *For the system* (4.1)*, suppose* $h : \mathcal{T} \times \mathcal{X} \to \mathbb{R}$ *is an RCBF as in Definition 4.2 on the set* $\mathcal{H}$ *in* (4.3)*. Assume that* $\alpha \in \mathcal{K}_e$ *is locally Lipschitz continuous. Then any control law* $u : \mathcal{T} \times \mathcal{X} \to \mathcal{U}$ *satisfying* $u(t,x) \in \mathcal{U}_{\text{rcbf}}(t,x), \forall (t,x) \in \mathcal{H}^{\mathcal{T}}$ *will render* $\text{int}(\mathcal{H}^{\mathcal{T}})$ *forward invariant.*

Proof. Let $x(\mathcal{T})$ be any solution to (4.1) and define $\theta(t) = h(t, x(t))$. Solutions to (4.1) must be absolutely continuous, so any solution starting at a point $(t_0, x(t_0)) \in \text{int}(\mathcal{H}^{\mathcal{T}})$ will satisfy $\theta(t_0) = h(t_0, x(t_0)) < 0$ and must pass a point where $\theta(t) = 0$ as it exits $\text{int}(\mathcal{H}^{\mathcal{T}})$. Moreover, θ satisfies $\dot{\theta} \leq \alpha(-\theta)$ by (4.8). Since α is locally Lipschitz continuous, in any neighborhood $[0, \epsilon]$ of the origin, the function α is linearly upper bounded as $\alpha(\lambda) \leq L\lambda$ for all $\lambda \in [0, \epsilon]$ for some Lipschitz constant $L \in \mathbb{R}_{>0}$. The system $\dot{z}(t) = -Lz(t)$ has solution $z(t) = z(t_0)e^{-L(t-t_0)}$, so if $z(t_0) < 0$, then $z(t) < 0$ for all $t \in \mathcal{T}$. Thus, by the comparison lemma [191, Lemma IX.2.6], any solution θ with $\theta(t_0) < 0$ will satisfy $\theta(t) < 0$ for all $t \in \mathcal{T}$. This is equivalent to the solution to (4.1) always staying in $\text{int}(\mathcal{H}^{\mathcal{T}})$ and thus rendering $\text{int}(\mathcal{H}^{\mathcal{T}})$ forward invariant. ■

That is, just by 1) choosing α in (4.8) to be locally Lipschitz continuous, and 2) choosing an initial state in the interior of the CBF set, one can avoid the need for the larger open set $\mathcal{D}$ in Theorem 4.1. Alternatively, one can also state a version of Theorem 4.1 with Lipschitz assumptions similar to Theorem 2.6 to achieve a similar result.

Theorem 4.3 (Extension of Theorem 2.6 to RCBFs)**.** *Let* $u : \mathcal{T} \times \mathcal{X} \to \mathbb{R}$ *be a control law, let* $h : \mathcal{T} \times \mathcal{X} \to \mathbb{R}$ *be an RCBF as in Definition 4.2 on the set* $\mathcal{H}$ *in* (4.3) *and for the dynamics* (4.1)*. Let* $\mathcal{D} : \mathcal{T} \to 2^{\mathcal{X}}$ *be an open set such that* $\partial\mathcal{H}(\tau) \subset \mathcal{D}(\tau)$ *for all* $\tau \in \mathcal{T}$ *and let* $\mathcal{D}^{\mathcal{T}} = \{(t,x) \in \mathcal{T} \times \mathcal{X} \mid x \in \mathcal{D}(t)\}$ *and let* $\alpha \in \mathcal{K}$*. Suppose that* w_u *and* w_x *are functions* $w_u : \mathcal{T} \times \mathcal{X} \to \mathbb{R}^m$ *and* $w_x : \mathcal{T} \times \mathcal{X} \to \mathbb{R}^n$*. Suppose that, for all* $(t,x) \in \mathcal{D}^{\mathcal{T}}$*, it holds that* f*,* g*,* u*,* w_u*,* w_x*,* $\partial_t h$*, and* ∇h *are locally Lipschitz continuous in* x *and piecewise locally Lipschitz continuous in* t*, with a minimum delay of* $T \in \mathbb{R}_{>0}$ *between discontinuities. Then any control law satisfying the above and* $u(t,x) \in \mathcal{U}_{\text{rcbf}}(t,x), \forall (t,x) \in \mathcal{H}^{\mathcal{T}}$ *will render* $\mathcal{H}^{\mathcal{T}}$ *forward invariant.*

As with Theorem 4.1, the proof of Theorem 4.3 is identical to the proof of Theorem 2.6, and a similar extension of Theorem 2.8 is as follows.

Theorem 4.4 (Extension of Theorem 2.8 to RCBFs)**.** *Let* $\mathcal{T}$ *be a closed set (as in Theorem 2.8), so that* $\mathcal{H}^{\mathcal{T}}$ *in* (2.3) *is a closed set. Given an RCBF as in Definition 4.2 on the set* $\mathcal{H}$ *in* (4.4) *and for the dynamics* (4.1)*, assume that* $|\partial_t h(t,x)| + \|\nabla h(t,x)\| \neq 0$ *for all* $(t,x) \in \mathcal{H}^{\mathcal{T}}$*, and let* $\alpha \in \mathcal{K}$*. Then any control law* $u : \mathcal{T} \times \mathcal{X}$ *satisfying* $u(t,x) \in \mathcal{U}_{\text{rcbf}}, \forall (t,x) \in \mathcal{H}^{\mathcal{T}}$ *will render* $\mathcal{H}^{\mathcal{T}}$ *forward invariant.*

Going forward, I will refer to the inclusion $u(t,x) \in \mathcal{U}_{\text{rcbf}}(t,x)$ in the above theorems as the *RCBF condition*. I continue to refer to $\mathcal{H}$ as simply the *CBF set*, not the *RCBF set*, as $\mathcal{H}$ has not changed any since $\mathcal{H}$ was first defined in (2.2). The consequence of the above theorems is that if one knows 1) h is an RCBF on $\mathcal{H}$, 2) the initial condition satisfies $x(t_0) \in \mathcal{H}(t_0)$, and 3) $\mathcal{H}(t) \subseteq \mathcal{S}(t), \forall t \in \mathcal{T}$, then one immediately knows that there exists a safe trajectory beginning at $(t_0, x(t_0))$ satisfying the input constraints.

4.2.4 Set Invariance for the Restricted CBF Set

Next, I consider the possibility that $\mathcal{H}(t) \not\subseteq \mathcal{S}(t)$, in which case I seek to extend the invariance theorems in the prior subsection to establish invariance of $\mathcal{H}_{\text{res}}$ instead. Define

$$\mathcal{H}_{\text{com}}(t) \triangleq \{x \in \mathcal{H}(t) \mid \kappa(t,x) = 0\}, \tag{4.9}$$

$$\mathcal{H}^{\mathcal{T}}_{\text{com}} \triangleq \{(t,x) \in \mathcal{H}^{\mathcal{T}} \mid \kappa(t,x) = 0\}, \tag{4.10}$$

$$\mathcal{H}_{\text{unsafe}}(t) \triangleq \{x \in \mathcal{H}(t) \mid \kappa(t,x) > 0\}, \tag{4.11}$$

$$\mathcal{H}^{\mathcal{T}}_{\text{unsafe}} \triangleq \{(t,x) \in \mathcal{H}^{\mathcal{T}} \mid \kappa(t,x) > 0\}. \tag{4.12}$$

An illustration of $\mathcal{H}_{\text{com}}$ and $\mathcal{H}_{\text{unsafe}}$ is shown in Fig. 4.2, specifically for the CBF that will be designed in Lemma 4.9. Recall that in the related high-order CBF literature [87–89], the CBF set is also defined as an intersection, so it is not surprising that, under certain conditions, one can also show that the RCBF condition is sufficient to render the intersection $\mathcal{H}_{\text{res}} = \mathcal{H} \cap \mathcal{S}$ forward invariant as well. I now present these additional conditions and extend the theorems in Section 4.2.3 to establish forward invariance of $\mathcal{H}_{\text{res}}$.

Corollary 4.5 (Corollary to Theorem 4.1)**.** *Suppose that the assumptions of Theorem 4.1 hold with $\mathcal{H}_{\text{res}}$ in place of $\mathcal{H}$. Suppose also that*

$$\max_{(w_u,w_x)\in\mathcal{W}} \dot{\kappa}(t,x,u(t,x),w_x,w_u) < 0, \forall (t,x) \in (\mathcal{D}^{\mathcal{T}} \cap \mathcal{H}^{\mathcal{T}}_{\text{unsafe}}) \cup (\text{int}(\mathcal{H}^{\mathcal{T}}) \cap \mathcal{H}^{\mathcal{T}}_{\text{com}}) \tag{4.13}$$

and

$$\max_{(w_u,w_x)\in\mathcal{W}} \dot{\kappa}(t,x,u(t,x),w_x,w_u) \leq 0, \forall (t,x) \in (\partial\mathcal{H}^{\mathcal{T}}) \cap (\partial\mathcal{S}^{\mathcal{T}}) \tag{4.14}$$

Then any control law $u : \mathcal{T} \times \mathcal{X}$ satisfying $u(t,x) \in \mathcal{U}_{\text{rcbf}}(t,x), \forall (t,x) \in \mathcal{D}^{\mathcal{T}}$ will render $\mathcal{H}^{\mathcal{T}}_{\text{res}}$ forward invariant.

Proof. Recall from Nagumo's theorem (Lemma 2.7) that a set is forward invariant if and only if trajectories never cross the set boundary. By the same argument as in Theorem 2.2, trajectories will never exit $\mathcal{H}^{\mathcal{T}}$ along the manifold $(\partial\mathcal{H}^{\mathcal{T}}) \cap \mathcal{S}^{\mathcal{T}}$. Moreover, since $\dot{\kappa}(t,x) < 0$ for all $(t,x) \in$

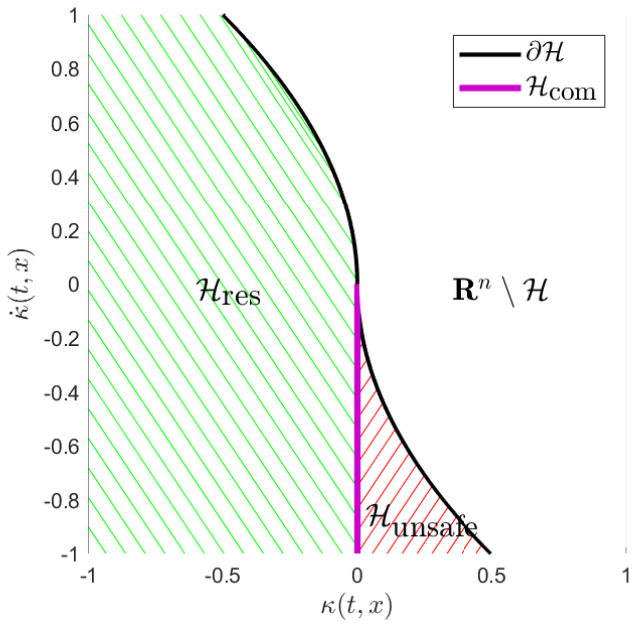

Figure 4.2: Illustration of Restricted CBF Set Boundary. An illustration of the sets $\mathcal{H}_{\text{res}}$ and $\mathcal{H}_{\text{unsafe}}$ and the manifold $\mathcal{H}_{\text{com}}$ between them in magenta, drawn for the CBF (4.17). Only states where $\kappa(t,x) \leq 0$ are safe, and only $\mathcal{H}_{\text{res}}$ is rendered forward invariant.

$\text{int}(\mathcal{H}^{\mathcal{T}}) \cap \mathcal{H}^{\mathcal{T}}_{\text{com}}$ in (4.13), trajectories will not exit $\mathcal{S}^{\mathcal{T}}$ along the manifold $\text{int}(\mathcal{H}^{\mathcal{T}}) \cap \mathcal{H}^{\mathcal{T}}_{\text{com}}$ either. Finally, at the manifold $(\partial\mathcal{H}^{\mathcal{T}}) \cap (\partial\mathcal{S}^{\mathcal{T}})$, I have already shown that trajectories will not leave $\mathcal{H}^{\mathcal{T}}$, and because $\dot{\kappa}(t,x) < 0$ for all $(t,x) \in \mathcal{D}^{\mathcal{T}} \cap \mathcal{H}^{\mathcal{T}}_{\text{unsafe}}$ in (4.13), trajectories are unable to exit $\mathcal{S}^{\mathcal{T}}$ at that manifold either. The union of these three manifolds forms the entire boundary of $\mathcal{H}^{\mathcal{T}}_{\text{res}}$. Thus, trajectories satisfying the corollary conditions can never leave $\mathcal{H}^{\mathcal{T}}_{\text{res}}$, so $\mathcal{H}^{\mathcal{T}}_{\text{res}}$ is forward invariant. ■

Next, if one is only concerned with the interior of $\mathcal{H}_{\text{res}}$, similar to Corollary 4.2, then one can drop condition (4.14), as follows.

Corollary 4.6 (Corollary to Corollary 4.2)**.** *For the system* (4.1)*, suppose* $h : \mathcal{T} \times \mathcal{X} \to \mathbb{R}$ *is an RCBF as in Definition 4.2 on the set* $\mathcal{H}_{\text{res}}$ *in* (4.4)*. Assume that* $\alpha \in \mathcal{K}_e$ *is locally Lipschitz continuous. Let* $\mathcal{D} : \mathcal{T} \to 2^{\mathcal{X}}$ *be a set (not necessarily open or closed) satisfying* $\mathcal{H}_{\text{res}}(\tau) \subseteq \mathcal{D}(\tau)$ *for all* $\tau \in \mathcal{T}$*, and suppose that* (4.13) *holds. Then any control law* $u : \mathcal{T} \times \mathcal{X}$ *satisfying* $u(t,x) \in \mathcal{U}_{\text{rcbf}}(t,x), \forall (t,x) \in \mathcal{H}^{\mathcal{T}}$ *will render* $\text{int}(\mathcal{H}^{\mathcal{T}}) \cap \text{int}(\mathcal{S}^{\mathcal{T}})$ *forward invariant.*

Proof. By condition (4.13), trajectories will never cross the manifold $\text{int}(\mathcal{H}^{\mathcal{T}}) \cap \mathcal{H}^{\mathcal{T}}_{\text{com}} \equiv \text{int}(\mathcal{H}^{\mathcal{T}}) \cap (\partial\mathcal{S}^{\mathcal{T}})$. Also, by the same logic as Corollary 4.2, trajectories starting inside $\text{int}(\mathcal{H}^{\mathcal{T}}) \cap \text{int}(\mathcal{S}^{\mathcal{T}})$ will never reach $(\partial\mathcal{H}^{\mathcal{T}}) \cap \text{int}(\mathcal{S}^{\mathcal{T}})$. Since trajectories cannot reach either of these two manifolds, any control law as described above will render $\text{int}(\mathcal{H}^{\mathcal{T}}) \cap \text{int}(\mathcal{S}^{\mathcal{T}})$ forward invariant. ■

Next, note that condition (4.14) does not make use of the larger open set $\mathcal{D}$, and that condition (4.13) only makes use of a subset of $\mathcal{D}$ that intersects with $\mathcal{H}$. Also, conditions (4.13)-(4.14) are only conditions on $\dot{\kappa}$, so if $\kappa \in \mathcal{G}^r$ for $r \geq 2$, then these conditions are independent of the control law. Thus, it is easy to make a similar extension to Theorem 4.3 and Theorem 4.4 as follows.

Corollary 4.7 (Corollary to Theorem 4.3)**.** *Suppose that the assumptions of Theorem 4.3 hold with $\mathcal{H}_{\text{res}}$ in place of $\mathcal{H}$. Let $\mathcal{D} : \mathcal{T} \to 2^{\mathcal{X}}$ be an open set satisfying $\mathcal{H}_{\text{res}}(\tau) \subset \mathcal{D}(\tau)$ for all $\tau \in \mathcal{T}$, and suppose that* (4.13)-(4.14) *hold. Then any control law $u : \mathcal{T} \times \mathcal{X}$ satisfying $u(t,x) \in \mathcal{U}_{\text{rcbf}}(t,x), \forall (t,x) \in \mathcal{H}^{\mathcal{T}}$ will render $\mathcal{H}^{\mathcal{T}}_{\text{res}}$ forward invariant.*

Proof. The proof follows the same logic as in Corollary 4.5. By the same argument as in Theorem 2.6, trajectories will never exit $\mathcal{H}^{\mathcal{T}}$ along the manifold $(\partial \mathcal{H}^{\mathcal{T}}) \cap \mathcal{S}^{\mathcal{T}}$. And by the same logic as in Corollary 4.5, trajectories will never exit $\mathcal{S}^{\mathcal{T}}$ along either of the manifolds $\text{int}(\mathcal{H}^{\mathcal{T}}) \cap \mathcal{H}^{\mathcal{T}}_{\text{com}}$ or $(\partial \mathcal{S}^{\mathcal{T}}) \cap (\partial \mathcal{H}^{\mathcal{T}})$. Thus, by the same logic as Corollary 4.5, the set $\mathcal{H}^{\mathcal{T}}_{\text{res}}$ is forward invariant. ■

Note that Corollary 4.7 still introduced an open set $\mathcal{D}$ as was done in Theorem 2.2 and Corollary 4.5, but this set $\mathcal{D}$ is only used to define the set where $\dot{\kappa}$ is required to be negative. The control law in Corollary 4.7 still only needs to satisfy the RCBF condition on $\mathcal{H}^{\mathcal{T}}$, not on the larger open set $\mathcal{D}^{\mathcal{T}}$. I chose this proof strategy because 1) I did not want to complicate the corollary conditions by introducing a Lipschitz assumption on the evolution of κ (which could have replaced the need for introducing $\mathcal{D}$ entirely), and 2) the assumption on $\dot{\kappa}$ in both Corollary 4.5 and Corollary 4.7 will be automatically satisfied by both of the RCBFs in Theorem 4.10 and Theorem 4.11, so these conditions are not restrictive.

Finally, I present a similar extension of Theorem 4.4.

Corollary 4.8 (Corollary to Theorem 4.4)**.** *Let $\mathcal{T}$ be a closed set (as in Theorem 2.8) so that $\mathcal{H}^{\mathcal{T}}_{\text{res}}$ is a closed set. Given an RCBF as in Definition 4.2 on the set $\mathcal{H}_{\text{res}}$ in* (4.4)*, assume that $|\partial_t h(t,x)| + \|\nabla h(t,x)\| \neq 0$ for all $(t,x) \in (\partial \mathcal{H}^{\mathcal{T}}) \cap \mathcal{S}^{\mathcal{T}}$ and that $|\partial_t \kappa(t,x)| + \|\nabla \kappa(t,x)\| \neq 0$ for all $(t,x) \in (\partial \mathcal{S}^{\mathcal{T}}) \cap \mathcal{H}^{\mathcal{T}}$ Suppose that $\max_{(w_u,w_x)\in\mathcal{W}} \dot{\kappa}(t,x,u(t,x),w_u,w_x) \leq 0$ for all $(t,x) \in \mathcal{H}^{\mathcal{T}}_{\text{com}}$. Then any control law satisfying $u(t,x) \in \mathcal{U}_{\text{rcbf}}(t,x)$ for all $(t,x) \in \mathcal{H}^{\mathcal{T}}_{\text{res}}$ will render $\mathcal{H}^{\mathcal{T}}_{\text{res}}$ forward invariant.*

Proof. By the assumed conditions and following the same logic as the proof of Theorem 2.8, trajectories can never leave $\mathcal{H}^{\mathcal{T}}$ via the manifold $(\partial \mathcal{H}^{\mathcal{T}}) \cap \mathcal{S}^{\mathcal{T}}$. Also, due to the condition on $|\partial_t \kappa| + \|\nabla \kappa\|$ and the assumption that $\dot{\kappa}(t,x) \leq 0$ for all $(t,x) \in \mathcal{H}^{\mathcal{T}}_{\text{com}}$, Lemma 2.7 implies that trajectories can never leave $\mathcal{S}^{\mathcal{T}}$ via the manifold $\mathcal{H}^{\mathcal{T}}_{\text{com}}$. These two manifolds cover the entire boundary of $\mathcal{H}_{\text{res}}$, so $\mathcal{H}_{\text{res}}$ is rendered forward invariant by the same logic as in Theorem 2.8. ■

Note that I present Corollary 4.8 for completeness, though in practice, both of the RCBFs in Theorem 4.10 and Theorem 4.11 violate the conditions of Corollary 4.8, because for these two RCBFs, it always holds that $|\partial_t \kappa(t,x)| + \|\nabla \kappa(t,x)\| = 0$ for (t,x) such that $h(t,x) = \kappa(t,x) = 0$. One open area for future work is determining under what conditions one can prove forward invariance as in Corollary 4.8 without explicitly requiring this condition at these specific states. Intuitively, Fig. 4.2 suggests that $\mathbf{T}_{\mathcal{H}\mathcal{T}}(t,x) = \mathbf{T}_{\mathcal{S}\mathcal{T}}(t,x)$ at the points (t,x) such that $h(t,x) = \kappa(t,x) = 0$. If this holds, then one could relax the requirement that $|\partial_t \kappa(t,x)| + \|\nabla \kappa(t,x)\| \neq 0$ at these specific points, but this intuitive conjecture still needs to be proven.

Now that I have established the conditions under which one can prove forward invariance of both $\mathcal{H}$ and $\mathcal{H}_{\text{res}}$, in the next section I present the specific forms of RCBFs with which I am interested in using the above theorems.

4.3 Robust Control Barrier Functions for Input Constraints

Given some constraint function κ and associated safe set $\mathcal{S}$, along with input constraints $\mathcal{U}$ and disturbances $\mathcal{W}$, the goal of this section is to develop an RCBF h and associated sets $\mathcal{H}$ in (4.3) and $\mathcal{H}_{\text{res}}$ in (4.4) for dynamics relevant to spacecraft, namely relative-degree 2 dynamics. The primary challenge addressed by all the subsequent methods is that for constraint functions with high relative-degree, the input constraints and disturbances must also be incorporated into the form of the RCBF. For example, if $\kappa \in \mathcal{G}^2$, there may exist a state $(t_0, x(t_0)) \in \mathcal{S}(t_0)$ such that $\kappa(t_0, x(t_0)) < 0$ and $\dot{\kappa}(t_0, x(t_0)) > 0$. Without loss of generality, κ can be thought of as the position of an agent, $\dot{\kappa}$ its velocity, and $\ddot{\kappa}$ its acceleration. This state can be in the zero sublevel set $\mathcal{H}$ of some RCBF h only if there exists a control input satisfying the input constraints such that the agent decelerates to $\dot{\kappa}(t, x(t)) = 0$ before leaving $\mathcal{S}$. Thus, an RCBF must be a function of the input constraints, and this section seeks systematic methods of finding such functions. The following subsections identify three forms of RCBFs, presented in order of increasing complexity and decreasing conservatism. Each form is developed as a function of κ and is applicable under different conditions on the input constraints and disturbances. Sections 4.3.1 and 4.3.2 consider only relative-degree $r = 2$ constraint functions and require specific system properties, while Section 4.3.3 provides an approach for relative-degree $r \geq 2$ with distinct requirements.

4.3.1 Constant Control Authority

In this subsection, suppose the constraint function κ is of relative-degree $r = 2$, and has the special property that $\ddot{\kappa}$ can always be made less than some negative constant, similar to (3.24). Intuitively, this means that the controller can add/remove "energy" from the system at a constant

rate. Then, assuming a time-invariant system and no disturbances, Example 3.1 provides one possible form of CBF, repeated here as follows

$$h(x) = \begin{cases} \kappa(x) & \dot{\kappa}(x) < 0 \\ \kappa(x) + \dfrac{\dot{\kappa}(x)^2}{2a_{\max}} & \dot{\kappa}(x) \geq 0 \end{cases} \tag{4.15}$$

where $a_{\max}$ is a parameter.

Next, note that (4.15) can be modified to remove the piecewise definition. The following alternate form of (4.15) provides a more intuitive metric for safety, especially when disturbances are added.

Lemma 4.9. *Suppose $w_u \equiv w_x \equiv 0$. If $\kappa \in \mathcal{G}^2$, f, g, and κ are time-invariant, and there exists $a_{\max}$ such that*

$$\inf_{u \in \mathcal{U}} \ddot{\kappa}(x,u) \leq -a_{\max}, \forall x \in \mathcal{S}\,, \tag{4.16}$$

then the function

$$h(x) = \kappa(x) + \frac{|\dot{\kappa}(x)|\dot{\kappa}(x)}{2a_{\max}} \tag{4.17}$$

is a CBF on $\mathcal{H}_{\text{res}}$ in (4.4) as in Definition 4.1 for any $\alpha \in \mathcal{K}_e$. Moreover, Theorem 2.2 and Theorem 2.6 hold with $\mathcal{H}_{\text{res}}$ in place of $\mathcal{H}$.

Proof. First note that h in (4.17) is continuously differentiable (recall that $z : \mathbb{R} \to \mathbb{R}$, $z(\lambda) = \lambda|\lambda|$ is once continuously differentiable). Next, note that the derivative of h is

$$\dot{h}(x,u) = \dot{\kappa}(x) + \frac{|\dot{\kappa}(x)|\ddot{\kappa}(x,u)}{a_{\max}}\,. \tag{4.18}$$

When $\dot{\kappa}(x) \leq 0$, (4.16) implies that one can always choose a u that renders (4.18) nonpositive. When $\dot{\kappa}(x) > 0$, (4.18) reduces to $\dot{h}(x,u) = \dot{\kappa}(x)\left(1 + \frac{\ddot{\kappa}(x,u)}{a_{\max}}\right)$. Any u such that $\ddot{\kappa}(x,u) \leq -a_{\max}$ will thus render $\dot{h}$ nonpositive, and by assumption in (4.16), such a u always exists in $\mathcal{U}$ for all $x \in \mathcal{S}$. Thus, for every $x \in \mathcal{H}_{\text{res}} \subseteq \mathcal{S}$, there exists $u(x) \in \mathcal{U}$ such that $\dot{h}(x,u(x)) \leq 0 \leq \alpha(-h(x))$ for any $\alpha \in \mathcal{K}_e$, so h satisfies the conditions of a CBF on $\mathcal{H}_{\text{res}}$ in Definition 4.1 for any $\alpha \in \mathcal{K}_e$.

Next, note that for h as in (4.17), the set $\mathcal{H}$ in (4.3) is not a subset of $\mathcal{S}$ in (4.2); i.e., there exist states $x \in \mathcal{H}$ where $h(x) \leq 0$ and $\kappa(x) > 0$, as shown by the red hashed region in Fig. 4.2. Note that no such states occurred previously for h as in (4.15) due to the piecewise definition. Thus, one can divide the set $\mathcal{H}$ into two disjoint subsets, the restricted CBF set $\mathcal{H}_{\text{res}}$ as in (4.4) and the set $\mathcal{H}_{\text{unsafe}}$ as in (4.11), which is unsafe. Denote the manifold between these two subsets as $\mathcal{H}_{\text{com}}$ in (4.9), illustrated in Fig. 4.2. It follows from Theorem 2.2 and Theorem 2.6 that closed-loop trajectories satisfying $u(t,x) \in \mathcal{U}_{\text{rcbf}}(t,x)$ cannot transition directly from $\mathcal{H}_{\text{res}}$ to $\mathcal{X} \setminus \mathcal{H}$ (i.e. cannot

cross the black line in Fig. 4.2). I next show that trajectories also cannot transition directly from $\mathcal{H}_{\text{res}}$ to $\mathcal{H}_{\text{unsafe}}$.

A trajectory starting in $\mathcal{H}_{\text{res}}$ can only transition directly from $\mathcal{H}_{\text{res}}$ to $\mathcal{H}_{\text{unsafe}}$ along $\mathcal{H}_{\text{com}}$ (i.e. cross the magenta line in Fig. 4.2). Divide $\mathcal{H}_{\text{com}}$ into two disjoint sets $\mathcal{H}_{\text{c},1} = \{x \in \mathcal{H}_{\text{com}} \mid h(x) < 0\}$ and $\mathcal{H}_{\text{c},2} = \{x \in \mathcal{H}_{\text{com}} \mid h(x) = 0\}$. For all $x \in \mathcal{H}_{c,1}$, it holds that $\dot{\kappa}(x) < 0$, so, under the regularity conditions of Theorem 2.2 or Theorem 2.6, it follows that trajectories can never cross the manifold $\mathcal{H}_{c,1}$ from inside $\mathcal{H}_{\text{res}}$. Similarly, the need for an open set $\mathcal{D}$ in Theorem 2.2 prevents trajectories from crossing the manifold $\mathcal{H}_{c,2}$. Additionally, for all $x \in \mathcal{H}_{c,2}$, it holds that $\dot{\kappa}(x) = 0$. Note that any controller satisfying $u(t,x) \in \mathcal{U}_{\text{rcbf}}(t,x)$ at $x \in \{x \in \mathcal{H} \mid h(x) = 0 \text{ and } \dot{\kappa}(x) > 0\}$ must yield $\ddot{\kappa}(x) \leq -a_{\text{max}}$. By the regularity assumptions in Theorem 2.6, it follows that $\ddot{\kappa}(x) \leq -a_{\text{max}} < 0, \forall (x) \in \mathcal{H}_{c,2}$, so trajectories can never cross the manifold $x \in \mathcal{H}_{c,2}$ under the conditions of Theorem 2.6 either. Thus, trajectories starting in $\mathcal{H}_{\text{res}}$ never directly transition across $\mathcal{H}_{\text{com}} = \mathcal{H}_{c,1} \cap \mathcal{H}_{c,2}$ to $\mathcal{H}_{\text{unsafe}}$. Since closed-loop trajectories can never transition from $\mathcal{H}_{\text{res}}$ to either $\mathcal{H}_{\text{unsafe}}$ or $\mathcal{X} \setminus \mathcal{H}$, it follows that $\mathcal{H}_{\text{res}}$ is rendered forward invariant. ■

Remark 4.2. *If* (4.16) *holds for all* $x \in \mathcal{H}$*, then any controller satisfying* $u(t,x) \in \mathcal{U}_{\text{rcbf}}(t,x), \forall (t,x) \in \mathcal{H}^{\mathcal{T}}$ *also necessarily causes trajectories originating in* $\mathcal{H}_{\text{unsafe}}$ *to approach* $\mathcal{H}_{\text{res}}$ *(see also [89, Rem. 2]).*

Note that by definition $\mathcal{H}_{\text{res}}$ is a subset of $\mathcal{S}$, so rendering $\mathcal{H}_{\text{res}}$ forward invariant also ensures safety for all future time. Moreover, Lemma 4.9 says that the same CBF conditions as in Section 2.3.2 causes $\mathcal{H}_{\text{res}}$ to be rendered forward invariant (assuming no disturbances). In fact, the set $\mathcal{H}_{\text{res}}$ for the CBF in (4.17) is identical to the set $\mathcal{H}$ for the CBF in (4.15), but the form of (4.17) is mathematically more convenient. In particular, the CBF in (4.17) is of relative-degree 1 everywhere, and captures the rate at which κ decreases, unlike the piecewise form in (4.15). Note also that Lemma 4.9 does not include an extension of Theorem 2.8, because, as noted in Section 4.2.4, the conditions of Theorem 2.8 may not hold at $x \in \mathcal{H}_{c,2}$.

Note that the function (4.17) by construction satisfies conditions (4.13) and (4.14), as will the next two RCBFs introduced. Thus, one can come to the same conclusion as in Theorem 4.9 that $\mathcal{H}_{\text{res}}$ can be substituted for $\mathcal{H}$ by noting that, with this substitution, Theorem 2.2 and Theorem 2.6 become special cases of Corollary 4.5 and Corollary 4.7, respectively. The above serves as an alternate proof of this result. Note that one could still define the upcoming RCBF in this section with the piecewise definition in (4.15); i.e. the switch from (4.15) to (4.17) is not necessary for the main result of this section to hold; this switch is just helpful for interpretability of the CBF metric and for consistency when tuning the controller.

Next, I add disturbances to the CBF in (4.17). Note that (4.17) is a function of the constraint function derivative $\dot{\kappa}$, and that in the case of unmatched disturbances (i.e. when $w_x \neq 0$), $\dot{\kappa}$ is a

function of the disturbance w_x and thus not exactly known. I define the following upper bound on $\dot{\kappa}$:

$$\begin{aligned}\dot{\kappa}_w(t,x) &\triangleq \max_{\|w_x\|\le w_{x,\max}} \dot{\kappa}(t,x,w_x) \\ &= \partial_t\kappa(t,x) + \nabla\kappa(t,x)f(t,x) + \|\nabla\kappa(t,x)\|w_{x,\max}\end{aligned} \tag{4.19}$$

(recall that κ is relative-degree 2, so $\nabla\kappa(t,x)g(t,x) \equiv 0$ and thus $\dot{\kappa}$ does not depend on u, w_u) and its derivative

$$\begin{aligned}\ddot{\kappa}_w(t,x,u,w_u,w_x) &= \frac{d}{dt}\dot{\kappa}_w(t,x) \\ &= \partial_t\dot{\kappa}_w(t,x) + \nabla\dot{\kappa}_w(t,x)F(t,x,u,w_u,w_x)\,.\end{aligned} \tag{4.20}$$

Note that $\dot{\kappa}_w$ is a known quantity, while $\ddot{\kappa}_w$ is a function of the unknown quantities w_u, w_x in F in (4.1). Assume that $\|\nabla\kappa\|$ does not vanish, so that $\dot{\kappa}_w$ in (4.19) is differentiable (note that $\|\nabla\kappa\| \equiv 1$ in Section 4.5). I now present the robust formulation of (4.15),(4.17).

Theorem 4.10. *Suppose $\kappa \in \mathcal{G}^2$ and there exists $a_{\max} > 0$ such that $\forall (t,x) \in \mathcal{S}^{\mathcal{T}}$,*

$$\max_{(w_u,w_x)\in\mathcal{W}} \Big(\inf_{u\in\mathcal{U}} \ddot{\kappa}_w(t,x,u,w_u,w_x)\Big) \le -a_{\max}\,. \tag{4.21}$$

Then the function

$$h(t,x) = \kappa(t,x) + \frac{|\dot{\kappa}_w(t,x)|\dot{\kappa}_w(t,x)}{2a_{\max}} \tag{4.22}$$

is an RCBF on $\mathcal{H}_{\text{res}}$ in (4.4) *for the system* (4.1) *for any $\alpha \in \mathcal{K}_e$. Moreover,* (4.22) *satisfies conditions* (4.13)-(4.14) *for any control law $u : \mathcal{T} \times \mathcal{X} \to \mathcal{U}$.*

Proof. As in Lemma 4.9, h is continuously differentiable, and its derivative satisfies

$$\begin{aligned}\dot{h}(t,x,u,w_u,w_x) &= \dot{\kappa}(t,x,w_x) + \frac{|\dot{\kappa}_w(t,x)|\ddot{\kappa}_w(t,x,u,w_u,w_x)}{a_{\max}} \\ &\overset{(4.19)}{\le} \dot{\kappa}_w(t,x) + \frac{|\dot{\kappa}_w(t,x)|\ddot{\kappa}_w(t,x,u,w_u,w_x)}{a_{\max}}\,.\end{aligned} \tag{4.23}$$

By assumption in (4.21), there exists $u \in \mathcal{U}$ independent of the disturbances w_u, w_x such that $\max_{(w_u,w_x)\in\mathcal{W}} \ddot{\kappa}_w(t,x,u,w_u,w_x) \le -a_{\max}$. Such a u will render the right hand side of (4.23) nonpositive similar to (4.18) in Lemma 4.9, and therefore will also render $\max_{(w_u,w_x)\in\mathcal{W}} \dot{h}(t,x,u,w_u,w_x)$ nonpositive. Thus, condition (4.6) is satisfied for any $\alpha \in \mathcal{K}_e$ and all $(t,x) \in \mathcal{H}_{\text{res}}^{\mathcal{T}}$, so h is an RCBF on $\mathcal{H}_{\text{res}}$ for any $\alpha \in \mathcal{K}_e$. Next, since κ is of relative-degree $r = 2$ and $\dot{\kappa}_w(t,x) < 0$ for all $(t,x) \in \mathcal{H}_{\text{unsafe}}$ in (4.11), there exists an open set $\mathcal{D}$ for which κ will

satisfy (4.13)-(4.14) regardless of the control law. ■

Thus, I have presented the first of three methods in this chapter for rendering trajectories always inside sublevel sets of relative-degree $r = 2$ constraint functions under input constraints and disturbances, by constructing a general form of RCBF h in (4.22) as a function of the constraint function κ, the input constraints (encoded in $a_{\max}$), and the disturbance bounds. Note that Theorem 4.10 showed that (4.6) is satisfied for any $\alpha \in \mathcal{K}_e$, so Section 4.4 will suggest a choice for the free parameter $\alpha \in \mathcal{K}_e$. The RCBF in Theorem 4.10 is particularly useful for systems similar to the double integrator, as illustrated by the simulations in Section 3.4.1. It is also easy to check if this method is applicable to a system or not. The largest allowable $a_{\max}$ is

$$a_{\max,0} \triangleq -\max_{(t,x)\in\mathcal{S}^{\mathcal{T}}} \big[\partial_t \dot{\kappa}_w(t,x) + \nabla\dot{\kappa}_w(t,x)(f(t,x)+g(t,x)u_{\min}(t,x)) + \|\nabla\dot{\kappa}_w(t,x)\| w_{x,\max} + \|\nabla\dot{\kappa}_w(t,x)g(t,x)\| w_{u,\max}\big], \quad (4.24)$$

where

$$u_{\min}(t,x) = \arg\min_{u\in\mathcal{U}} \nabla\dot{\kappa}_w(t,x)g(t,x)u\,. \quad (4.25)$$

Note that (4.24) can be solved offline. If $a_{\max,0} > 0$, then any $a_{\max} \in (0, a_{\max,0}]$ satisfies the requirements of Theorem 4.10, while if $a_{\max,0} < 0$, then no such $a_{\max}$ exists for the particular κ. The form of RCBF in (4.22) is constructive, since if an $a_{\max} > 0$ exists, then one knows that a subset $\mathcal{H}_{\text{res}} \subseteq \mathcal{S}$ can be rendered forward invariant, and one has explicit expressions for the restricted CBF set $\mathcal{H}_{\text{res}}$ in (4.4) and the set of safe control inputs $\mathcal{U}_{\text{rcbf}}$ in (4.8).

However, even if an $a_{\max} > 0$ does exist, this method can be overly conservative. That is, there may exist $(t,x) \in \mathcal{S}^{\mathcal{T}}$ such that (t,x) can be robustly rendered inside $\mathcal{S}$ but (t,x) is not in $\mathcal{H}_{\text{res}}^{\mathcal{T}}$ with h as in (4.22), as illustrated in simulation in Section 4.5. The following section describes an alternative to Theorem 4.10 that aims at reducing conservatism, and which further treats certain cases where no $a_{\max}$ exists.

4.3.2 Variable Control Authority

Using the methodology in the prior section, the set $\mathcal{H}_{\text{res}}$ only includes states that can be rendered always inside $\mathcal{S}$ by setting $\ddot{\kappa}_w$ equal to a negative constant $-a_{\max}$. For agents with a wide operating range, such as a spacecraft operating at various altitudes, this restriction may result in an overly conservative set $\mathcal{H}_{\text{res}}$, or no such $a_{\max}$ may exist. On the other hand, in the spacecraft scenario, the gravity of a central attractive body is known to high precision at every altitude. Similarly, electric/magnetic field strengths, spring force, buoyancy force, and aircraft lift/drag forces are well-characterized across operating ranges. Thus, instead of assuming that $\ddot{\kappa}_w$ is upper bounded

by a constant as in (4.21), suppose it is upper bounded by a known function, denoted ϕ. This is the idea central to the following theorem.

Theorem 4.11. *Let $\kappa \in \mathcal{G}^2$ define a safe set as in* (4.2)*. Suppose there exists an invertible, strictly monotone decreasing, and continuously differentiable function $\Phi : \mathbb{R} \to \mathbb{R}$, whose derivative is $\Phi' = \phi$ for $\phi : \mathbb{R} \to \mathbb{R}$, such that*

$$\max_{(w_u, w_x) \in \mathcal{W}} \inf_{u \in \mathcal{U}} \ddot{\kappa}_w(t, x, u, w_u, w_x) \leq \phi(\kappa(t, x)) < 0, \forall (t, x) \in \mathcal{S}^{\mathcal{T}} . \tag{4.26}$$

Let Φ^{-1} be the function for which $\Phi^{-1}(\Phi(\lambda)) = \lambda, \forall \lambda \in \mathbb{R}$. Then the function

$$h(t, x) = \Phi^{-1}\left(\Phi(\kappa(t, x)) - \frac{1}{2}\dot{\kappa}_w(t, x)|\dot{\kappa}_w(t, x)|\right) \tag{4.27}$$

is an RCBF on $\mathcal{H}_{\text{res}}$ in (4.4) *for the system* (4.1) *for any $\alpha \in \mathcal{K}_e$. Moreover,* (4.22) *satisfies conditions* (4.13)-(4.14) *for any control law $u : \mathcal{T} \times \mathcal{X} \to \mathcal{U}$.*

Proof. The function h is an RCBF if $\dot{h}$ satisfies the condition (4.6). By the chain rule, the total derivative of h is

$$\begin{aligned} \dot{h} &= (\Phi^{-1})'\left(\Phi(\kappa) - \tfrac{1}{2}\dot{\kappa}_w|\dot{\kappa}_w|\right)\left(\Phi'(\kappa)\dot{\kappa} - |\dot{\kappa}_w|\ddot{\kappa}_w\right) \\ &\overset{(4.27)}{=} (\Phi^{-1})'(\Phi(h))\left(\Phi'(\kappa)\dot{\kappa} - |\dot{\kappa}_w|\ddot{\kappa}_w\right) \end{aligned} \tag{4.28}$$

where the arguments of $\kappa, \dot{\kappa}_w, h$ are omitted for brevity. By definition, $\Phi'(\cdot) = \phi(\cdot)$, and by the Inverse Function Theorem, $(\Phi^{-1})'(\Phi(\cdot)) = \frac{1}{\phi(\cdot)}$, so (4.28) becomes

$$\dot{h} = \frac{1}{\phi(h)}\left(\phi(\kappa)\dot{\kappa} - |\dot{\kappa}_w|\ddot{\kappa}_w\right) \overset{(4.19)}{\leq} \frac{1}{\phi(h)}\left(\phi(\kappa)\dot{\kappa}_w - |\dot{\kappa}_w|\ddot{\kappa}_w\right). \tag{4.29}$$

By assumption in (4.26), there exists $u \in \mathcal{U}$ independent of w_u, w_x such that $\max_{(w_u, w_x) \in \mathcal{W}} \ddot{\kappa}_w(t, x, u, w_u, w_x) \leq \phi(\kappa(t, x))$. Since $\phi(h) \leq 0$, it follows that such a u will render $\max_{(w_u, w_x) \in \mathcal{W}} \dot{h}(t, x, u, w_u, w_x)$ with $\dot{h}$ as derived above nonpositive. Since this holds independent of w_u, w_x, condition (4.6) is satisfied for any $\alpha \in \mathcal{K}_e$ and all $(t, x) \in \mathcal{H}_{\text{res}}^{\mathcal{T}}$, so h is an RCBF on $\mathcal{H}_{\text{res}}$ for any $\alpha \in \mathcal{K}_e$.

As in Lemma 4.9 and Theorem 4.10, for h as in (4.27), it holds that $\mathcal{H}(t) \not\subseteq \mathcal{S}(t)$. Following

the logic of these two theorems, $\forall (t,x) \in \mathcal{H}_{\text{res}}^{\mathcal{T}}$, it holds that

$$\begin{aligned} 0 \geq & \Phi^{-1}\left(\Phi(\kappa(t,x)) - \frac{1}{2}\dot{\kappa}_w(t,x)|\dot{\kappa}_w(t,x)|\right), \\ \Phi(0) \leq & \Phi(\kappa(t,x)) - \frac{1}{2}\dot{\kappa}_w(t,x)|\dot{\kappa}_w(t,x)|, \end{aligned} \tag{4.30}$$

since Φ is monotone decreasing. If $x \in \mathcal{H}_{\text{com}}(t)$, then $\kappa(t,x) = 0$ and (4.30) reduces to $\dot{\kappa}_w(t,x) \leq 0$. Thus, the function (4.27) also satisfies (4.13)-(4.14) for any control law $u : \mathcal{T} \times \mathcal{X} \to \mathcal{U}$. ■

Remark 4.3. *In most cases, the function ϕ is derived from the dynamics (e.g. ϕ might represent a potential force that can be read directly from the equations of motion), in which case Φ can be any anti-derivative of ϕ. The results are invariant under different constants of integration. For instance, gravity may be described by $\phi(\lambda) = -\frac{\mu}{\lambda^2}$, in which case either $\Phi(\lambda) = \frac{\mu}{\lambda}$ or $\Phi(\lambda) = \frac{\mu}{\lambda} - \frac{\mu}{\lambda_0}$ for fixed λ_0 meets the requirements of Theorem 4.11.*

Thus, I have presented the second form of RCBF in this chapter that is applicable to relative-degree $r = 2$. Intuitively, the function Φ in (4.26) is usually a potential field, in which states below a certain potential value are unsafe. One can think of the argument of Φ^{-1} in (4.27) as analogous to the sum of potential energy $-\Phi(\kappa)$ and kinetic energy $\frac{1}{2}\dot{\kappa}_w^2$; the inverse of this quantity provides an "effective constraint value" h in the same units as the original constraint κ. Using this analogy, Theorem 4.11 gives a condition for ensuring an agent moving in this potential field never falls below the minimum safe potential threshold. If an expression for potential energy is known, then finding Φ is often straightforward, as shown in Section 4.5, but this may be difficult otherwise. There is no general formula for Φ as there was for $a_{\max}$ in (4.24), though Φ can possibly be learned [157].

The conditions (4.21) and (4.26) may appear to be restrictive assumptions, but are reasonable for many systems. Similar assumptions are implicit in [87, Eqs. 11-14] and [89, Eq. 16], except that the assumptions in [87, 89] only apply to $\mathcal{H}_{\text{res}}$ rather than to all of $\mathcal{S}$, and are relaxed in $\text{int}(\mathcal{H}_{\text{res}})$ by using one additional class-$\mathcal{K}_e$ function. Similar relaxations can be made in Theorem 4.10 and Theorem 4.11, but then the theorems would be less constructive, since one would need to know $\mathcal{H}_{\text{res}}$ in advance and would need to be able to find an appropriate class-$\mathcal{K}_e$ function; instead, Theorem 4.10 and Theorem 4.11 assume stricter conditions than [87, 89] so that any class-$\mathcal{K}_e$ function will work. In fact, h as in (4.27) can be expressed using the conventions of [87, 89] as $\psi_1(t,x) = \dot{\kappa}_w(t,x) - \sqrt{2(\Phi(\kappa(t,x)) - \Phi(0))}$. However, in this case, ψ_1 no longer has an interpretation as energy, and this form is problematic for the robustness strategy in [89] because the square-root function is not differentiable at the origin.

At time of writing, the Exponential CBF (ECBF) has become a standard method to construct CBFs for high relative-degree constraint functions [85, 138, 210, 211], and a similar RCBF exten-

sion exists. Suppose that the function g in (4.1) is a constant, as occurs in many systems. In this case, one downside to the ECBF is that it can only be defined with input constraints for compact $\mathcal{S}$. For non-compact $\mathcal{S}$, the ECBF definition will require increasing control authority along $\partial\mathcal{H}$ as the state goes further from $\partial\mathcal{S}$, thus requiring $\mathcal{U}$ to be unbounded if $\mathcal{S}$ is also unbounded. By contrast, the forms of CBF (4.22) and (4.27) are potentially applicable with input constraints even to non-compact CBF sets.

4.3.3 General Case Control Authority

I now consider the case where a function Φ may not exist, or may still yield overly conservative results. To accomplish this, suppose there is a known control law u^*, called the *nominal evading maneuver* in [86], which encourages safety by driving the agent towards the interior of $\mathcal{S}$. For instance, u^* can be a control law which drives an agent away from an obstacle, but which does not necessarily provide for stability or convergence to an objective, such as the control law in (4.58). The core idea of this subsection is to propagate the state forward in time using the nominal evading maneuver, and analyze the resultant trajectory for safety. If the propagated trajectory from $(t_0, x(t_0))$ is always in $\mathcal{S}$, then I conclude that $(t_0, x(t_0))$ should be in the CBF set for some CBF to be defined. This was formalized mathematically for the case without disturbances in Theorem 3.4 (see also [86, Thm. 2], though that is more restrictive).

Note that Theorem 3.4 required an extended notion of CBF because (3.12) may not be continuously differentiable; this extended definition is not used subsequently to avoid confusion in Section 4.4. The extension of Theorem 3.4 to the time-varying case is straightforward, but accounting for the presence of disturbances introduces several more restrictions, as I will show. Note that this section allows for constraint functions κ of arbitrary relative-degree unlike Sections 4.3.1-4.3.2.

For clarity, in this section, I will use the notation $\frac{d\kappa}{dt}, \frac{\partial\kappa}{\partial t}, \frac{\partial\kappa}{\partial x}$ instead of $\dot{\kappa}, \partial_t\kappa, \nabla\kappa$, respectively. Suppose existence of a control law (the proposed nominal evading maneuver) $u^* : \mathcal{T}\times\mathcal{X} \to \mathcal{U}$ that is sufficiently regular to yield unique system solutions. Define the function $\chi : \mathbb{R}_{\geq 0}\times\mathcal{T}\times\mathcal{X} \to \mathcal{X}$ (this was previously ψ_x in Section 3.3.1) as the solution to the following initial value problem

$$
\begin{aligned}
&\chi(\beta, t_0, x_0) = y(\beta), \text{ where } y(0) = x_0, \Theta(0) = I, \theta(0) = 0,\\
&Y(\beta, t_0, y) \triangleq f(t_0+\beta, y) + g(t_0+\beta, y)u^*(t_0+\beta, y)\,,\\
&\frac{dy}{d\beta} = Y(\beta, t_0, y)\,,\\
&\frac{d\theta}{d\beta} = \frac{\partial Y(\beta, t_0, y)}{\partial t_0}\,,\\
&\frac{d\Theta}{d\beta} = \frac{\partial Y(\beta, t_0, y)}{\partial y}\Theta\,.
\end{aligned}
\tag{4.31}
$$

Here, (t_0, x_0) is some given state, β is the amount of time in the future from t_0 that one wishes to propagate the trajectory, and χ is the propagated state. Since β is "time since t_0", I use the absolute time $t_0 + \beta$ in the arguments of f, g, u^* in (4.31). The solution to (4.31) also defines θ and Θ, which I will show are the sensitivities (i.e. derivatives) of $\chi(\beta, t_0, x_0)$ with respect to t_0 and x_0, respectively. Let $\theta(\beta, t_0, x_0)$ and $\Theta(\beta, t_0, x_0)$ denote the values of θ and Θ, respectively, at β. To determine safety, one is interested in the values of $\kappa(t_0 + \beta, \chi(\beta, t_0, x(t_0)))$ for various times in the future β.

First, I make some remarks about the derivatives of χ from (4.31) in the following lemma.

Lemma 4.12. *Suppose f, g, u^* are all continuously differentiable, and χ in* (4.31) *exists everywhere in a neighborhood of β, t_0, x_0. Then the partial derivatives of χ are*

$$\frac{\partial \chi(\beta, t_0, x_0)}{\partial \beta} = Y(\beta, t_0, \chi(\beta, t_0, x_0)) \,, \tag{4.32}$$

$$\frac{\partial \chi(\beta, t_0, x_0)}{\partial t_0} = \theta(\beta, t_0, x_0) \,, \tag{4.33}$$

$$\frac{\partial \chi(\beta, t_0, x_0)}{\partial x_0} = \Theta(\beta, t_0, x_0) \,. \tag{4.34}$$

Proof. First, the partial derivative with respect to β in (4.32) follows immediately from (4.31), since y in (4.31) evolves with respect to β.

Second, note that at $\beta = 0$, $\chi(0, t_0, x_0) = x_0$, so $\frac{\partial}{\partial t_0}[\chi(0, t_0, x_0)] = 0$. Also, since f, g, u^* are continuously differentiable, one can describe the evolution of the partial derivative with β as follows

$$\frac{d}{d\beta}\left[\frac{\partial}{\partial t_0}[\chi]\right] = \frac{\partial}{\partial t_0}\left[\frac{d}{d\beta}[\chi]\right] = \frac{\partial}{\partial t_0}[Y]$$

where I omit the arguments for brevity. Thus, $\frac{\partial}{\partial t_0}[\chi]$ in (4.33) is given exactly by the construction of θ in (4.31).

Third, note that since $\chi(0, t_0, x_0) = x_0$, it follows that $\frac{\partial}{\partial x_0}[\chi(0, t_0, x_0)] = I$. The partial derivative evolves with β similarly to the prior case as

$$\frac{d}{d\beta}\left[\frac{\partial}{\partial x_0}[\chi]\right] = \frac{\partial}{\partial x_0}\left[\frac{d}{d\beta}[\chi]\right] = \frac{\partial}{\partial x_0}[Y] = \frac{\partial Y}{\partial y}\underbrace{\frac{\partial y}{\partial x_0}}_{=\Theta} \,.$$

Noting that $\chi = y$, the above equation is a linear ordinary differential equation (ODE) in β for $\frac{\partial y}{\partial x_0}$, and this ODE takes the exact same form as the equation for Θ in (4.31). Thus, $\frac{\partial}{\partial x_0}[\chi]$ in (4.34) is also given by Θ in (4.31). This completes the proof. ■

A similar result is also given in [206, Thm 6.1] and [96, Eq. 23]. Note that whereas related work in [86] solved for χ explicitly, I assume that in most cases no explicit expression will exist, so the ODEs for θ and Θ will be necessary for implementation. Next, define the set

$$\mathcal{U}_w = \left\{ u \in \mathbb{R}^m \;\middle|\; \tfrac{\|u\|+\lambda w_{u,\max}}{\|u\|} u \in \mathcal{U}, \forall \lambda \in [-1,1] \right\}, \tag{4.35}$$

which represents a subset of the allowable control inputs with a margin for the disturbance.

Next, recall that h^* in Theorem 3.4 was defined as a maximization. Proving the robust extension of Theorem 3.4 requires arguments using the maximizer times. Thus, given a u^*, define the set-valued function $\mathcal{B} : \mathcal{T} \times \mathcal{X} \to \mathbb{R}$ as

$$\mathcal{B}(t,x) \triangleq \left\{ \arg\max_{\beta \geq 0} \kappa(t+\beta, \chi(\beta,t,x)) \right\}. \tag{4.36}$$

Note that $\mathcal{B}$ could have finitely many elements or could be dense. If no maximizer exists along the trajectory χ starting from (t,x) (e.g. $\kappa(t+\beta,\chi(\beta,t,x))$ is unbounded or asymptotically approaches its supremum), then the set $\mathcal{B}(t,x)$ is empty. I now present some assumptions and notations, followed by the robust extension of Remark 3.1.

Assumption 4.1. *For the remainder of this subsection, assume that χ is continuously differentiable, $w_x \equiv 0$, $\kappa \in \mathcal{G}^r$ for $r \geq 2$, $\mathcal{U}_w$ is nonempty, and $u^* : \mathcal{T} \times \mathcal{X} \to \mathcal{U}_w$ has co-domain $\mathcal{U}_w \subset \mathcal{U}$. Assume also that for all $(t,x) \in \mathcal{S}^{\mathcal{T}}$ the set $\mathcal{B}(t,x)$ in (4.36) is nonempty and contains at most one nonzero element. Denote the nonzero element of $\mathcal{B}(t,x)$, if it exists, as $\beta^*(t,x)$.*

Let $\frac{\partial \kappa(\cdot)}{\partial \lambda_t}, \frac{\partial \kappa(\cdot)}{\partial \lambda_x}$ refer to the partial derivative of κ with respect to its first and second arguments, respectively, evaluated at $(\cdot)$. This is to remove ambiguity when $(\cdot)$ includes functions of t and x. Similarly, define $\frac{\partial \chi(\cdot)}{\partial \lambda_\beta}, \frac{\partial \chi(\cdot)}{\partial \lambda_{t_0}}, \frac{\partial \chi(\cdot)}{\partial \lambda_{x_0}}$ and $\frac{\partial h(\cdot)}{\partial \lambda_t}, \frac{\partial h(\cdot)}{\partial \lambda_x}$. Denote the total derivative of any of these quantities as $\frac{d\varphi}{dt}$ with κ, h, or χ in place of φ. For compactness, I also abbreviate certain function arguments as a_1, a_2, a_3, defined by underbraces in the following equations. Finally, note that w_x is omitted from the arguments of $\dot{\kappa}$, $\dot{h}$, and F from (4.1) in the following equations, because I require $w_x \equiv 0$.

Lemma 4.13. *Let Assumption 4.1 hold. Consider the function $h : \mathcal{T} \times \mathcal{X} \to \mathbb{R}$ defined as*

$$h(t,x) = \max_{\beta \geq 0} \kappa(t+\beta, \chi(\beta,t,x)). \tag{4.37}$$

Suppose the control law $u : \mathcal{T} \times \mathcal{X} \to \mathcal{U}_w$ is Lebesgue-integrable, and let $x(\mathcal{T})$ be a trajectory of (4.1) under $u(t,x)$. Then $h(t,x(t))$ is absolutely continuous in t and satisfies

$$\dot{h}(t,x,u,w_u) = \max_{\beta_c \in \mathcal{B}(t,x)} \left[\frac{\partial \kappa(a_1)}{\partial \lambda_t} + \frac{\partial \kappa(a_1)}{\partial \lambda_x} \theta(\beta_c,t,x) + \frac{\partial \kappa(a_1)}{\partial \lambda_x} \Theta(\beta_c,t,x) F(t,x,u,w_u) \right] \tag{4.38}$$

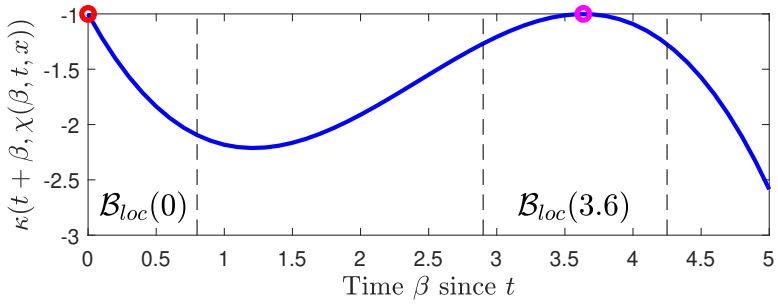

Figure 4.3: Explanation of Backup Trajectory Critical Points. An example trajectory of $\kappa(t + \beta, \chi(\beta, t, x))$ with two maximizers, $\{0, 3.6\}$. Each maximizer β_c is surrounded by a set $\mathcal{B}_{loc}(\beta_c)$ on which β_c is the unique maximizer. At time $t + \Delta t$ for small Δt, the trajectory may only have a single global maximizer, but both sets $\mathcal{B}_{loc}$ will still each contain a local maximizer, which is used to compute $\frac{d\beta_c}{dt}$ for each β_c.

almost everywhere along $x(\mathcal{T})$*, where* θ, Θ *are as in* (4.31) *and* a_1 *is as in* (4.39).

Proof. First, since $\dot{x}$ satisfies (4.1), any trajectory $x(\mathcal{T})$ must be absolutely continuous [212, Eq. C.2]. Since $\mathcal{B}(t, x)$ is assumed nonempty, h in (4.37) always exists, and on any compact interval $[t_1, t_2]$ the elements β_c of $\mathcal{B}(t, x(t))$ are bounded. Note that h in (4.37) is equivalent to

$$h(t, x) \equiv \kappa(\underbrace{t + \beta_c, \chi(\beta_c, t, x)}_{=a_1}) \tag{4.39}$$

for any $\beta_c \in \mathcal{B}(t, x)$. Since κ and χ are both continuously differentiable in all arguments, β_c is locally bounded, and $x(t)$ is absolutely continuous, it follows that $h(t, x(t))$ in (4.39), and therefore (4.37) also, is absolutely continuous in t.

Next, to prove (4.38), I first consider how each maximizer $\beta_c \in \mathcal{B}(t, x)$ varies with (t, x), and then will study how the value of h in (4.39) varies with β_c. Since $\mathcal{B}$ is assumed to contain at most one nonzero element, $\mathcal{B}$ cannot be dense. Thus, there exist open subsets of $\mathbb{R}_{\geq 0}$, denoted $\mathcal{B}_{loc}(\beta_c)$, containing only one element β_c of $\mathcal{B}(t, x)$. These sets are visualized in Fig. 4.3. Within their respective sets $\mathcal{B}_{loc}(\beta_c)$, each $\beta_c \in \mathcal{B}(t, x)$ is a strict maximizer. Thus, given a $\beta_c \in \mathcal{B}$, the derivative of $\beta_c(t, x)$ with respect to some variable is equivalent to the derivative of the maximizer of (4.37) restricted to $\beta \in \mathcal{B}_{loc}(\beta_c)$ with respect to that variable. I analyze these derivatives in three cases. First, if $\beta_c = 0 \in \mathcal{B}(t, x)$, then β_c is a strict maximizer on $\mathcal{B}_{loc}(0) = [0, c)$ for some c, and it must hold that $\dot{\kappa}(t + \beta_c, \chi(\beta_c, t, x))\big|_{\beta_c=0} \leq 0$. Note also the equivalency $\dot{\kappa}(t + \beta_c, \chi(\beta_c, t, x))\big|_{\beta_c=0} \equiv \frac{d\kappa(a_1)}{d\beta_c}\big|_{\beta_c=0} \equiv \dot{\kappa}(t, x)$, and note that this quantity is independent of u, as $\kappa \in \mathcal{G}^r$ for $r \geq 2$ (note how unlike in Theorem 3.4, I do not allow $\kappa \in \mathcal{G}^1$ here). For the first case, further suppose that $\dot{\kappa}(t, x) < 0$ strictly. Then at an infinitesimal time in the future $t + \Delta t$ (or the past if $\Delta t < 0$), the point $\beta_c = 0$ will still be a strict maximizer on $\mathcal{B}_{loc}(0)$, so it must

hold that $\frac{d\beta_c}{dt}\big|_{\beta_c=0} = \lim_{\Delta t\to 0}\frac{\beta_c(t+\Delta t,x(t+\Delta t))-\beta_c(t,x(t))}{\Delta t} = \lim_{\Delta t\to 0}\frac{0-0}{\Delta t} = 0$. Second, if $\beta_c = 0$ and $\dot\kappa(t,x) = 0$, then it holds equivalently that $\frac{d\kappa(a_1)}{d\beta_c}\big|_{\beta_c=0} \equiv \dot\kappa(t,x) = 0$. Third, if $\beta_c \neq 0, \beta_c \in \mathcal{B}(t,x)$, then $\beta_c = \beta^*(t,x)$ is a maximizer of the continuously differentiable function κ on an open interval $\mathcal{B}_{loc}(\beta_c) = (c_1, c_2)$ for some $0 < c_1 < c_2$, so it must hold that the derivative of κ at the maximizer is zero, i.e. $\frac{d\kappa(a_1)}{d\beta_c}\big|_{\beta_c=\beta^*} = 0$. It follows that $\frac{d\kappa(a_1)}{d\beta_c}\frac{d\beta_c}{dt} = 0$ in all three cases and regardless of u and β_c, which is the result I will need in (4.40) below. Next, note that $\dot h$ satisfies

$$
\begin{aligned}
\dot h(t,x,u,w_u) &= \frac{dh(t,x)}{dt} = \frac{d\kappa(a_1)}{dt} \\
&= \frac{\partial\kappa(a_1)}{\partial\lambda_t}\frac{d(t+\beta_c)}{dt} + \frac{\partial\kappa(a_1)}{\partial\lambda_x}\frac{d\chi(\beta_c,t,x)}{dt} \\
&= \frac{\partial\kappa(a_1)}{\partial\lambda_t}\left(1+\frac{d\beta_c}{dt}\right) + \frac{\partial\kappa(a_1)}{\partial\lambda_x}\left(\frac{\partial\chi(\beta_c,t,x)}{\partial\lambda_\beta}\frac{d\beta_c}{dt} + \frac{\partial\chi(\beta_c,t,x)}{\partial\lambda_{t_0}} + \frac{\partial\chi(\beta_c,t,x)}{\partial\lambda_{x_0}}\frac{dx}{dt}\right) \\
&\overset{(4.1)}{=} \left(\frac{\partial\kappa(a_1)}{\partial\lambda_t} + \frac{\partial\kappa(a_1)}{\partial\lambda_x}\frac{\partial\chi(\beta_c,t,x)}{\partial\lambda_\beta}\right)\frac{d\beta_c}{dt} + \frac{\partial\kappa(a_1)}{\partial\lambda_t} \\
&\qquad + \frac{\partial\kappa(a_1)}{\partial\lambda_x}\frac{\partial\chi(\beta_c,t,x)}{\partial\lambda_{t_0}} + \frac{\partial\kappa(a_1)}{\partial\lambda_x}\frac{\partial\chi(\beta_c,t,x)}{\partial\lambda_{x_0}}F(t,x,u,w_u) \\
&= \underbrace{\frac{d\kappa(a_1)}{d\beta_c}\frac{d\beta_c}{dt}}_{=0} + \frac{\partial\kappa(a_1)}{\partial\lambda_t} + \frac{\partial\kappa(a_1)}{\partial\lambda_x}\underbrace{\frac{\partial\chi(\beta_c,t,x)}{\partial\lambda_{t_0}}}_{=\theta(\beta_c,t,x)} + \frac{\partial\kappa(a_1)}{\partial\lambda_x}\underbrace{\frac{\partial\chi(\beta_c,t,x)}{\partial\lambda_{x_0}}}_{=\Theta(\beta_c,t,x)}F(t,x,u,w_u) \qquad (4.40) \\
&= \frac{\partial\kappa(t,x)}{\partial t}\bigg|_{\substack{t=t+\beta_c\\ x=\chi(\beta_c,t,x)}} + \frac{\partial\kappa(t,x)}{\partial x}\bigg|_{\substack{t=t+\beta_c\\ x=\chi(\beta_c,t,x)}}\Big(\theta(\beta_c,t,x) + \Theta(\beta_c,t,x)F(t,x,u,w_u)\Big)
\end{aligned}
$$

for one of the $\beta_c \in \mathcal{B}(t,x)$. Specifically, if $\mathcal{B}(t,x(t))$ contains N elements at time t, then there are N sets $\mathcal{B}_{loc}$. At an infinitesimal time in the future $t+\Delta t$, each set $\mathcal{B}_{loc}$ will still contain a local maximizer, and at least one of these will still be a global maximizer (for sufficiently small Δt). The derivative $\dot h$ corresponds to the rate of change of whichever local maximizer(s) is the global maximizer at both t and $t+\Delta t$, i.e. the $\beta_c \in \mathcal{B}(t,x)$ that maximizes (4.40) (similar to in Remark 3.1). This is then encoded in (4.38). ■

Now that I have an expression for the total derivative (4.38) of h in (4.37), the next lemma derives an upper bound on this derivative that I then use in Theorem 4.15 to prove that the corresponding set $\mathcal{H}$ can be rendered forward invariant while satisfying the input constraints.

Lemma 4.14. *Suppose the assumptions of Lemma 4.13 hold. Then h in* (4.37) *satisfies*

$$
\dot h(t,x,u,w_u) \leq \begin{cases} 0 & \mathcal{B}(t,x(t)) = \{0\} \\ \max\{0,\, q(\beta^*(t,x),t,x)[u+w_u-u^*(t,x)]\} & \mathcal{B}(t,x(t)) \neq \{0\} \end{cases} \qquad (4.41)
$$

almost everywhere along $x(\mathcal{T})$, where

$$q(\beta,t,x) \triangleq \left.\frac{\partial\kappa(t,x)}{\partial x}\right|_{\substack{t=t+\beta_c\\ x=\chi(\beta_c,t,x)}} \Theta(\beta,t,x)g(t,x) \tag{4.42}$$
$$\equiv \nabla\kappa(t+\beta,\chi(\beta,t,x))\Theta(\beta,t,x)g(t,x)\,.$$

Proof. Let β_c be any element of $\mathcal{B}(t,x)$. Since β_c is a maximizer of (4.37), it must hold [213, Eq. 11.35] that

$$0 \geq \frac{d}{d\beta_c}[\kappa(\underbrace{t+\beta_c,\chi(\beta_c,t,x)}_{=a_1})]\frac{\partial\kappa(a_1)}{\partial\lambda_t} + \frac{\partial\kappa(a_1)}{\partial\lambda_x}\frac{\partial\chi(\beta_c,t,x)}{\partial\lambda_\beta}\,. \tag{4.43}$$

Next, define the quantity ω as follows,

$$\omega \triangleq \kappa(t+\beta_c,\chi(\beta_c,t,x)) \tag{4.44}$$

and note that ω has the equivalent form

$$\omega \equiv \underbrace{\kappa(t+\beta_c,\chi(\underbrace{\beta_c-\tau,t+\tau,\chi(\tau,t,x)}_{=a_2}))}_{=a_3} \tag{4.45}$$

for any $\tau\in[0,\beta_c]$. Since ω is constant with respect to τ, it follows that $\frac{d}{d\tau}(\omega)=0$, so

$$0 = \frac{d\omega}{d\tau} = \frac{\partial\kappa(a_3)}{\partial\lambda_x}\left(-\frac{\partial\chi(a_2)}{\partial\lambda_\beta}+\frac{\partial\chi(a_2)}{\partial\lambda_{t_0}}+\frac{\partial\chi(a_2)}{\partial\lambda_{x_0}}\frac{\partial\chi(\tau,t,x)}{\partial\lambda_\beta}\right). \tag{4.46}$$

At $\tau=0$, this becomes

$$\begin{aligned} 0 &= \frac{\partial\kappa(a_1)}{\partial\lambda_x}\left(-\frac{\partial\chi(\beta_c,t,x)}{\partial\lambda_\beta}+\frac{\partial\chi(\beta_c,t,x)}{\partial\lambda_{t_0}}+\frac{\partial\chi(\beta_c,t,x)}{\partial\lambda_{x_0}}Y(0,t,x)\right)\\ &\overset{(4.43)}{\geq} \frac{\partial\kappa(a_1)}{\partial\lambda_t}+\frac{\partial\kappa(a_1)}{\partial\lambda_x}\left(\frac{\partial\chi(\beta_c,t,x)}{\partial\lambda_{t_0}}+\frac{\partial\chi(\beta_c,t,x)}{\partial\lambda_{x_0}}Y(0,t,x_0)\right). \end{aligned} \tag{4.47}$$

$\dot{h}$ in (4.40) then simplifies to

$$\begin{aligned} \dot{h}(t,x,u,w_x) &\overset{(4.47)}{\leq} \frac{\partial\kappa(a_1)}{\partial\lambda_x}\frac{\partial\chi(\beta_c,t,x)}{\partial\lambda_{x_0}}\left(F(t,x,u,w_u)-Y(0,t,x)\right)\\ &\overset{(4.1),(4.31)}{=} \frac{\partial\kappa(a_1)}{\partial\lambda_x}\frac{\partial\chi(\beta_c,t,x)}{\partial\lambda_{x_0}}g(t,x)\left(u+w_u-u^*(t,x)\right)\\ &\overset{(4.42)}{=} q(\beta_c,t,x)(u+w_u-u^*(t,x))\,. \end{aligned} \tag{4.48}$$

As in Lemma 4.13, $\dot{h}$ is upper bounded by the maximum of (4.48) over all $\beta_c \in \mathcal{B}(t,x)$. Next, recall that $\Theta(0,t,x) = I$ in (4.31), so $q(0,t,x) = \nabla\kappa(t,x)g(t,x)$ in (4.42). By definition of κ being of relative-degree $r \geq 2$, it follows that $q(0,t,x) = 0$ for any (t,x). Therefore, (4.48) is at most 0 when $\beta_c = 0$. It follows that $\dot{h}$ in (4.38) is at most 0 when $\beta_c = 0$ is the only element of $\mathcal{B}$, and $\dot{h}$ is upper bounded by the second case of (4.41) when there is a nonzero $\beta_c = \beta^* \in \mathcal{B}$. ■

That is, due to the $\max$ function, h in (4.37) is potentially non-smooth, so the partial derivatives $\partial_t h$ and ∇h are not well defined. Instead, Lemma 4.13 and Lemma 4.14 provide expressions for the total derivative $\dot{h}$. However, (4.41) still contains the unknown disturbance w_u, so based on (4.41) and with a slight abuse of notation, I now re-define

$$W(t,x) = \|q(\beta^*(t,x),t,x)\| w_{u,\max}\,, \tag{4.49}$$

$$\begin{aligned}\mathcal{U}_{\text{rcbf}}(t,x) = \{u \in \mathcal{U} \mid \text{if } \mathcal{B}(t,x) \neq \{0\} \text{ then} \\ q(\beta^*(t,x),t,x)(u - u^*(t,x)) \leq \alpha(-h(t,x)) - W(t,x)\}\,.\end{aligned} \tag{4.50}$$

If $\mathcal{B}(t,x) = \{0\}$, then $\mathcal{U}_{\text{rcbf}}(t,x) = \mathcal{U}$. I now present the robust extension of Theorem 3.4.

Theorem 4.15. *Let Assumption 4.1 hold. For h as in* (4.37)*, define $\mathcal{U}_{\text{rcbf}}$ as in* (4.50) *and define $\mathcal{H}$ as in* (4.3)*, where $\mathcal{H} \equiv \mathcal{H}_{\text{res}}$. Then for any $\alpha \in \mathcal{K}_e$, the set $\mathcal{U}_{\text{rcbf}}(t,x)$ is nonempty for all $(t,x) \in \mathcal{H}^{\mathcal{T}}$, and any control law $u : \mathcal{T} \times \mathcal{X}$ satisfying the conditions of Theorem 4.1, Theorem 4.3, or Theorem 4.4 will render $\mathcal{H}^{\mathcal{T}}$ forward invariant.*

Proof. If $\mathcal{B}(t,x) = \{0\}$, then $\mathcal{U}_{\text{rcbf}}(t,x) = \mathcal{U}$, which is always nonempty, and $\dot{h}$ in (4.41) is nonpositive regardless of the control input u. If instead $\mathcal{B}(t,x) \neq \{0\}$, then the disturbance w_u that maximizes (4.41) has magnitude $w_{u,\max}$ and is in the direction of $q(\beta^*(t,x),t,x)$. Denote this disturbance as $w_{u,\text{worst}} = \frac{q(\beta^*(t,x),t,x)}{\|q(\beta^*(t,x),t,x)\|} w_{u,\max}$, and note that this is a known quantity because I assumed at most one nonzero $\beta_c \in \mathcal{B}(t,x)$ (note that if I instead allowed multiple nonzero β_c, then I could not uniquely define $w_{u,\text{worst}}$ here). By Assumption 4.1, the control input $u = u^*(t,x) - w_{u,\text{worst}}$ exists in $\mathcal{U}$ since u^* is restricted to $\mathcal{U}_w$ in (4.35). This choice of u also renders $\dot{h}(t,x,u,w_u) \leq 0$ in (4.41) for any w_u, and therefore belongs to $\mathcal{U}_{\text{rcbf}}(t,x)$ in (4.50) for any $\alpha \in \mathcal{K}_e$, so $\mathcal{U}_{\text{rcbf}}$ is always nonempty regardless of the choice of $\alpha \in \mathcal{K}_e$.

With W as in (4.49), any $u(t,x) \in \mathcal{U}_{\text{rcbf}}(t,x)$ in (4.50) causes $\dot{h}$ in (4.41) to satisfy $\dot{h}(t,x,u,w_u) \leq \alpha(-h(t,x))$ regardless of the disturbance w_u. Thus, if the additional regularity assumptions of any of Theorem 4.1, Theorem 4.3, or Theorem 4.4 hold, then u will render $\mathcal{H}^{\mathcal{T}}$ forward invariant. ■

Corollary 4.16. *Suppose the assumptions of Theorem 4.15 hold. If additionally $\mathcal{B}(t,x)$ has exactly one element for all $(t,x) \in \mathcal{H}^{\mathcal{T}}$, then h in* (4.37) *is an RCBF on $\mathcal{H}$ in* (4.3) *for the system* (4.1) *with $w_x \equiv 0$ for any $\alpha \in \mathcal{K}_e$, and* (4.49),(4.50) *are equivalent to* (4.7),(4.8)*, where*

$$\partial_t h(t,x) = \left.\frac{\partial \kappa(t,x)}{\partial t}\right|_{\substack{t=t+\beta_c \\ x=\chi(\beta_c,t,x)}} + \left.\frac{\partial \kappa(t,x)}{\partial x}\right|_{\substack{t=t+\beta_c \\ x=\chi(\beta_c,t,x)}} \theta(\beta_c,t,x)\,, \tag{4.51}$$

$$\nabla h(t,x) = \left.\frac{\partial \kappa(t,x)}{\partial x}\right|_{\substack{t=t+\beta_c \\ x=\chi(\beta_c,t,x)}} \Theta(\beta_c,t,x)\,. \tag{4.52}$$

with $\beta_c = \mathcal{B}(t,x)$.

Proof. Under this stricter condition on $\mathcal{B}(t,x)$, the function h in (4.37) is now continuously differentiable [86, Lemma 1]. By the same argument as in Theorem 4.15, the control input $u = u^*(t,x) - w_{u,\text{worst}}$ must exist in $\mathcal{U}$, and makes $\dot{h}$ at most 0 independent of the actual disturbance w_u. Thus, condition (4.6) is satisfied for any $\alpha \in \mathcal{K}_e$, so h is an RCBF on $\mathcal{H}$ for any $\alpha \in \mathcal{K}_e$.

Next, since h is continuously differentiable, I can now properly define its partial derivatives. I proved in Theorem 4.15 that $\frac{d\kappa(a_1)}{d\beta_c}\frac{d\beta_c}{dt} = 0$, where now $\beta_c(t,x) \equiv \mathcal{B}(t,x)$. Expanding this, it follows that

$$0 = \left(\frac{\partial \kappa(a_1)}{\partial \lambda_t} + \frac{\partial \kappa(a_1)}{\partial \lambda_x}\frac{\partial \chi(\beta_c,t,x)}{\partial \lambda_\beta}\right)\left(\frac{\partial \beta_c(t,x)}{\partial t} + \frac{\partial \beta_c(t,x)}{\partial x}\frac{dx}{dt}\right) \tag{4.53}$$

Note that (4.53) must hold for any $\frac{dx}{dt}$, so it follows that $\left(\frac{\partial \kappa(a_1)}{\partial \lambda_t} + \frac{\partial \kappa(a_1)}{\partial \lambda_x}\frac{\partial \chi(\beta_c,t,x)}{\partial \lambda_\beta}\right)\frac{\partial \beta_c(t,x)}{\partial x} \equiv 0$. It follows that $\left(\frac{\partial \kappa(a_1)}{\partial \lambda_t} + \frac{\partial \kappa(a_1)}{\partial \lambda_x}\frac{\partial \chi(\beta_c,t,x)}{\partial \lambda_\beta}\right)\frac{\partial \beta_c(t,x)}{\partial t} \equiv 0$ too. That is, I have extended the result on the sensitivity of κ to the total derivative of β_c from Theorem 4.15 to the sensitivity of κ to the partial derivatives of β_c as well. This allows the separation of (4.40) into the desired partial derivatives.

Next, the partial derivatives of (4.37) are equivalent to the partial derivatives of (4.39), which are

$$\partial_t h(t,x) = \frac{\partial \kappa(a_1)}{\partial \lambda_t}\left(1 + \frac{\partial \beta_c(t,x)}{\partial t}\right) + \frac{\partial \kappa(a_1)}{\partial \lambda_x}\left(\frac{\partial \chi(\beta_c,t,x)}{\partial \lambda_{t_0}} + \frac{\partial \chi(\beta_c,t,x)}{\partial \lambda_\beta}\frac{\partial \beta_c(t,x)}{\partial t}\right) \tag{4.54}$$

$$\nabla h(t,x) = \frac{\partial \kappa(a_1)}{\partial \lambda_t}\frac{\partial \beta_c(t,x)}{\partial x} + \frac{\partial \kappa(a_1)}{\partial \lambda_x}\left(\frac{\partial \chi(\beta_c,t,x)}{\partial \lambda_{x_0}} + \frac{\partial \chi(\beta_c,t,x)}{\partial \lambda_\beta}\frac{\partial \beta_c(t,x)}{\partial x}\right)\,. \tag{4.55}$$

Cancelling the zero terms identified above, (4.54)-(4.55) simplify to

$$\partial_t h(t,x) = \frac{\partial \kappa(a_1)}{\partial \lambda_t} + \frac{\partial \kappa(a_1)}{\partial \lambda_x}\frac{\partial \chi(\beta_c,t,x)}{\partial \lambda_{t_0}}, \tag{4.56}$$

$$\nabla h(t,x) = \frac{\partial \kappa(a_1)}{\partial \lambda_x}\frac{\partial \chi(\beta_c,t,x)}{\partial \lambda_{x_0}}, \tag{4.57}$$

which are equivalent to (4.51)-(4.52). ■

Remark 4.4. *The extra condition that $\mathcal{B}(t,x)$ have exactly one element in Corollary 4.16 is a technicality, but is required to ensure that h in* (4.37) *is continuously differentiable, which was a condition of Definition 4.2. If I had instead defined RCBFs using directional derivatives in this chapter, then I would not need to make this distinction.*

Remark 4.5. *I am not aware of a method by which Theorem 4.15 and Corollary 4.16 can be extended to the case of $w_x \not\equiv 0$ with guaranteed safety. In this case, I refer readers to the robustness method in [116, Prop. 5]. This could require expansion of the control set $\mathcal{U}$, and/or could be an interesting area for future work.*

Remark 4.6. *In practice, computing* (4.37) *and* (4.38) *requires computing the maximum and maximizer of the solution to an ODE. Most differential equation simulators output only samples along a curve rather than a continuous curve. One should not simply search for the maximum of these samples, because that might be less than the maximum of the continuous curve. In practice, I found it effective to fit a quadratic curve to the maximum three samples, and then use the maximum and maximizer of this fitted curve as the values of $h(t+\beta_c, \chi(\beta_c,t,x))$ and β_c, respectively. See also the discussion of how this was performed for the case study in Section 3.4.1.*

In other words, if the undisturbed trajectory described by $\chi(\beta,t,x(t))$ remains safe for all future time, then there always exists a safe trajectory from $(t,x(t))$ regardless of the actual disturbance (for matched disturbances only). It is trivial to show that h in (4.37) satisfies $h(t,x) \geq \kappa(t,x), \forall (t,x) \in \mathcal{T} \times \mathcal{X}$, so the CBF set $\mathcal{H}$ in Theorem 4.15 is a subset of the safe set $\mathcal{S}$. In Chapter 3, Theorem 3.5 was presented as a special case of theorem 3.4. In this chapter, the assumptions of Theorem 4.10 and Theorem 4.11 are different from those of Theorem 4.15 (i.e. these theorems are no longer just a special case), but Theorem 4.15 is still the most generally applicable method of finding a CBF set $\mathcal{H}$ because there may exist u^* satisfying Assumption 4.1 even when there is no $a_{\max}$ or Φ satisfying the assumptions of Theorems 4.10 or 4.11, respectively.

Note that satisfying the assumptions on $\mathcal{B}(t,x)$ in Assumption 4.1 could be easy or challenging depending on the system and choice of u^*, as will be elaborated upon in application in Section 4.5. The need to verify these assumptions is perhaps the primary limitation of Theorem 4.15, and is a topic of future study. In the case of relative-degree $r=2$ or $r=3$, a sufficient condition for the

assumption on $\mathcal{B}$ to hold is the existence of $\gamma \in \mathbb{R}_{<0}$ such that $\kappa^{(r)}(t, x, u^*(t, x), w_u) \leq \gamma, \forall (t, x) \in \mathcal{S}^{\mathcal{T}}, \forall w_u \in \mathcal{W}$. I also suspect that Theorem 4.15 will often hold even when $\mathcal{B}$ has several elements, likely with additional conditions, but how to show this generally is an open area for future research.

Applying Theorem 4.15 also requires that one knows a control law u^* that generally drives the trajectories χ towards safe states. For a system with relative-degree r, one obvious choice of control law is

$$u^*_{\text{opt}}(t, x) = \underset{u \in \mathcal{U}_w}{\arg\min} \nabla \kappa^{(r-1)}(t, x) g(t, x) u \,. \tag{4.58}$$

Certain known phenomena of system trajectories might also motivate other control laws, such as the control law u^*_{orth} in Section 4.5, inspired by satellite orbits.

Of the methods presented so far, Theorem 4.15 is the most computationally intensive. Unlike the previous methods, the ODE in (4.31) rarely simplifies to explicit algebraic expressions. That said, the simulations in Section 4.5 show that the computation costs are reasonable on a 6-dimensional system. In practice, one also needs an upper bound on the amount of time to propagate (4.31) by which time all maximizers of (4.37) will be guaranteed to have occurred, though computation of such a bound is system-specific and not addressed in this dissertation.

4.4 Hysteresis-Switched Control Barrier Functions

This section develops a condition on the control input that establishes set forward invariance similar to Theorem 4.1, but which places fewer constraints on the system trajectories. The core idea of this section is that, as I will show, the RCBF condition $u \in \mathcal{U}_{\text{rcbf}}(t, x)$ in (4.8) does not need to be enforced everywhere in $\mathcal{H}_{\text{res}}$ to ensure forward invariance of $\mathcal{H}_{\text{res}}$, and this section will develop a systematic way to relax the conditions in Theorem 4.1, Theorem 4.3, and Theorem 4.4.

Recall that the theorems in Section 4.3 showed that the presented functions h are RCBFs for any $\alpha \in \mathcal{K}_e$. Thus, α is a free parameter, though α still impacts system performance by its role in defining $\mathcal{U}_{\text{rcbf}}$ in (4.8),(4.50). Near the boundary of the set $\mathcal{H}_{\text{res}}$, the function α works to ensure invariance of $\mathcal{H}_{\text{res}}$ by bounding the rate at which the system approaches this boundary. However, in the interior of the $\mathcal{H}_{\text{res}}$, the bound provided by α is arbitrary (since α is a free parameter) and can prevent the system from following otherwise safe trajectories. For example, consider the CBF in (4.17). The CBF condition for this function becomes

$$\ddot{\kappa}(x, u) \leq -\frac{\dot{\kappa}(x)}{|\dot{\kappa}(x)|} a_{\max} + \frac{\alpha(-h(x))}{|\dot{\kappa}(x)|} a_{\max} \tag{4.59}$$

Suppose $\kappa, \dot{\kappa}, \ddot{\kappa}$ represent position, velocity, and acceleration, respectively. If an agent is far away from the boundary of $\mathcal{H}_{\text{res}}$ (i.e. $h(x) \ll 0$ and $\kappa(x) \ll 0$) and moving quickly (i.e. $|\dot{\kappa}| \gg 0$),

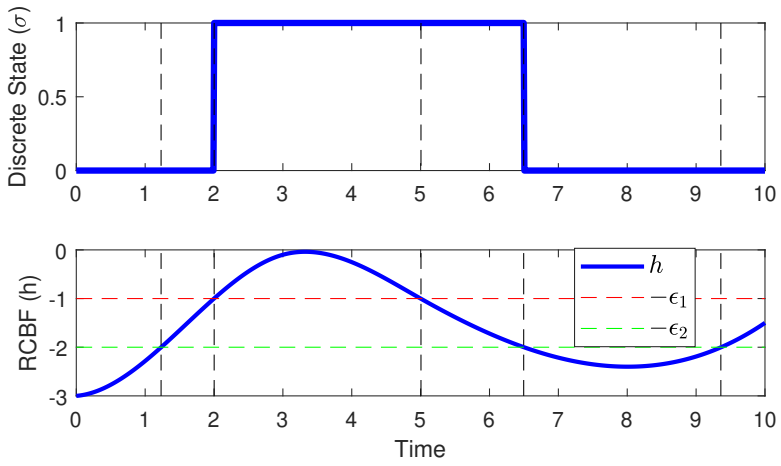

Figure 4.4: Hysteresis Switching Control Law Example Trajectory. A visualization of state-dependent hysteresis-switching in (4.60).

then the acceleration $\ddot{\kappa}$ may be constrained even though the state is far from the boundary of $\mathcal{H}_{\text{res}}$. This is a consequence of how all CBFs for high relative-degree constraint functions are necessarily functions of the constraint function derivatives. In Section 4.5, I will apply the RCBF condition in a *safety filter* similar to (2.9), where I want this safety filter to only minimally modify the nominal control law u_{nom}. Thus, this section seeks to remove the above unnecessary constraint when h is far less than zero.

To this end, I introduce a discrete state $\sigma \in \{0, 1\}$, which captures whether the state is near the boundary of $\mathcal{H}$. I say a particular RCBF condition is *active* if $\sigma = 1$, and *inactive* if $\sigma = 0$. If there are multiple RCBFs, then one would construct $\sigma_i, i = 1, 2, \cdots$ for each RCBF. Define $\epsilon_2 > \epsilon_1 \geq 0$ and at a time instant t, let

$$\sigma(t) = \begin{cases} 0 & h(t, x(t)) \leq -\epsilon_2 \\ 1 & h(t, x(t)) \geq -\epsilon_1 \\ \sigma(t^-) & \text{otherwise} \end{cases}. \tag{4.60}$$

where $\sigma(t_0) = 0$, and t^- denotes the time instant immediately preceding t. That is, σ exhibits hysteresis, as visualized in the example in Fig. 4.4; the constraint becomes active ($\sigma \to 1$) when h first exceeds some tolerance ($h \geq -\epsilon_1$), and remains active until the state is far away from the boundary of $\mathcal{H}$ ($h \leq -\epsilon_2$). One can then use this discrete state to choose between two control laws depending on whether the constraint is active or inactive. The result is a switched control law with state-dependent switching, where a *switch* occurs when $\sigma(t)$ changes between discrete states. I denote a switch to active as $\sigma \to 1$ and a switch to inactive as $\sigma \to 0$. Since trajectories of x are continuous and h is continuous, the state can never leave $\mathcal{H}_{\text{res}} \subseteq \mathcal{S}$ without the corresponding RCBF h becoming active. I now demonstrate set forward invariance under such a switched control

law in the following theorem.

Theorem 4.17. *Let $h : \mathcal{T} \times \mathcal{X} \to \mathbb{R}$ be an RCBF on $\mathcal{H}$ in* (4.3) *and let $u : \mathcal{T} \times \mathcal{X} \times \{0, 1\} \to \mathcal{U}$, denoted $u(t, x, \sigma)$, be a control law, with σ as in* (4.60)*. Then the result of Theorem 4.1 applies for any control law satisfying $u(t, x, 1) \in \mathcal{U}_{\text{rcbf}}(t, x), \forall (t, x) \in \{(t, x) \in \mathcal{D}^{\mathcal{T}} \mid h(t, x) \geq -\epsilon_2\}$, and the result of Theorem 4.3 and Theorem 4.4 applies for any control law satisfying $u(t, x, 1) \in \mathcal{U}_{\text{rcbf}}(t, x), \forall (t, x) \in \{(t, x) \in \mathcal{H}^{\mathcal{T}} \mid h(t, x) \geq -\epsilon_2\}$.*

Proof. By construction, solutions $x(\mathcal{T})$ to dynamics (4.1) are absolutely continuous [212, Eq. C.2], and are assumed to be unique. Therefore, $\theta(t) = h(t, x(t))$ is absolutely continuous, and regardless of the disturbances w_u, w_x, $\dot{\theta}$ satisfies $\dot{\theta}(t) = \dot{h}(t, x(t), u(t, x(t), 1), w_u, w_x) \leq \alpha(-\theta(t))$ for all $t \in \mathcal{T}_1 \triangleq \{t \in \mathcal{T} \mid \sigma(t) = 1\}$. Thus, by Lemma 2.3, if $h(t_0, x(t_0)) \leq 0$, then $h(t, x(t)) = \theta(t) \leq 0, \forall t \in \mathcal{T}_1$. Also, by construction, $h(t, x(t)) \leq -\epsilon_1 \leq 0, \forall t \in \mathcal{T} \setminus \mathcal{T}_1$. Thus, under the assumptions of Theorem 4.1, it follows that $h(t, x(t)) \leq 0, \forall t \in \mathcal{T}$, which implies $\mathcal{H}^{\mathcal{T}}$ is forward invariant. Recall that Theorem 4.3 is a special case of Theorem 4.1, so the result holds in that case as well. Finally, note that $\partial\mathcal{H}^{\mathcal{T}} \subset \{(t, x) \in \mathcal{H}^{\mathcal{T}} \mid h(t, x) \geq -\epsilon_2\}$, so the argument in Theorem 4.4 still holds as well. Thus, $\mathcal{H}^{\mathcal{T}}$ is rendered forward invariant by such a switched control law under any of the three sets of assumptions in Theorem 4.1, Theorem 4.3, or Theorem 4.4. ■

Note that one can come to a similar conclusion as in Theorem 4.17 from [151, Thm. 3], but by using a hysteresis switching logic, Theorem 4.17 avoids the need for differential inclusions and non-smooth analysis. Such switching prevents sliding modes with chattering control inputs that may be introduced by differential inclusions, and thus results in more realistic actuator behavior, while still providing the benefits of allowing potentially discontinuous controllers. A related approach for smooth switching based on the value of κ rather than h is presented in [89], but this approach requires $\mathcal{U} = \mathbb{R}^m$. Compared to [28, 67], the switching approach in Theorem 4.17 still utilizes the RCBF condition in (4.8), and thus can be included in optimization-based controllers such as (2.9), as is done in Section 4.5. Note that if $\epsilon_1 = \infty$, then Theorem 4.17 reduces to Theorem 4.1.

Remark 4.7. *Note that the argument used to prove invariance of $\mathcal{H}_{\text{res}}$ in Corollaries 4.5, 4.6, 4.7, 4.8 and Lemma 4.9 is unaffected by the addition of switching as long as* (4.13)-(4.14) *still hold for the switched control law. If $\kappa \in \mathcal{G}^r$ for $r \geq 2$, then* (4.13)-(4.14) *are independent of the control law and thus agnostic to whether the control law is continuous or switched. Thus, $\mathcal{H}$ may be replaced by $\mathcal{H}_{\text{res}}$ in Theorem 4.17 for the RCBFs* (4.22) *and* (4.27)*. Similarly, the RCBF in* (4.37) *and Theorem 4.15 may be used with a switched control law even if the stricter conditions of Corollary 4.16 do not hold.*

For practical implementation, it is often desirable to tune a controller to reduce the number of switches of the discrete state, which could cause jumps in the control input that wear out the actuators. This tuning can be done directly in the RCBF development by proper choice of the class-$\mathcal{K}_e$ function α and the switching tolerances ϵ_1, ϵ_2. To this end, I propose using the function

$$\alpha_r(\lambda, t, x) \triangleq \frac{W(t,x)\lambda}{\epsilon_1} \tag{4.61}$$

in place of α in (4.6),(4.8),(4.50) and the choices $\epsilon_2 > 2\epsilon_1$ and $\epsilon_1 > 0$. The reasoning for choosing $\epsilon_2 > 2\epsilon_1$ is that the disturbances w_u, w_x could be helpful or harmful to safety. If $\sigma = 1$ and the disturbance is temporarily helpful to safety (i.e. causing h to decrease), I seek to avoid a switch $\sigma \to 0$ at states where a disturbance harmful to safety (i.e. causing h to increase) could quickly cause another switch back to $\sigma \to 1$. To this end, suppose that the control input satisfies the RCBF condition in (4.8) or (4.50) with equality. Then the set $\mathcal{S}_f(t) \triangleq \{x \in \mathcal{H}(t) \mid \dot{h}(t,x) = 0\}$ is asymptotically stable. Next, suppose $\alpha = \alpha_r$ as in (4.61). Then $\dot{h}$ is given by

$$\begin{aligned}
\dot{h}(\cdot) &= \partial_t h(\cdot) + \nabla h(\cdot)\big(f(\cdot) + g(\cdot)(u + w_u) + w_x\big) \\
&\overset{(4.8)}{=} \alpha(-h(\cdot)) - W(\cdot) + \nabla h(\cdot)\big(g(\cdot)w_u + w_x\big) \\
&\overset{(4.61)}{=} -\frac{W(\cdot)h(\cdot)}{\epsilon_1} + \underbrace{\nabla h(\cdot)(g(\cdot)w_u + w_x)}_{=W_{\text{real}}(t,x,w_u,w_x)} - W(\cdot)
\end{aligned} \tag{4.62}$$

where W_{real} captures the effect on $\dot{h}$ of the disturbances that occur online. If the online disturbances w_u, w_x are maximally harmful to safety (i.e. tending to increase $\dot{h}$), then $W_{\text{real}} = W$ in (4.62) and $\mathcal{S}_f(t) \equiv \{x \in \mathcal{H} \mid h(t,x) = 0\}$, so h converges to 0 asymptotically. If instead w_u, w_x are maximally helpful to safety (i.e. tending to decrease $\dot{h}$), then $W_{\text{real}} = -W$ and $\mathcal{S}_f(t) \equiv \{x \in \mathcal{H}(t) \mid h(t,x) = -2\epsilon_1\}$, so h converges to $-2\epsilon_1$ asymptotically. Thus, choosing $\epsilon_2 > 2\epsilon_1$ in the switching conditions prevents a switch induced purely by the disturbance. Instead, switches will occur when the system objectives drive the trajectory closer to or further from the boundary of $\mathcal{H}$, i.e. when the control input changes between satisfying (4.8),(4.50) with and without equality. Finally, if the disturbance is zero, then $\mathcal{S}_f(t) \equiv \{x \in \mathcal{H}(t) \mid h(t,x) = -\epsilon_1\}$, so h will converge to $-\epsilon_1$. That is, using $\alpha = \alpha_r$ in (4.61) allows one to tune how closely the closed-loop trajectories approach the boundary of $\mathcal{H}$ as a function of the disturbances. Note that α_r does not belong to class-$\mathcal{K}_e$ (as required by Theorems 4.1,4.3,4.4 and Theorem 4.17), so the following corollary shows that α_r can still be used for safety.

Corollary 4.18. *Suppose $g(t,x)$ and $\nabla h(t,x)$ are bounded for all $(t,x) \in \mathcal{H}^{\mathcal{T}}$. Then Theorem 4.1, Theorem 4.3, Theorem 4.4 and Theorem 4.17 hold when $\mathcal{U}_{\text{rcbf}}(t,x)$ in* (4.8) *is defined with $\alpha = \alpha_r$ as in* (4.61)*. Similarly, if $q(\beta^*(t,x),t,x)$ is bounded for all $(t,x) \in \mathcal{H}^{\mathcal{T}}$, then Theorem 4.15 holds*

when (4.50) *is defined with* $\alpha = \alpha_r$.

Proof. Under the above assumptions, $W(t,x)$ in (4.7) and (4.49) is upper bounded. Let w_m denote this bound. Then $\alpha_r(\lambda,t,x) \leq \bar{\alpha}_r(\lambda) \triangleq w_m\lambda/\epsilon_1$, where $\bar{\alpha}_r \in \mathcal{K}_e$. Thus, a control input $u \in \mathcal{U}_{\text{rcbf}}(t,x)$ yields $\dot{\theta}(t) \leq \bar{\alpha}_r(-\theta(t))$ as in Theorem 4.17, so the result of Theorem 4.17 still holds. By the same argument, Theorem 4.1, Theorem 4.3, Theorem 4.4, and Theorem 4.15 hold as well. ■

Note that the extra assumption of boundedness of g and ∇h in Corollary 4.18 serves the same purposes as the compactness assumption reviewed in Lemma 2.1.

Thus, this section has proposed a method of regulating safety that allows for switched controllers that mitigate the potentially undesirable constraints following from high relative-degree constraint functions. Using the position/velocity/acceleration interpretation of κ, $\dot{\kappa}$, $\ddot{\kappa}$, respectively, this switching approach allows an agent to choose control inputs without considering the CBF condition when in the interior of its safe set and then to decelerate approximately along the surface $\mathcal{S}_f(t)$ as it approaches an unsafe region. This approach also allows for a provably safe means of adding and removing RCBFs over time, for instance, as an agent explores an unknown environment and identifies obstacles.

4.5 Spacecraft Simulation and Discussion

4.5.1 Spacecraft Dynamics

In this section, I apply the previously presented RCBFs and hysteresis-switching strategy to satellite dynamics and demonstrate these approaches in simulation. The satellite state is $x = \begin{bmatrix} r^{\mathrm{T}} & v^{\mathrm{T}} \end{bmatrix}^{\mathrm{T}}$, with dynamics

$$\dot{x} = \begin{bmatrix} \dot{r} \\ \dot{v} \end{bmatrix} = \underbrace{\begin{bmatrix} v \\ f_\mu(t,r) \end{bmatrix}}_{f(t,x)} + \underbrace{\begin{bmatrix} 0_{3\times 3} \\ I_{3\times 3} \end{bmatrix}}_{g(t,x)} (u + w_u) + \begin{bmatrix} w_x \\ 0 \end{bmatrix} \tag{4.63}$$

for position and velocity $r, v \in \mathbb{R}^3$ resolved in an inertial frame and gravitational force $f_\mu : \mathcal{T} \times \mathbb{R}^3 \to \mathbb{R}^3$. For ease of implementation, I assume w_x only acts on the $\dot{r}$ equation (any effects on $\dot{v}$ can be grouped with w_u). In this system, the matched disturbance w_u could represent unmodelled forces like higher-order gravity effects or solar radiation pressure, while the unmatched disturbance could represent filtered sensor updates. Suppose the satellite mass m is approximately constant and the satellite contains 6 orthogonal thrusters (or fewer thrusters capable of changing orientation sufficiently fast) capable of outputting a continuously variable thrust in $[0, u_{\max} m]$ for

some $u_{\max} > 0$. Then the control set is $\mathcal{U} = \{u \in \mathbb{R}^3 \mid \|u\|_\infty \leq u_{\max}\}$ and represents the satellite allowable acceleration. The following examples compute trajectories entirely online (i.e. no advance path-planning) and assume no global velocity bound. All simulation code for this section may be found at https://github.com/jbreeden-um/phd-code/tree/main/2021/Automatica%20Robust%20CBFs%20for%20Satellite%20Trajectories.

4.5.2 Robust CBF Setup

The first proposed mission centers around a flyby of the asteroid Ceres. A second simulation around the asteroid Eros is also included in Section 4.5.4. The safe set $\mathcal{S}$ is the set of states with positions r sufficiently far from the asteroid and arbitrary velocities v, while the sets $\mathcal{H}$, $\mathcal{H}_{\text{res}}$ add restrictions on v. Safety is encoded by the constraint function

$$\kappa_c(t, x) \triangleq \rho - \|r - r_c(t)\|, \tag{4.64}$$

where $r_c : \mathcal{T} \to \mathbb{R}^3$ is the point to be avoided and ρ is the minimum allowable distance. Suppose this point has known velocity $v_c : \mathcal{T} \to \mathbb{R}^3$ and acceleration $u_c : \mathcal{T} \to \mathbb{R}^3$. The derivatives of κ_c following from (4.19) are as follows.

$$\dot{\kappa}_{w,c}(t, x) = -\frac{(r - r_c(t))^{\mathrm{T}}(v - v_c(t))}{\|r - r_c(t)\|} + w_{x,\max} \tag{4.65}$$

$$\begin{aligned}
\ddot{\kappa}_{w,c}(t, x, u, w_u) &= -\frac{(r - r_c(t))^{\mathrm{T}}(f_\mu(t, r) + u + w_u - u_c(t))}{\|r - r_c(t)\|} \\
&\quad + \frac{(r - r_c(t))^{\mathrm{T}}(v + w_x - v_c(t))^\times (v - v_c(t))^\times (r - r_c(t))}{\|r - r_c(t)\|^3} \\
&\leq -\frac{(r - r_c(t))^{\mathrm{T}}(f_\mu(t, r) + u + w_u - u_c(t))}{\|r - r_c(t)\|} \\
&\quad + \frac{w_{x,\max}^2}{4\rho} - \frac{\|(r - r_c(t))^\times (v - v_c(t))\|^2}{\|r - r_c(t)\|^3}
\end{aligned} \tag{4.66}$$

For this mission, I assume that only low-thrust actuators are available, meaning that $u_{\max} \ll \|f_\mu\|$ in the vicinity of the asteroid. Specifically, let $u_{\max} = (10)^{-4}$ m/s^2, which is approximately the peak acceleration achievable by the DAWN spacecraft halfway through its mission[1], or a modern SmallSat ion thruster[2]. The radius of Ceres is approximately $\rho_{\text{Ceres}} = 476000$ m, and gravitational

[1] Ref: https://solarsystem.nasa.gov/missions/dawn/technology

[2] E.g. Busek Bit-3 on a 12 kg CubeSat: http://www.busek.com/technologies__ion.htm

RCBF	Theorem	Form	ρ	Parameter
h_1^A	4.10	(4.22)	$3.63(10)^7$	$a_{\max}$ in (4.68)
h_2^A	4.11	(4.27)	$3.21(10)^7$	Φ_c in (4.70)
h_3^A	4.16	(4.37)	$2.50(10)^7$	u^*_{rad} in (4.71)
h_4^A	4.15	(4.37)	$4.76(10)^5$	u^*_{orth} in (4.72)

Table 4.1: Summary of RCBFs for Ceres Simulation. A summary of the 4 RCBFs tested and the parameters used for simulations around Ceres

acceleration near Ceres is given by

$$f_{\mu,c}(t,r) = -\frac{\mu(r-r_c)}{\|r-r_c(t)\|^3}\,, \tag{4.67}$$

where $\mu = 6.26325(10)^{10}$ is fixed. As the following simulations all take place in a frame centered around Ceres, let $u_c = 0$.

Simulations were run with 4 different RCBFs, summarized in Table 4.1 and detailed as follows. First, I apply Theorem 4.10. Substituting the dynamics (4.63) into (4.24), the parameter $a_{\max}$ is given by

$$a_{\max} = u_{\max} - \frac{w_{x,\max}^2}{4\rho} - w_{u,\max} - \sup_{t\in\mathcal{T},x\in\mathcal{S}(t)} \|f_\mu(t,r)\|. \tag{4.68}$$

One possible RCBF, which I denote h_1^A, is then given by (4.22) with $\kappa_c, \dot\kappa_{w,c}$ in place of $\kappa, \dot\kappa_w$, respectively. Because $u_{\max} \ll \|f_\mu\|$ near the asteroid, one must choose ρ large enough that $a_{\max}$ in (4.68) is positive. Specifically, I chose ρ as $\rho_1^A = 3.63(10)^7$ m, which along with $w_{u,\max} = 5(10)^{-6}$ m/s^2 leads to $a_{\max} = 4.55(10)^{-5}$ m/s^2 in (4.68).

Second, I note that $\ddot\kappa_{w,c}$ in (4.66) satisfies

$$\max_{\|w_u\|\le w_{u,\max}} \inf_{u\in\mathcal{U}} \ddot\kappa_{w,c}(t,x,u,w_u) \le \frac{\mu}{\|r-r_c(t)\|^2} - u_{\max} + w_{u,\max} + \frac{w_{x,\max}^2}{4\rho}. \tag{4.69}$$

Since $\|r - r_c(t)\| = \rho - \kappa_c(t,x)$, the right hand side of (4.69) can be written as a function $\phi_c(\kappa_c(t,x))$ only dependent on κ_c and constants. The anti-derivative of ϕ_c is then

$$\Phi_c(\lambda) = \frac{\mu}{\rho-\lambda} + (w_{u,\max} + \tfrac{w_{x,\max}^2}{4\rho} - u_{\max})\lambda\,. \tag{4.70}$$

Applying Theorem 4.11, a second possible RCBF, which I denote h_2^A, is then given by (4.27) with $\kappa_c, \dot\kappa_{w,c}, \Phi_c$ in place of $\kappa, \dot\kappa_w, \Phi$, respectively. Note that Theorem 4.11 requires that the right hand side of (4.69) is always negative, so I chose ρ as $\rho_2^A = 3.21(10)^7$ m. In this system, the function Φ_c in (4.70) follows directly from f_μ being a potential force, and while that is not necessary in

general, it may be hard to find a function Φ satisfying the conditions of Theorem 4.11 for systems without known expressions for potential energy. On the other hand, it is comparatively easy to find a constant $a_{\max}$ (or show than none exists) using (4.24) even for complex systems.

Note that one also could have used the equivalent constraint function $\kappa_{\text{alt}}(t,x) \triangleq \rho^2 - \|r - r_c(t)\|^2$. However, no valid $a_{\max}$ or Φ exists for κ_{alt}, so one may need to choose the constraint function carefully as well. Intuitively, κ_c represents position inside a potential field, while κ_{alt} has no physical interpretation in the context of the dynamics in (4.63).

Third, I apply Theorem 4.15 for two different control laws u^*. Note that the constructions of both $a_{\max}$ in (4.68) and Φ in (4.70) ignore the effect of the second term of $\ddot{\kappa}_{w,c}$ in (4.66), which is always nonpositive (i.e. helpful to safety), but not amenable to simple RCBF formulas. Theorem 4.15 allows for accounting of this term as well, and thus decreases conservatism. First, define the control law

$$u^*_{\text{rad}}(t,x) \triangleq \underset{u \in \mathcal{U}_w}{\arg\min} -(r - r_c(t))u\,, \tag{4.71}$$

where $\mathcal{U}_w$ is given in (4.35). This is a special case of u^*_{opt} in (4.58). In addition to capturing the effect of the second term of (4.66), using Theorem 4.15 with this control law also captures how the spacecraft has more control authority when $r - r_c$ has components in the direction of more than one thruster. A third possible RCBF, which I denote h_3^A, is then given by (4.37) with $\kappa_c, u^*_{\text{rad}}$ in place of κ, u^*, respectively.

Lastly, define the control law

$$u^*_{\text{orth}}(t,x) \triangleq \underset{u \in \mathcal{U}_w}{\arg\min}\, v_{\text{orth}}(t,x)u\,, \tag{4.72}$$

where

$$v_{\text{orth}}(t,x) \triangleq v - v_c(t) - \frac{(r - r_c(t))^{\mathrm{T}}(v - v_c(t))}{\|r - r_c(t)\|^2}(r - r_c(t))\,. \tag{4.73}$$

While u^*_{rad} thrusts away from the asteroid, the control law u^*_{orth} thrusts tangential to the asteroid, thereby making the second term of (4.66) (which does not depend directly on u) more negative. A fourth possible RCBF, which I denote h_4^A, is then given by (4.37) with $\kappa_c, u^*_{\text{orth}}$ in place of κ, u^*, respectively. This choice of u^* is motivated by orbital dynamics, and intuitively, $h_4^A(t_0, x(t_0)) \leq 0$ implies that a safe orbit can be established from $(t_0, x(t_0))$.

Note that it still needs to be verified that u^*_{rad} and u^*_{orth} satisfy the requirement of Theorem 4.15 that the set $\mathcal{B}(t,x)$ in (4.36) contains at most one nonzero element. Assuming a constant $a_{\max} > 0$ as in (4.68) exists, it follows that $\ddot{\kappa}_c(t,x,u^*_{\text{rad}}(t,x),w_u) \leq -a_{\max}$, so $\mathcal{B}(t,x)$ in (4.36) always has exactly one element, and thus meets the requirements of Corollary 4.16. However, u^*_{rad} can still be used even if an $a_{\max}$ does not exist (i.e. if one chooses ρ small enough that there exists states $(t,x) \in \mathcal{S}^{\mathcal{T}}$ such that $\ddot{\kappa}_c(t,x,u^*_{\text{rad}}(t,x),w_u) > 0$). That said, if this is the case, then Theorem 4.15

must be applied very carefully. This is because there exists a manifold of initial conditions (t_a, x_a) for which $\chi(t, t_a, x_a), t \geq t_a$ is a closed periodic orbit. Safe trajectories exist on both sides of this manifold, but along this manifold, $\mathcal{B}(t_a, x_a)$ has an infinite number of elements, which violates the assumption of Theorem 4.15. For this reason, here I let $\rho_3^A = 2.50(10)^7$ m, which places this manifold entirely outside the safe set.

When using u^*_{orth}, there is no constant $a_{\max}$ which bounds $\ddot{\kappa}_c$ (or any higher derivatives), so proving that $\mathcal{B}(t, x)$ in (4.36) is nonempty and contains at most one nonzero element is challenging and is an open area for future work. In practice, many nominal evading maneuvers $u^* : \mathcal{T} \times \mathcal{X} \to \mathcal{U}_w$ may not allow for straightforward proofs of how many maximizers of $\kappa(t + \beta, \chi(\beta, t, x))$ the system admits. In these cases, how to prove or disprove the applicability of Theorem 4.15 is an open research question. For this particular system, my strategy was to examine sample trajectories of $\kappa(t + \beta, \chi(\beta, t, x))$ according to (4.31) and originating from many different states (t, x). I verified numerically that each trajectory 1) remain bounded, 2) achieved its upper bound, and 3) had at most one nonzero local maximizer. I also sought out corner cases that might violate these three conditions. As no trajectory violating these conditions could be found, I conclude that u^*_{orth} is consistent with the conditions on $\mathcal{B}(t, x)$ in Theorem 4.15, though an analytic proof of this observation would provide a stronger guarantee. One also needs to ensure that the initial conditions are such that $\|v_{\text{orth}}\|$ is never zero along the trajectories, so u^*_{orth} is uniquely defined. Thus, u^*_{orth} meets the requirements of Theorem 4.15 for any $\rho > 0$. For this simulation, I chose $\rho_4^A = 476000$ m, which is the radius of Ceres.

In summary, I have constructed four RCBFs $h_1^A, h_2^A, h_3^A, h_4^A$. Each RCBF was constructed using specific properties of the system and the input constraints, so given an initial condition $(t_0, x(t_0)) \in \mathcal{H}_{\text{res}}$ for any of these RCBFs, it is guaranteed that there exists at least one safe trajectory beginning from $(t_0, x(t_0))$ along which the control input always satisfies the input constraints. Moreover, the spacecraft will remain on a safe trajectory as long as the control input satisfies the RCBF condition $u \in \mathcal{U}_{\text{rcbf}}$ in (4.8). Thus, even though the trajectories are not known in advance, one still knows that the trajectories will stay within the set $\mathcal{H}_{\text{res}}$.

4.5.3 Simulations: Ceres

Next, I validate the RCBFs $h_1^A, h_2^A, h_3^A, h_4^A$ from Section 4.5.2 in simulation. Suppose a control input of the form

$$u(t, x) = \underset{u \in \mathcal{U}_s(t,x,\sigma)}{\arg\min} \|u - u_{\text{nom}}(x)\|^2, \tag{4.74}$$

where there is a nominal control input u_{nom} that may be unsafe and $\mathcal{U}_s$ is the set of allowable control inputs. In this case, the nominal control input is the linear control law

$$u_{\text{nom}}(x) = -k_p(r - (r \cdot \hat{e}_x)\hat{e}_x) - k_d \left(v - \sqrt{\tfrac{2\mu}{\|r\|} + 10^4}\hat{e}_x\right) \tag{4.75}$$

where $k_p = 1.2(10)^{-11}$ and $k_d = 6(10)^{-5}$, and $\hat{e}_x$ is the x-axis unit vector. That is, the nominal control input tries to drive the spacecraft along the x-axis and through the center of Ceres, which is an unsafe state. The set $\mathcal{U}_s$ captures the switching described in Section 4.4 and is given by

$$\mathcal{U}_s(t, x, \sigma) = \begin{cases} \mathcal{U} & \sigma = 0 \\ \mathcal{U}_{\text{rcbf}}(t, x) & \sigma = 1 \end{cases}. \tag{4.76}$$

The switching tolerances are $\epsilon_1 = 5(10)^4$ m and $\epsilon_2 = 1.5(10)^5$ m, and the class-$\mathcal{K}_e$ function used to determine $\mathcal{U}_{\text{rcbf}}$ was chosen as α_r in (4.61). Note how the quadratic program in (4.74) contains only 3 degrees of freedom, and is thus much less computationally expensive than most MPC-based controllers.

I ran four simulations using the above controller with the four different RCBFs $h_1^A, h_2^A, h_3^A, h_4^A$ to specify $\mathcal{U}_{\text{rcbf}}$ and one simulation with $\mathcal{U}_{\text{rcbf}} = \mathcal{U}$ (i.e. with no RCBF) for comparison. All five simulations had a random zero-mean matched disturbance of magnitude upper bounded by $w_{u,\max} = 5(10)^{-6}$ m/s^2. The simulations using h_1^A, h_2^A, and no RCBF also had a random zero-mean unmatched disturbance of magnitude upper bounded by $w_{x,\max} = 2(10)^{-6}$ m/s (recall that Theorem 4.15 requires $w_{x,\max} = 0$ for h_3^A, h_4^A). The initial condition was $x_0 = [-6(10)^7, -10^6, 0, 20, -2, 0]^{\mathrm{T}}$ in all five simulations, and each was run for 69 simulated days, taking on the order of 10 minutes to compute. The resultant trajectories are shown together in Fig. 4.5.

Note how the trajectories under h_1^A, h_2^A, h_3^A come progressively closer to Ceres because conservatism is reduced with each added layer of complexity. The distance along the x-axis traveled in the same amount of time also increases, as the satellite spends less time reshaping its trajectory around Ceres. Physically, the RCBF derivative $\dot{h}$ for h_1^A, h_2^A, h_3^A is minimized when the satellite thrusts away from the asteroid. On the other hand, the RCBF derivative for h_4^A is minimized when the satellite thrusts tangentially to the asteroid. As a result, the trajectory under h_4^A is able to keep h_4^A nonpositive while approaching far closer to Ceres than any of the other RCBFs, as shown in the zoomed-in plot in Fig. 4.6. Here, the satellite comes so close to Ceres that it is redirected by Ceres' gravity (which is much greater than $u_{\max}$), but the satellite is moving fast enough to avoid being pulled outside the safe set, unlike the trajectory with no RCBF, which crashes into the surface of Ceres. In practice, one might wish to choose a larger ρ_4^A to result in a hyperbola around Ceres

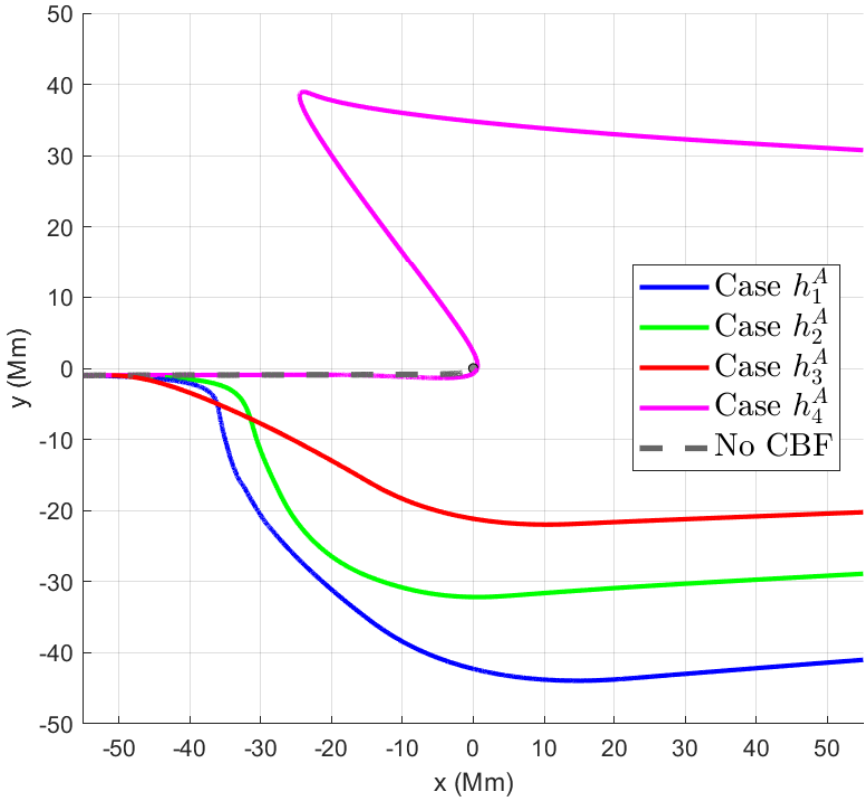

Figure 4.5: Ceres Flyby Simulation Trajectories. Trajectories under each of the RCBFs moving from left to right. Ceres is located at the grey dot at the origin and is not to scale. Note how the different RCBFs allow their respective trajectories to approach within differing distances of the asteroid and how the magenta trajectory approaches close enough to be redirected by Ceres' gravity. The grey and magenta trajectories trace very similar paths.

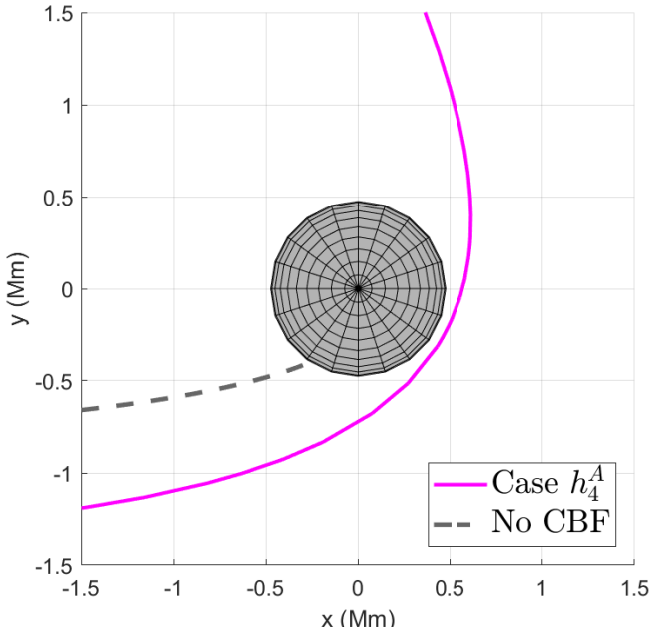

Figure 4.6: Ceres Flyby Close Up. Zoomed-in view of Fig. 4.5 with Ceres to scale (trajectories moving from left to top)

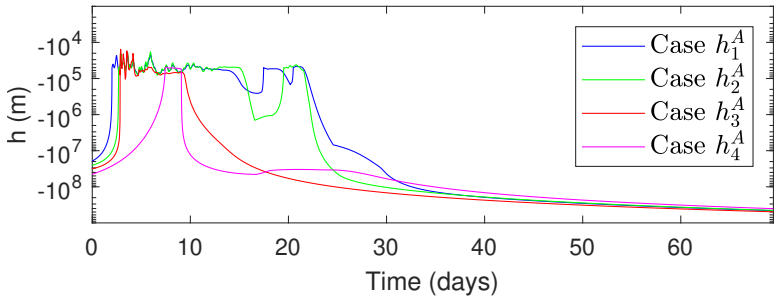

Figure 4.7: Ceres Flyby CBF Values and Asymptotic Surface. The RCBF values along the trajectories in Fig. 4.5. Note how each trajectory converges approximately to $h = -\epsilon_1$ due to the choice of $\alpha = \alpha_r$ in (4.61), but the corresponding distances to the asteroid are different in Figs 4.5,4.8 due to the differing constructions of the RCBFs.

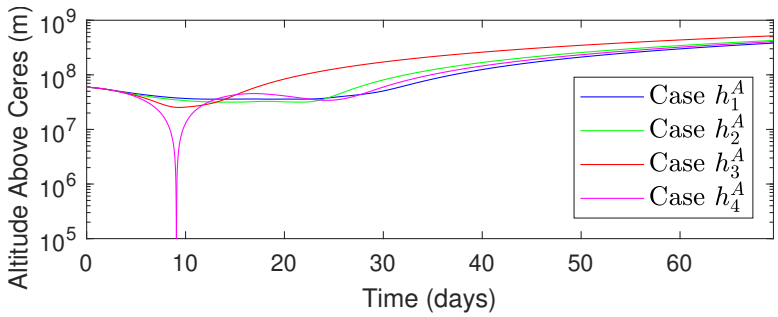

Figure 4.8: Ceres Flyby Altitude versus Time. The altitudes above Ceres of the trajectories in Fig. 4.5

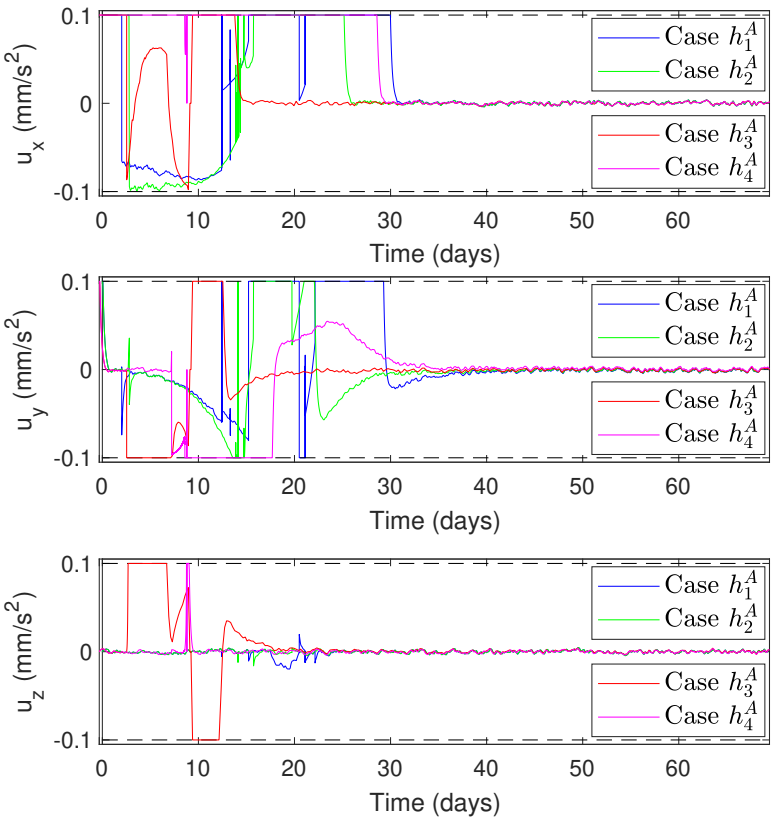

Figure 4.9: Ceres Flyby Control Inputs. Control inputs along the trajectories in Fig. 4.5. The controller (4.74) remained feasible in the presence of input constraints, as expected from Theorems 4.10,4.11,4.15.

with a smaller turn angle than in Fig. 4.6, but the RCBF ensures the trajectory does not impact the asteroid regardless.

The values of the RCBFs along these four trajectories are shown in Fig. 4.7. Note how all four trajectories initially approach the surface $\{x \in \mathcal{H}_{\text{res}} \mid h(t,x) = -\epsilon_1\}$ and then remain near this surface until at least $t = 9$ days. Next, the distance to Ceres is plotted in Fig. 4.8. As expected from Fig. 4.5, the trajectory under h_4^A came the closest to the surface, which may be desirable for an inspection mission, though the satellite was moving very fast during its closest approach. Finally, the control inputs are shown in Fig. 4.9, and indeed stay within the specified bounds. The z-axis control inputs are negligible, except under h_3^A and h_4^A, as u_{rad}^* and u_{orth}^* have the potential to magnify the effects of disturbances in the z direction.

If this were a high-thrust scenario (i.e. $u_{\max} > \|f_\mu\|$), any of these four RCBFs would allow the satellite to get equally close to Ceres (in differing time spans), but since the actuators are assumed limited to low-thrust, the choice of RCBF and resultant conservatism made a significant difference in the trajectories. In particular, the RCBF motivated by orbital dynamics, h_4^A, allowed for the

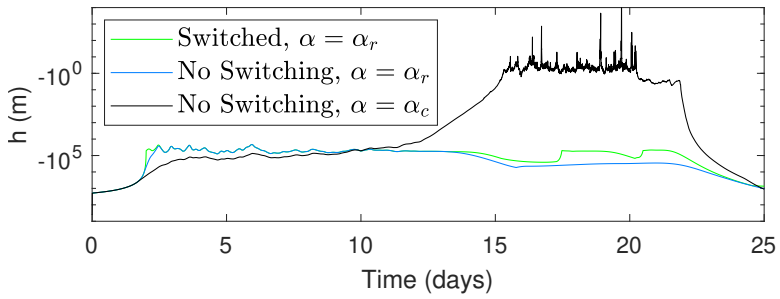

Figure 4.10: Ceres Flyby Comparison of Switched and Unswitched Control Laws. The RCBF values under h_1^A with and without switching and with α both as α_r in (4.61) and as a "regular" class-$\mathcal{K}_e$ function α_c.

closest approach to the asteroid. The RCBF h_3^A and its derivatives θ, Θ took between 0.25-12 ms to compute, while h_4^A and its derivatives took between 0.36-560 ms to compute on a 3.5 GHz computer, though the code for these calculations could likely be further optimized. In particular, u^*_{orth} often results in χ trajectories that travel around Ceres many times, and thus require small time steps to accurately propagate using this state representation (e.g., computation time could likely be improved by using Keplerian elements as a state representation). This computation time is also very small relative to time-scale of the problem.

Finally, I considered the effect of switching and the choice of function α using the RCBF h_1^A. First, I ran an additional simulation with $\alpha = \alpha_r$ without switching the RCBF (i.e. $\sigma(t) = 1, \forall t \in \mathcal{T}$). The resultant trajectory was qualitatively similar, and a comparison of the RCBF values is shown in Fig. 4.10. As shown in Fig. 4.10, the trajectory with switching reached the surface $\{x \in \mathcal{H}_{\text{res}} \mid h(t,x) = -\epsilon_1\}$ 0.4 days ahead of the trajectory without switching, as expected since the growth rate was not limited by α_r until the satellite came close to the boundary of $\mathcal{H}^A_{\text{res},1}$. Second, I ran a simulation with α equal to the "regular" class-$\mathcal{K}_e$ function α_c, where $\alpha_c(\lambda) \triangleq k\lambda$, again without switching. I let k equal the average value of $\frac{W}{\epsilon_1}$ when h_1^A was active, which was $k = 3.42(10)^{-5}$. Again, the shape of the resultant trajectory around Ceres was qualitatively similar, but Fig. 4.10 shows that the two trajectories (switched and unswitched) using $\alpha = \alpha_r$ in (4.61) approached $-\epsilon_1$ much faster than the trajectory using α_c. However, the trajectory under $\alpha = \alpha_c$ approached much closer to the edge of the safe set than the other two trajectories, because the analysis of the steady state surface $\mathcal{S}_f$ following (4.62) no longer applies.

4.5.4 Simulations: Eros

Next, I consider the problem of finding a safe trajectory when the spacecraft is near an irregularly-shaped asteroid, in this case Eros. I consider a mesh model of Eros with $N = 3897$ points [208], shown in Fig. 4.11, where the satellite should stay sufficiently

far from every point in the mesh. Note that Eros spins with angular velocity $\omega = [3.101(10)^{-4}, 6.232(10)^{-5}, 9.810(10)^{-5}]$ rad/s [214], so each point is moving and thus the time varying component $\partial_t h$ is important in this scenario (this model of Eros is identical to the model used in Chapter 3, except for the addition of asteroid spin, which was not present in the simulations in Chapter 3). Denote each point as $r_{c,i},\ i = 1, 2, \cdots N$. Then I construct N time-varying constraint functions

$$h_{c,i}(t, x) \triangleq \rho - \|r - r_{c,i}(t)\|, \tag{4.77}$$

$$r_{c,i}(t) = e^{\omega^\times (t-t_0)} r_{c,i}(t_0), \tag{4.78}$$

where $\omega^\times$ is the cross product matrix for ω. For this scenario, let ρ be $\rho^B = 500$ m, which represents the closest allowable distance to any point r_c in the mesh. Note that the maximum distance between points in the model [208] is 850 m, so one should require that ρ be at least half this distance (otherwise, the agent could travel inside the asteroid between mesh points), and one can expect smoother results when ρ and/or N are larger.

I model the gravity of Eros f_μ using the 16th order spherical harmonics model in [209]. The full model is assumed known to the controller in the present simulation, but note that one could also use a simplified model and account for higher order effects as disturbances (e.g. $f_\mu = -\frac{\mu}{\|r\|^3} r + \xi$, as the harmonics of greater order than μ are difficult to obtain prior to orbiting the asteroid), as was done in my preliminary work [203] (which has not been reproduced in this dissertation). Since Eros is rotating and asymmetric, f_μ will also be time-varying. Let $u_{\max} = 0.1$ m/s^2, which is larger than the peak gravitational acceleration at the surface of Eros, so this is a high-thrust simulation. Suppose there are random matched and unmatched disturbances upper bounded by $w_{u,\max} = 0.005$ m/s^2 and $w_{x,\max} = 0.001$ m/s, respectively.

Using these N constraint functions, I construct N RCBFs of the form given in (4.22), denoted $h_i^B, i = 1, 2, \cdots N$. Here, the parameter $a_{\max}$ is the same for every RCBF and is

$$a_{\max} = u_{\max} - \frac{w_{x,\max}^2}{4\rho} - w_{u,\max} - \sup_{t \in \mathcal{T}, x \in \mathcal{S}(t)} \|f_\mu(t, r) - u_c(t)\|, \tag{4.79}$$

where the asteroid surface acceleration $u_c \neq 0$ since every mesh point is moving. This results in $a_{\max} = 0.0523$ m/s^2. One could also have found a function Φ and used the form of RCBF in (4.27), but this would be less beneficial here than in Section 4.5.3 since the agent will always be close to the asteroid surface. I also construct a separate discrete state for each RCBF to describe whether each RCBF is active, which I concatenate into the vector $\Sigma \in \mathbb{Z}^N$, containing N discrete states σ_i. Each RCBF induces a set of allowable control inputs $\mathcal{U}_{\text{rcbf},i}$, so the control inputs must live in the

intersection of the allowable sets following from each active RCBF. To this end, define

$$\mathcal{U}_s(t,x,\Sigma) = \bigcap_{\{i|\sigma_i=1\}} \mathcal{U}_{\text{rcbf},i}(t,x)\,, \tag{4.80}$$

and let $\mathcal{U}_s$ equal $\mathcal{U}$ if there is no active RCBF. I assume that $\mathcal{U}_s$ is always nonempty (see Chapter 6 for more information about how to guarantee this assumption). The controller is then

$$u(t,x) = \underset{u\in\mathcal{U}_s(t,x,\Sigma)}{\arg\min} \|u-u_{\text{nom}}\|^2\,, \tag{4.81}$$

$$u_{\text{nom}}(x) = -k_p(r-r_t) - k_d v \tag{4.82}$$

where $k_p = 3(10)^{-5}$, $k_d = 0.03$, and $r_t = [20(10)^3,\, 0,\, 0]^{\text{T}}$. Let the switching tolerances be $\epsilon_1 = 100$ m and $\epsilon_2 = 300$ m, and the class-$\mathcal{K}_e$ functions required to define each $\mathcal{U}_{\text{rcbf},i}$ are all identical and given by α_r in (4.61).

As observed in [151] and in Section 3.4.2, one only needs to enforce each RCBF condition when it is close to being violated. This is facilitated by the switching approach in Section 4.4 and (4.80), so the quadratic program in (4.81) never has all 3897 constraints active simultaneously. However, unlike in [151], the use of hysteresis-switching allows for this simplification without requiring non-smooth analysis.

I then simulated the above controller and all 3897 RCBFs around Eros, starting from initial condition $x_0 = [-20(10)^3,\, -4(10)^3,\, 0,\, 1,\, 1,\, 0]^{\text{T}}$. The value of the maximum of the 3897 RCBFs is shown in Fig. 4.12. In this simulation, there were never more than 4 RCBFs active simultaneously. The trajectory around the asteroid in an Eros-fixed frame is shown in Fig. 4.11 and a video in the inertial frame that also highlights the active RCBFs can be found at https://youtu.be/ArQ84sdMTqo. The control inputs are shown in Fig. 4.13. As expected, the spacecraft stays safe for all time, despite the natural motion of the asteroid and the nominal linear controller attempting to drive the trajectory through the asteroid. Also, the maximal RCBF value stays very close to $-\epsilon_1$ during the interval $t \in [110, 2760]$ in Fig. 4.12. The choppiness of the control input may be attributed to the sparsity of the mesh relative to the size of the asteroid, and the relatively small values of ρ and ϵ_1.

4.6 Conclusions on Robust CBFs

This chapter has presented three forms for RCBFs for relative-degree 2 systems that constructively consider input constraints and disturbance bounds to ensure the RCBF condition is always feasible in the presence of input constraints. Thus, systems meeting the theorem requirements

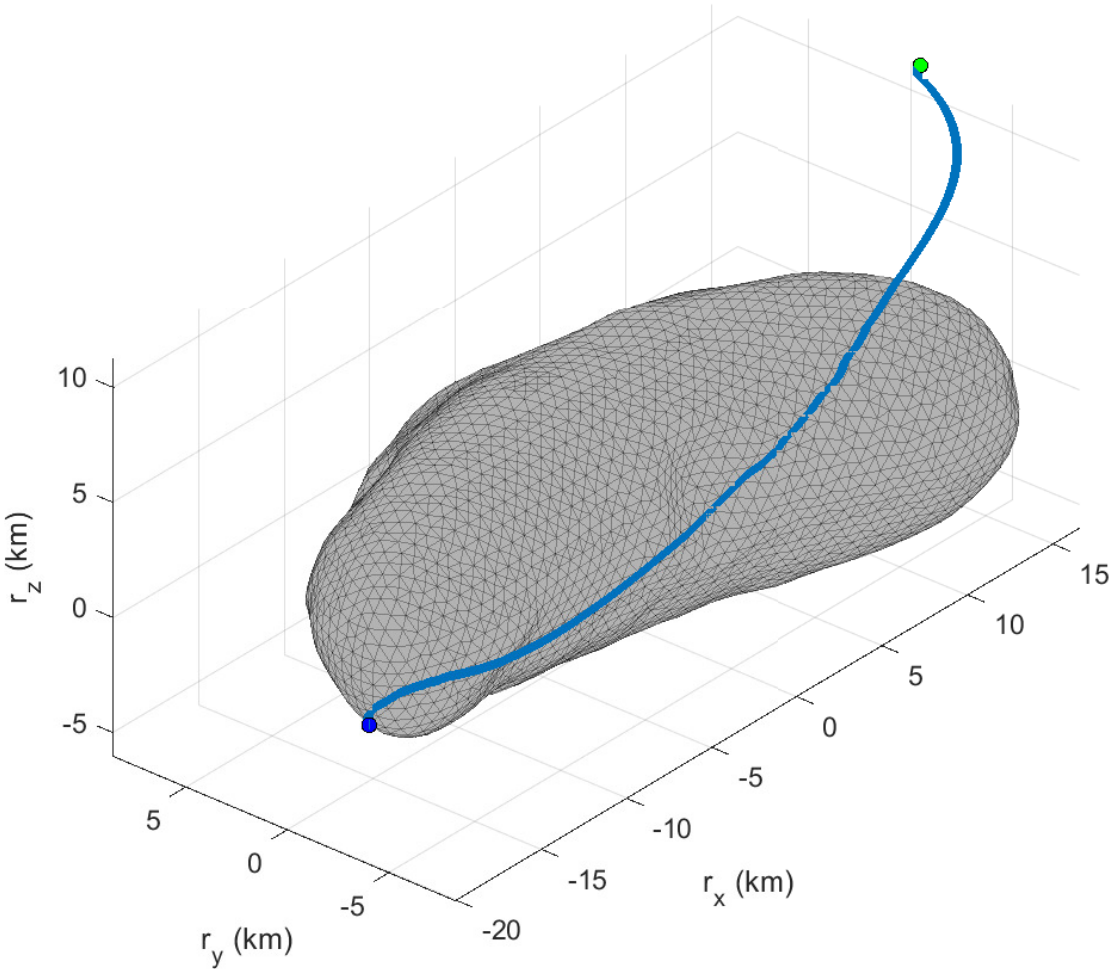

Figure 4.11: Eros Close Approach Trajectory (Rotating). Trajectory of the spacecraft around Eros

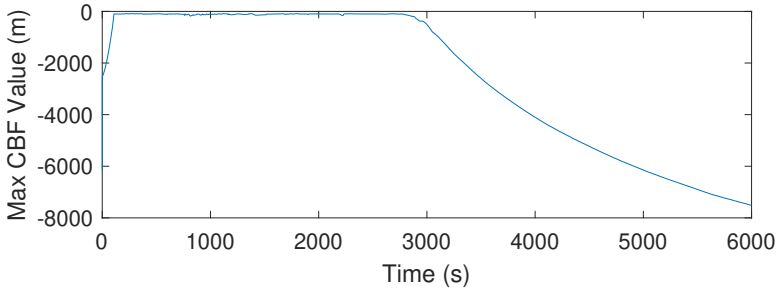

Figure 4.12: Eros Close Approach CBF Values. Maximum of the 3897 RCBF values

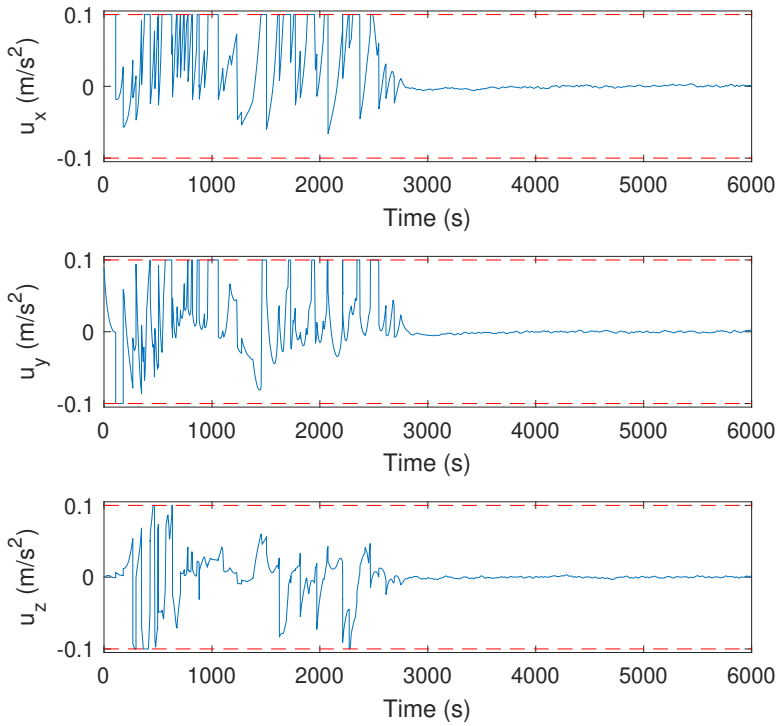

Figure 4.13: Eros Close Approach Control Inputs. Control inputs of the spacecraft around Eros

are guaranteed to have safe closed-loop trajectories despite the input constraints and disturbances. This chapter also introduced a switching approach for enforcing the RCBF condition only near the boundary of the CBF set, and a class-$\mathcal{K}_e$-like function that allows one to predict how close the state will approach the boundary of this set as a function of the disturbance. Finally, I applied these methods to create five spacecraft-relevant RCBFs and demonstrated these RCBFs in simulation. The simulations show that such RCBFs can be used to plan safe trajectories online, though three of the flyby trajectories were overly conservative. In these simulations, the nominal linear control laws were very simple, so the resultant control inputs were not fuel-efficient. Increasing this efficiency is one area for future work, partially considered in Chapter 7.

Finally, in the following section, I present an extension of the above RCBF formulation for a specific type of problem motivated by the observations in Section 4.4. I previously noted that using the class-$\mathcal{K}_e$-like function α_r in (4.61) allowed one to a prior determine to which surfaces the trajectories of (4.1) would converge in the presence of a worst-case disturbance, no disturbance, or a best-case (i.e. safety encouraging) disturbance. Because these surfaces are asymptotically stable, a trajectory starting on or between these three surfaces will stay between these surfaces, and Section 4.4 showed that one could determine a priori the location of these surfaces by choice of ϵ_1. This is used to deliberate effect in the following definition of "tight tolerance problems", including a case study applied to the common problem of satellite docking.

4.7 Tight Tolerance Specifications Using RCBFs

This section is concerned with the application of RCBFs as in Definition 4.2 to the specific problem of spacecraft docking, and more generally to the problem of "tight-tolerance objectives". The presence of disturbances prevents convergence of the satellites/spacecraft to a unique docking state, so instead docking is defined as occurring within a set constructed using prescribed tolerances. Forward invariance of the set of non-colliding states in the presence of any considered disturbance is ensured using RCBFs. However, as I will show, the RCBF strategy necessarily presumes a worst-case disturbance, and thus, if a worst-case disturbance is not present, this strategy will restrict trajectories to a smaller subset of the interior of the CBF set, which may have no overlap with the set of docking states. Thus, this section presents a method for ensuring that the set of docking states has non-empty intersection with the set of reachable states under a robust control law, for any considered disturbance. The control law provably achieves docking in finite time regardless of the disturbance and is validated in a docking simulation in Section 4.7.4.

4.7.1 Introduction to Spacecraft Docking Problem

A common requirement for spacecraft systems is the capability for one spacecraft, called the *chaser*, to dock with another spacecraft, called the *target*. Successful docking requires the satisfaction of several tolerances, such as a minimum and maximum relative velocity, maximum displacement between docking mechanisms, and maximum spacecraft attitude deviation, among others. In the remainder of this chapter, I propose encoding these tolerances as Control Barrier Functions (CBFs) [66] and applying CBF theory to guarantee tolerance satisfaction.

There are two principal approaches to robustness with CBFs. First, a CBF can be designed to drive state trajectories that lie outside the allowable *safe set* into this set [112, 116]. Second, a CBF can be designed to ensure that a disturbance with a known upper bound never causes the state to leave the safe set [117, 118]. In this chapter, I have thus far utilized the latter approach, and will continue to follow this construction so that docking tolerances are never violated. However, tight-tolerance objectives, such as docking, in principle require that the system operate very close to the boundary of its safe set (e.g. the chaser comes very close to the target). In the presence of a disturbance pointing toward the interior of the safe set, or no disturbance, an RCBF may prevent the system from approaching sufficiently close to the boundary of its safe set to execute its mission. Thus, this section develops conditions under which one can guarantee that system trajectories always remain safe, yet also approach to within the required proximity to the boundary of the safe set in finite time. While this section is most focused on spacecraft docking to other spacecraft, or landing on celestial bodies, the developed approach can be applied to operating any system near the boundary of a safe set using RCBFs.

Autonomous spacecraft rendezvous and docking has been extensively studied, and addressed using several methods, including artificial potential fields (APFs) [21, 25, 215, 216], path planning [217], model predictive control [65], sliding mode control [218], reinforcement learning [219], and linear control [220], among others. While the fundamental problem almost always centers on the Hill-Clohessy-Wiltshire (HCW) dynamics, different authors have considered various constraints. The work in [215, 218] specifically considers APFs for coupled rotation and translation, while [216, 217] consider fuel efficiency as well. The work in [220] considers adaptation to uncertain model parameters, and [217] considers a tumbling target. However, most of the aforementioned works attempt to accomplish docking exactly, or simply report the achieved tolerances when disturbances are added, rather than provably guaranteeing satisfaction of docking tolerances. Such guarantees can be obtained systematically through RCBFs, which can be used in conjunction with all of the above methods and constraints.

The preceding work in this chapter unifies the topics of input constraint satisfaction and disturbance rejection into a single RCBF methodology. This section extends the preceding work by considering the case when the spacecraft mission and safety requirements are opposite to each other, as in the case of docking, and thus require operations within tight tolerances. To address this problem, the following sections present precise definitions of "landing" and "docking" in an RCBF context, and then present methods of accomplishing landing and docking within prescribed tolerances. Section 4.7.4 then presents simulations for a spacecraft landing on an asteroid with nontrivial gravity, and docking with another spacecraft in low Earth orbit.

4.7.2 Preliminaries

In addition to the notations previously presented, for brevity, define the function $\mathrm{ssq} : \mathbb{R} \to \mathbb{R}$ as $\mathrm{ssq}(\lambda) = \lambda|\lambda|$, and recall from Lemma 4.9 that ssq is continuously differentiable and invertible everywhere on $\mathbb{R}$.

The remainder of this chapter continues to work with the model (4.1), but instead assumes that $\mathcal{T} = [t_0, t_f]$ is a closed set, as I seek to demonstrate docking in finite time. Let the function $\kappa : \mathcal{T} \times \mathcal{X} \to \mathbb{R}$ be a metric for distance between the chaser and target agents, defined so that $\kappa < 0$ when the agents are separated, and $\kappa = 0$ when the agents are in contact. When $\dot{\kappa} > 0$, the chaser is approaching the target. Assume that $\kappa \in \mathcal{G}^2$, as in Section 4.3.1-4.3.2, and that $\nabla\kappa$ does not vanish. I define landing and docking mathematically as follows.

Definition 4.3 (Landing)**.** *The state x is said to correspond to* landing *at time t if $\kappa(t, x) = 0$ and $\dot{\kappa}(t, x) \in [0, \gamma_2]$ simultaneously, where $\gamma_2 \geq 0$ is a specified constant.*

Definition 4.4 (Docking)**.** *The state x is said to correspond to* docking *at time t if $\kappa(t, x) = 0$ and $\dot{\kappa}(t, x) \in [\gamma_1, \gamma_2]$ simultaneously, where $\gamma_1, \gamma_2 > 0$ are specified constants.*

Figure 4.14: Visualization of Docking Requirements

Note that Definitions 4.3-4.4 pose the landing/docking objective as the value of $\dot{\kappa}$ belonging to an interval, rather than requiring a single value of $\dot{\kappa}$. Landing and docking differ only by the requirement of a minimum velocity $\gamma_1 > 0$ for docking. In both cases, I assume that γ_2 is sufficiently small to be dissipated by the spacecraft structure; i.e. landing/docking is a *controlled collision*. These tolerances are visualized in Fig. 4.14. The focus of this section is on ensuring that landing/docking occurs within the specified range of $\dot{\kappa}$ values in finite time. This will be accomplished in part by treating the $\dot{\kappa}$ upper bound requirement as a safety constraint, and constructing an RCBF h as in (4.27). Recall that $\dot{h}$ is not known precisely due to the disturbances, but one does know that $\dot{h}(t,x,u,w_u,w_x) \in [\dot{h}(t,x,u,0,0) - W(t,x),\ \dot{h}(t,x,u,0,0) + W(t,x)]$ where W is as in (4.7).

This section will only work with the RCBF proposed in (4.27) and Theorem 4.11, restated here as

$$h(t,x) = \Phi^{-1}\left(\Phi(\kappa(t,x)) - \frac{1}{2}\operatorname{ssq}\left(\dot{\kappa}_w(t,x)\right)\right) \tag{4.83}$$

for some function $\Phi : \mathbb{R} \to \mathbb{R}$ meeting the requirements of Theorem 4.11. Define $\mathcal{H}_{\text{res}}$ as in (4.4). Instead of α_r in (4.61), consider the more general form

$$\alpha_r(\lambda,t,x) = W(t,x)\alpha_w(\lambda) \tag{4.84}$$

for any $\alpha_w \in \mathcal{K}_e$. Note that Corollary 4.18 applies to (4.84) as well.

4.7.3 Methods for Best and Worst Case Disturbances

I divide the landing/docking problem into two parts: *robust safety* in Section 4.7.3.1 and *robust proximity* in Section 4.7.3.2. Robust safety refers to the requirement that, under any allowable disturbances w_u, w_x in (4.1), $\dot{\kappa}(t,x(t)) \leq \gamma_2$ for all t such that $\kappa(t,x(t)) = 0$. Robust proximity refers to the requirements that 1) $\kappa(t,x(t)) = 0$ for finite t, and 2) at the time when $\kappa(t,x(t)) = 0$,

it holds that 2a) $\dot{\kappa}(t, x(t)) \geq \gamma_1$ for docking, or 2b) $\dot{\kappa}(t, x(t)) \geq 0$ for landing.

4.7.3.1 Robust Safety

The set $\mathcal{H}_{\text{res}}$ as above does not contain any states such that $\kappa = 0$ and $\dot{\kappa} > 0$ simultaneously (as such states would immediately leave $\mathcal{H}_{\text{res}}$ regardless of the control input, which was previously undesirable), so for the same function Φ as in (4.27),(4.83) define the new function

$$h_{\text{dock}}(t, x) \triangleq \Phi^{-1}\left(\Phi(\kappa(t, x)) - \frac{1}{2}\,\text{ssq}\left(\dot{\kappa}_w(t, x)\right) + \delta\right) \tag{4.85}$$

for some parameter $\delta \geq 0$. This expands the set $\mathcal{H}_{\text{res}}$ to the set $\mathcal{H}_{\text{res,dock}}(t) \triangleq \{x \in \mathcal{X} \mid h_{\text{dock}}(t, x) \leq 0 \text{ and } \kappa(t, x) \leq \Phi^{-1}(\Phi(0) - \delta)\}$, where $\Phi^{-1}(\Phi(0) - \delta) \geq 0$. Unlike $\mathcal{H}_{\text{res}}$, the set $\mathcal{H}_{\text{res,dock}}$ contains docking states. First, I note that h_{dock} is also an RCBF.

Corollary 4.19 (Corollary to Theorem 4.11)**.** *Suppose the conditions of Theorem 4.11 hold for all (t, x) such that $\kappa(t, x) \leq \Phi^{-1}(\Phi(0) - \delta)$. Then $h_{\text{dock}} : \mathcal{T} \times \mathcal{X} \to \mathbb{R}$ in* (4.85) *is an RCBF on the set $\mathcal{H}_{\text{res,dock}}$ as in Definition 4.2, condition* (4.6) *is satisfied for any $\alpha \in \mathcal{K}_e$, and conditions* (4.13)-(4.14) *are both satisfied.*

The proof of Corollary 4.19 follows identical logic to that of Theorem 4.11. More importantly, the new set $\mathcal{H}_{\text{res,dock}}$ allows one to upper bound $\dot{\kappa}$ when $\kappa = 0$ as follows.

Theorem 4.20. *If a trajectory $x(\mathcal{T})$ satisfying $h_{\text{dock}}(t, x(t)) \leq 0, \forall t \in \mathcal{T}$ contains a point $(t_f, x(t_f))$ such that $\kappa(t_f, x(t_f)) = 0$, then $\dot{\kappa}(t_f, x(t_f)) \leq \sqrt{2\delta}$.*

Proof. The proof follows from the construction of h_{dock}. Suppose there exists t_f such that $\kappa(t_f, x(t_f)) = 0$ and $h_{\text{dock}}(t_f, x(t_f)) \leq 0$. Then

$$\begin{aligned}
0 \geq h_{\text{dock}}(\cdot) &= \Phi^{-1}\left(\Phi(\kappa(\cdot)) - \frac{1}{2}\,\text{ssq}\left(\dot{\kappa}_w(\cdot)\right) + \delta\right), \\
\Phi(0) &\leq \Phi\Big(\underbrace{\kappa(t_f, x(t_f))}_{=0}\Big) - \frac{1}{2}\,\text{ssq}\left(\dot{\kappa}_w(t_f, x(t_f))\right) + \delta, \\
0 &\leq -\frac{1}{2}\dot{\kappa}_w(t_f, x(t_f))|\dot{\kappa}_w(t_f, x(t_f))| + \delta.
\end{aligned}$$

Thus, $\sqrt{2\delta} \geq \dot{\kappa}_w(t_f, x(t_f)) \geq \dot{\kappa}(t_f, x(t_f))$. ■

I then choose $\delta = \frac{1}{2}\gamma_2^2$, from which it follows that any controller which renders $\mathcal{H}_{\text{res,dock}}$ forward invariant will also ensure the landing/docking velocity meets the upper bound requirement in Definitions 4.3-4.4.

4.7.3.2 Robust Proximity

Given a control law $u : \mathcal{T} \times \mathcal{X} \to \mathcal{U}$ satisfying $u(t,x) \in \boldsymbol{\mu}(t,x)$ for the RCBF (4.85), the next problem is that of ensuring the trajectory reaches a landing/docking state, denoted x_f, under any allowable disturbances w_u, w_x. Note that the state x_f is not necessarily an equilibrium of the system (4.1), so Definitions 4.3-4.4 do not require convergence to x_f. Rather, this section seeks to ensure that trajectories pass through an $x_f = x(t_f)$ meeting the criteria of the definitions. Note that this work does not consider the system evolution after the first time instance t_f when landing/docking is achieved.

Because of the disturbances, one cannot guarantee convergence of $\kappa, \dot{\kappa}, h$ to specific values. However, one can guarantee bounds on h, and by consequence κ and $\dot{\kappa}$ as well. To capture the possible impacts of the disturbances, define the set

$$\partial_\epsilon \mathcal{H}(t) \triangleq \{x \in \mathcal{H}(t) \mid h(t,x) \geq -\epsilon\} \tag{4.86}$$

$$\partial_\epsilon \mathcal{H}_{\text{res}}(t) \triangleq \{x \in \mathcal{H}(t) \cap \mathcal{S}(t) \mid h(t,x) \geq -\epsilon\} \tag{4.87}$$

where ϵ is a parameter. While Theorems 4.1, 4.3, 4.4 upper bound h (and by consequence upper bound $\dot{\kappa}$ in Theorem 4.20), the following result also lower bounds h for any allowable disturbances.

Lemma 4.21. *Suppose $h : \mathcal{T} \times \mathcal{X}$ is an RCBF on $\mathcal{H}$ as in Definition 4.2, and that there exist constants $w_1, w_2 \in \mathbb{R}_{>0}$ such that $W(t,x) \in [w_1, w_2]$ for all $(t,x) \in \mathcal{H}^{\mathcal{T}}$. Suppose the assumptions of any of Theorem 4.1, 4.3, or 4.4 hold. If the control law $u : \mathcal{T} \times \mathcal{X} \to \mathcal{U}$ satisfies*

$$\partial_t h(t,x) + \nabla h(t,x)(f(t,x) + g(t,x)u) = \alpha_r(-h(t,x), t, x) - W(t,x) \tag{4.88}$$

for all $(t,x) \in \mathcal{H}^{\mathcal{T}}$ for α_r in (4.84) for some $\alpha_w \in \mathcal{K}_e$, and the initial condition satisfies $x(t_0) \in \mathcal{H}(t_0)$, then $\lim_{t\to\infty} x(t) \in \partial_{\alpha_w^{-1}(2)}\mathcal{H}$ (i.e. $\partial_\epsilon \mathcal{H}$ as in (4.86) with $\epsilon = \alpha_w^{-1}(2)$).

Proof. First, since (4.88) applies, it follows that $u(t,x) \in \boldsymbol{\mu}(t,x)$. Moreover, since the conditions of at least one of Theorem 4.1,4.3,4.4 apply, Corollary 4.18 implies that closed-loop trajectories cannot leave $\mathcal{H}$ and thus $\limsup_{t\to\infty} h(t,x(t)) \leq 0$. Moreover, the derivative of h in the presence of disturbances is lower bounded by

$$\begin{aligned}
\dot{h}(\cdot) &= \partial_t h(t,x) + \nabla h(t,x)(f(\cdot) + g(\cdot)(u + w_u) + w_x) \\
&= \partial_t h(t,x) + \nabla h(t,x)(f(\cdot) + g(\cdot)u) + W(t,x) + \nabla h(t,x) g(\cdot) w_u + \nabla h(t,x) w_x - W(t,x) \\
&\overset{(4.88)}{=} \alpha_w(-h(t,x))W(t,x) + \nabla h(t,x) g(\cdot) w_u + \nabla h(t,x) w_x - W(t,x) \\
&\overset{(4.7)}{\geq} \alpha_w(-h(t,x))W(t,x) - 2W(t,x)\,.
\end{aligned} \tag{4.89}$$

Since $W \geq w_1 > 0$, (4.89) implies that $\dot{h}$ is strictly positive whenever $\alpha_w(-h) > 2$, or equivalently when $h < -\alpha_w^{-1}(2)$. It immediately follows that $\liminf_{t\to\infty} h(t, x(t)) \geq -\alpha_w^{-1}(2)$. Thus, as $t \to \infty$, the state $x(t)$ approaches $\partial_{\alpha_w^{-1}(2)}\mathcal{H}$. ■

Thus, regardless of the disturbance, a control law satisfying (4.88) guarantees a lower bound, determined by α_w, on h as $t \to \infty$. Note that (4.88) is simply the prior RCBF condition (4.8) enforced with equality instead of an inequality. Moreover, because Corollary 4.19 states that h_{dock} is an RCBF for any $\alpha \in \mathcal{K}_e$, the choice of α_w (and of the point $\alpha_w^{-1}(2)$) is a free parameter.

Remark 4.8. *Note that Lemma 4.21 only guarantees $x \to \partial_{\alpha_w^{-1}(2)}\mathcal{H}$ as $t \to \infty$. There exist finite and fixed time extensions of Lemma 4.21, provided the derivative α_w' of α_w satisfies $\alpha_w'(\lambda_c) = \infty$ where $\lambda_c = \alpha_w^{-1}(2)$, such as $\alpha_w(\lambda) = \frac{2}{\lambda_c^{1/3}}((\lambda - \lambda_c)^{1/3} + \lambda_c^{1/3})$. However, this choice of α_w is not locally Lipschitz continuous, as is often desired for other reasons, though this sort of design is a potential area for future study. For now, I instead employ the fact that for every $\epsilon > \alpha_w^{-1}(2)$, there exists a finite time T such that $x \to \partial_\epsilon\mathcal{H}$ as $t \to T$.*

Note that Lemma 4.21 applies to any RCBF, and thus to h_{dock} as well. The next step is then to use the terminal set $\partial_{\alpha_w^{-1}(2)}\mathcal{H}$ in Lemma 4.21 to generate the desired lower bounds on κ and $\dot{\kappa}$. First, I introduce one more metric as follows.

Definition 4.5 (Feasibility Margin)**.** *Let l_κ be the Lipschitz constant of κ in a neighborhood $\mathcal{Y} \subseteq \mathcal{X}$ of the set $\{(t, x) \in \mathcal{T} \times \mathcal{X} \mid h(t, x) = 0\}$. Given constants γ_1, γ_2, the* feasibility margin *is*

$$c_f(\gamma_1, \gamma_2) \triangleq \gamma_2 - (\gamma_1 + 2l_\kappa w_{x,max}). \tag{4.90}$$

The feasibility margin is important because the condition $\dot{\kappa} \in [\gamma_1, \gamma_2]$ is robustly guaranteed whenever $\dot{\kappa}_w \in [\gamma_2 - c_f, \gamma_2]$. Thus, I require that $[\gamma_2 - c_f, \gamma_2]$ be nonempty, or equivalently that $c_f \geq 0$; otherwise the problem of landing/docking with disturbances is not well-posed. To combined landing/docking with RCBFs, I then further require that $c_f > 0$ strictly, as in the following theorem.

Theorem 4.22. *Given constants $\gamma_2 > \gamma_1 > 0$ satisfying $c_f(\gamma_1, \gamma_2) > 0$ in* (4.90) *and a function $h_{\text{dock}} : \mathcal{T} \times \mathcal{X}$ of the form* (4.85) *with $\delta = \frac{1}{2}\gamma_2^2$, suppose the assumptions of Lemma 4.21 hold with h_{dock} and $\mathcal{H}_{\text{res,dock}}$ in place of h and $\mathcal{H}$. Suppose $\alpha_w \in \mathcal{K}$ in* (4.84) *satisfies*

$$\alpha_w^{-1}(2) = -\Phi^{-1}\left(\frac{1}{2}\gamma_2^2 + \Phi(0) - \frac{1}{2}\left(2l_\kappa w_{x,max} + \gamma_1\right)^2\right). \tag{4.91}$$

If the control input $u : \mathcal{T} \times \mathcal{X} \to \mathcal{U}$ satisfies (4.88) *for all $(t, x) \in \mathcal{H}_{\text{res,dock}}^{\mathcal{T}}$ and the initial condition satisfies $x(t_0) \in \mathcal{H}_{\text{res,dock}}(t_0)$, then there exists a finite $t_f \geq t_0$ such that $x(t_f)$ corresponds to landing at t_f.*

Proof. First, note that $c_f(\gamma_1, \gamma_2) > 0$ implies $\alpha_w^{-1}(2) > 0$ in (4.91) (if instead it held that $\alpha_w^{-1}(2) \leq 0$, then $\alpha_w \notin \mathcal{K}_e$, so the landing/docking problem would be poorly posed for the given disturbances).

Next, I note an important property of points inside the set $\partial_{\alpha_w^{-1}(2)}\mathcal{H}_{\text{res,dock}}$, visualized in gray in Fig. 4.15. Let $x_c \in \mathcal{X}$ be any point inside $\partial_{\alpha_w^{-1}(2)}\mathcal{H}_{\text{res,dock}}(t_c)$ for some $t_c \in \mathcal{T}$, and suppose that $\kappa(t_c, x_c) = 0$. Then at x_c, it holds that

$$\begin{aligned}
-\alpha_w^{-1}(2) &\leq h_{\text{dock}}(t_c, x_c) \\
&\overset{(4.85)}{=} \Phi^{-1}\left(\Phi\Big(\underbrace{h(t_c, x_c)}_{=0}\Big) - \frac{1}{2}\,\text{ssq}\left(\dot{\kappa}_w(t_c, x_c)\right) + \frac{1}{2}\gamma_2^2\right) \\
\Phi(-\alpha_w^{-1}(2)) &\geq \Phi(0) - \frac{1}{2}\dot{\kappa}_w(t_c, x_c)|\dot{\kappa}_w(t_c, x_c)| + \frac{1}{2}\gamma_2^2 \qquad (4.92) \\
\dot{\kappa}_w(t_c, x_c)|\dot{\kappa}_w(t_c, x_c)| &\geq 2\Phi(0) + \gamma_2^2 - 2\Phi(-\alpha_w^{-1}(2)) \\
&\overset{(4.91)}{=} \left(2l_\kappa w_{x,\max} + \gamma_1\right)^2 \qquad (4.93)
\end{aligned}$$

The right hand side of (4.93) is positive, so $\dot{\kappa}_w(t_c, x_c)$ will be positive as well. Specifically, $\dot{\kappa}_w(t_c, x_c) \geq 2l_\kappa w_{x,\max} + \gamma_1$. By definition, l_κ satisfies $l_\kappa \geq \|\nabla\kappa(t_c, x_c)\|$, and $\dot{\kappa}$ satisfies $\dot{\kappa}(t, x, w_x) \geq \dot{\kappa}_w(t, x) - 2\|\nabla\kappa(t, x)\|w_{x,\max}$. It follows that $\dot{\kappa}(t_c, x_c, w_x) \geq \dot{\kappa}_w(t_c, x_c) - 2l_\kappa w_{x,\max} \geq \gamma_1 > 0$. Visually, this means that $\dot{\kappa}_w(t_c, x_c)$ always lies on the intersection of the gray region and the magenta line in Fig 4.15, which implies $\dot{\kappa}(t_c, x_c, w_x)$ always lies on the magenta line (i.e. the docking states). Moreover, since Φ is monotone decreasing, $\Phi(\lambda) > \Phi(0)$ for all $\lambda < 0$. It follows from (4.92) that if $x \in \partial_{\alpha_w^{-1}(2)}\mathcal{H}_{\text{res,dock}}(t)$ and $\kappa(t, x) < 0$, then $\dot{\kappa}(t, x, w_x) > \gamma_1$, as shown by how the gray region in Fig. 4.15 occurs for larger $\dot{\kappa}_w$ (and therefore larger $\dot{\kappa}$) as κ decreases.

Safety with respect to γ_2 is already guaranteed by Theorem 4.20 since $\delta = \frac{1}{2}\gamma_2^2$, so I next use the above property and Lemma 4.21 to guarantee proximity, i.e. that there exists $t_f < \infty$ such that $\kappa(t_f, x(t_f)) = 0$. I divide this into two cases, depending on $x(t_0)$. The assumption $x(t_0) \in \mathcal{H}_{\text{res,dock}}(t_0)$ implies $\kappa(t_0, x(t_0)) \leq 0$, so going forward this proof assumes that $\kappa(t, x(t)) \leq 0$, as otherwise landing already occurred.

First, in the case that $x(t_0) \notin \partial_{\alpha_w^{-1}(2)}\mathcal{H}_{\text{res,dock}}(t_0)$, then by Lemma 4.21, h_{dock} is initially increasing and will keep increasing at least until the state reaches $\partial_{\alpha_w^{-1}(2)}\mathcal{H}_{\text{res,dock}}$. If the state reaches $\partial_{\alpha_w^{-1}(2)}\mathcal{H}_{\text{res,dock}}$ before landing, then see the second case. Otherwise, since $x(t)$ is converging to a set where $\kappa(t, x(t)) < 0 \implies \dot{\kappa}(t, x(t), w_x) > \gamma_1$, it follows from Remark 4.8 that for every $\gamma \in (0, \gamma_1)$, there exists a finite time $T(\gamma)$ such that the trajectory $x(t), t \geq T$ is sufficiently close to $\partial_{\alpha_w^{-1}(2)}\mathcal{H}_{\text{res,dock}}$ that $\kappa(t, x(t)) < 0 \implies \dot{\kappa}(t, x(t), w_x) > \gamma$. For example, in Fig. 4.15, all states on the top-most orange line (level set where $h_{\text{dock}} = -1.4\alpha_w^{-1}(2)$) satisfy $\dot{\kappa} > \gamma = 0.1$ as long as $\kappa < 0$. Since $\dot{\kappa}(t, x(t))$ becomes lower bounded by $\gamma > 0$ within finite time T, there must exist a

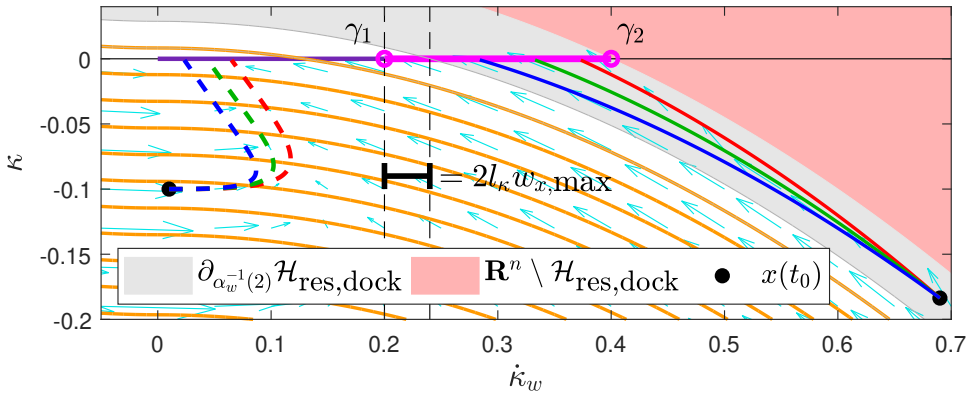

Figure 4.15: Phase Diagram of Landing and Docking Trajectories. A phase diagram of κ and $\dot{\kappa}_w$ when $w_x \equiv w_u \equiv 0$, along with trajectories from two initial states $x(t_0)$ inside and outside of $\partial_{\alpha_w^{-1}(2)}\mathcal{H}_{\text{res,dock}}$. From each initial condition, three trajectories are propagated in the presence of: disturbances that decrease $\dot{h}_{\text{dock}}$ (blue), no disturbance (green), and disturbances that increase $\dot{h}_{\text{dock}}$ (red). The magenta line represents docking states. The orange lines are level sets of h_{dock}. Trajectories converge asymptotically to the $h_{\text{dock}} \geq -\alpha_w^{-1}(2)$ superlevel set (gray region), and therefore converge in finite time to the lower superlevel sets and to the horizontal axis.

finite t_f at which $\kappa(t_f, x(t_f)) = 0$.

Second, in the case that $x(T) \in \partial_{\alpha_w^{-1}(2)}\mathcal{H}_{\text{res,dock}}(T)$ for any $T \in \mathcal{T}$ before landing occurs, then Lemma 4.21 implies that $x(t)$ will remain in $\partial_{\alpha_w^{-1}(2)}\mathcal{H}_{\text{res,dock}}(t)$ for all $t \geq T$ (as occurs for the trajectories plotted with solid lines in Fig. 4.15). It follows from (4.93) that $\dot{\kappa}(t, x(t), w_x) > \gamma_1$ for all $t \geq T$ as long as $\kappa(t, x(t)) < 0$. Thus, κ will keep increasing until t_f such that $\kappa(t_f, x(t_f)) = 0$, and at t_f it holds that $\dot{\kappa}(t_f, x(t_f)) \geq \gamma_1$. Thus, landing as in Definition 4.3 is guaranteed for finite t_f in both cases. ■

That is, I now have established a condition, given by (4.88) and (4.91), under which landing is guaranteed. Note that while landing occurs for $\dot{\kappa} \geq 0$, Theorem 4.22 requires one to encode the controller with a parameter γ_1 strictly greater than zero. Otherwise, landing is only guaranteed as $t \to \infty$. Also note that as the feasibility margin c_f becomes smaller, the value $\lambda_c = \alpha_w^{-1}(2)$ in (4.91) where $\alpha(\lambda_c) = 2$ becomes smaller, and thus the slope of α_w becomes larger. In practice, for digital controllers, following a steep α_w curve will require a faster controller update cycle. Next, I cover the docking case as follows.

Corollary 4.23. *Suppose the assumptions of Theorem 4.22. If furthermore* $x(t_0) \in \partial_{\alpha_w^{-1}(2)}\mathcal{H}_{res,dock}(t_0)$*, then there exists a finite* $t_f \geq t_0$ *such that* $x(t_f)$ *corresponds to docking at* t_f*.*

Proof. This result follows immediately from the proof of Theorem 4.22. Since $x(t_0) \in \partial_{\alpha_w^{-1}(2)}\mathcal{H}_{\text{res,dock}}(t_0)$, it follows that $x(t)$ remains in $\partial_{\alpha_w^{-1}(2)}\mathcal{H}_{\text{res,dock}}$ for the entire closed-loop tra-

jectory. By the argument in the final case of Theorem 4.22, it follows that $\dot{\kappa}(t, x(t)) \geq \gamma_1$ for all t until t_f such that $\kappa(t_f, x(t_f)) = 0$. Thus, docking as in Definition 4.4 occurs for finite t_f. ■

Thus, it is possible with this method to prescribe a minimum docking velocity γ_1 as well. Physically, the additional condition that $x(t_0) \in \partial_{\alpha_w^{-1}(2)} \mathcal{H}_{\text{res,dock}}(t_0)$ means that the chaser agent begins the maneuver with a sufficiently large velocity relative to the target. If this is not the case, then the chaser agent may not be able to accelerate to the required velocity before contacting the target, as occurs for the trajectories plotted with dashed lines in Fig. 4.15.

4.7.4 Spacecraft Landing and Docking Simulations

To verify the above conditions, I conducted two simulations. The first considered landing on an asteroid with nontrivial gravity and no atmosphere. The second considered docking in a low Earth orbit with multiple constraints. All simulation code can be found at https://github.com/jbreeden-um/phd-code/tree/main/2022/L-CSS%20Guaranteed%20Spacecraft%20Docking

The first problem sought for a spacecraft to land on the surface of the asteroid Ceres. Let $r, v \in \mathbb{R}^3$ be the position and velocity of the spacecraft with respect to the center of Ceres and $\mu = 6.26325(10)^{10}$ m^3/s^2 the gravitational parameter, so the dynamics are as in (4.63), repeated here

$$\dot{x} = \begin{bmatrix} \dot{r} \\ \dot{v} \end{bmatrix} = \begin{bmatrix} v \\ -\frac{\mu}{\|r\|^3} r \end{bmatrix} + \begin{bmatrix} 0 \\ u \end{bmatrix} + \begin{bmatrix} w_x \\ w_u \end{bmatrix}, \tag{4.94}$$

where $u \in \mathcal{U} \subset \mathbb{R}^3$ is the control input. Let $\mathcal{U} = \{u \in \mathbb{R}^3 \mid \|u\|_\infty \leq \bar{u}\}$ where $\bar{u} = 0.5$ m/s^2, $w_{u,\max} = 0.025$ m/s^2, and $w_{x,\max} = 0.01$ m/s. Let κ be the distance from the surface of Ceres, modelled as a perfect sphere, $\kappa = \rho - \|r\|$, where $\rho = 476000$ m. Note that condition (4.26) is satisfied for $\Phi(\lambda) = \frac{\mu}{\rho - \lambda} + (w_{u,\max} - \bar{u})\lambda$ as in (4.70). Let h_{dock} be as in (4.85), and choose $\gamma_1 = 0.1$ m/s and $\gamma_2 = 1.5$ m/s, so $\delta = 1.125$ in (4.85). This places the zero vector outside $\mathcal{H}_{\text{res,dock}}$, so κ and h_{dock} are continuously differentiable everywhere on $\mathcal{H}_{\text{res,dock}}$. For simplicity, choose $\alpha_w(\lambda) = k\lambda$, where condition (4.91) implies $k = 0.355$. I then appled the controller

$$u(t, x) = \nabla h_{\text{dock}}(t, x) g(t, x) \max_{b \in \mathbb{R}} b \quad \text{such that } (b \nabla h_{\text{dock}}(t, x) g(t, x)) \in \boldsymbol{\mu}_{\text{rcbf}}(t, x) \tag{4.95}$$

and simulated the spacecraft until landing occurred. In practice, u in (4.95) satisfies condition (4.88) at all points except where the RCBF condition (4.8) allows h_{dock} to increase at a rate that is unachievable within the input constraints (i.e. (4.88) is satisfied everywhere except where $\mathcal{U}$ is a strict subset of $\boldsymbol{\mu}_{\text{rcbf}}$). Since h_{dock} is constructed with Φ satisfying condition (4.26), the constraint on

the maximization in (4.95) will never require h_{dock} to decrease at a rate that is unachievable within the input constraints. Thus, the maximization in (4.95) is always feasible. For this simulation, I let w_u, w_x be random bounded disturbances. The resultant trajectory is shown in Fig. 4.16, the altitude above Ceres is shown in Fig. 4.17, and the control inputs are shown in Fig. 4.18. As expected, the control inputs always remained within the allowable set $\mathcal{U}$, and the spacecraft achieved landing in 3236 seconds with $\dot{\kappa}(t_f, x(t_f)) = 1.46$ m/s $< \gamma_2$.

The second problem sought for a chaser spacecraft to dock with a target spacecraft in a 400 km altitude circular Earth orbit. Suppose that the chaser coordinates relative to the target are $x_1, x_2 \in \mathbb{R}$ and follow the Hill-Clohessy-Wiltshire dynamics [8]

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \ddot{x}_1 \\ \ddot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 3n^2 & 0 & 0 & 2n \\ 0 & 0 & -2n & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ u_1 \\ u_2 \end{bmatrix} + \begin{bmatrix} w_{x,1} \\ w_{x,2} \\ w_{u,1} \\ w_{u,2} \end{bmatrix} \tag{4.96}$$

with $n = 0.00113$ rad/s. Let $\mathcal{U} = \{u \in \mathbb{R}^2 \mid \|u\|_\infty \leq \bar{u}\}$ where $\bar{u} = 0.082$ m/s^2, $w_{u,\max} = 0.002$ m/s^2, and $w_{x,\max} = 0.001$ m/s. For this scenario, suppose that the chaser is required to dock along a particular docking axis $\hat{a} = [0,\ 1]^{\mathrm{T}}$, starting from behind the target (i.e. $x_2(t_0) < 0$). First, let $\kappa = x_2$ encode the distance from the docking point, and let h_{dock} be a function of κ as in (4.85) using $\Phi(\lambda) = -\tilde{u}_1\lambda$ where $\tilde{u}_1 = 0.057$ m/s^2. Let $\gamma_1 = 0.07$ m/s and $\gamma_2 = 0.12$ m/s, so $\delta = 0.0072$. Again, let $\alpha_w(\lambda) = k_{\text{axis}}\lambda$, where condition (4.91) implies $k_{\text{axis}} = 25$. Next, to ensure convergence along the docking axis, define $\kappa_{\text{right}} = x_1 - \Delta$ and $\kappa_{\text{left}} = -x_1 - \Delta$, where the tolerance $\Delta = 0.03$ m. Then define the CBFs h_{right} and h_{left} as functions of κ_{right} and κ_{left}, respectively, as in the prior form of RCBF (4.83) using $\Phi(\lambda) = -\tilde{u}_0\lambda$ where $\tilde{u}_0 = 0.021$ m/s^2. For these two constraints, I used the RCBF h in (4.27),(4.83) discussed prior to this case study, instead of the relaxed RCBF h_{dock} in (4.85), as I want to ensure that the cross-track tolerance Δ is never violated. For these two RCBFs, I applied the class-$\mathcal{K}$ functions $\alpha_{\text{right}}(\lambda) = \alpha_{\text{left}}(\lambda) = k_{\text{ct}}\lambda$ with $k_{\text{ct}} = 200$. These three constraints are visualized in Fig. 4.14 and the complete two-spacecraft scenario is visualized in Fig. 4.19. Finally, I imposed a velocity constraint $h_{\text{vel}}(t, x) = \|[\dot{x}_1,\ \dot{x}_2]\|_\infty - v_{\max}$ with $v_{\max} = 10$ m/s and using $\alpha_{\text{vel}}(\lambda) = k_{\text{vel}}\lambda$ with $k_{\text{vel}} = 20$. Define

$$\begin{aligned} u_{\text{nom}}(t, x) = \big[&\alpha_w(-h_{\text{dock}}(\cdot))W(\cdot) - W(\cdot) - \partial_t h_{\text{dock}}(\cdot) \\ &- \nabla h_{\text{dock}}(\cdot) f(\cdot)\big] \nabla h_{\text{dock}}(\cdot) g(\cdot) / \|\nabla h_{\text{dock}}(\cdot) g(\cdot)\|^2 - k_p x_1 \end{aligned} \tag{4.97}$$

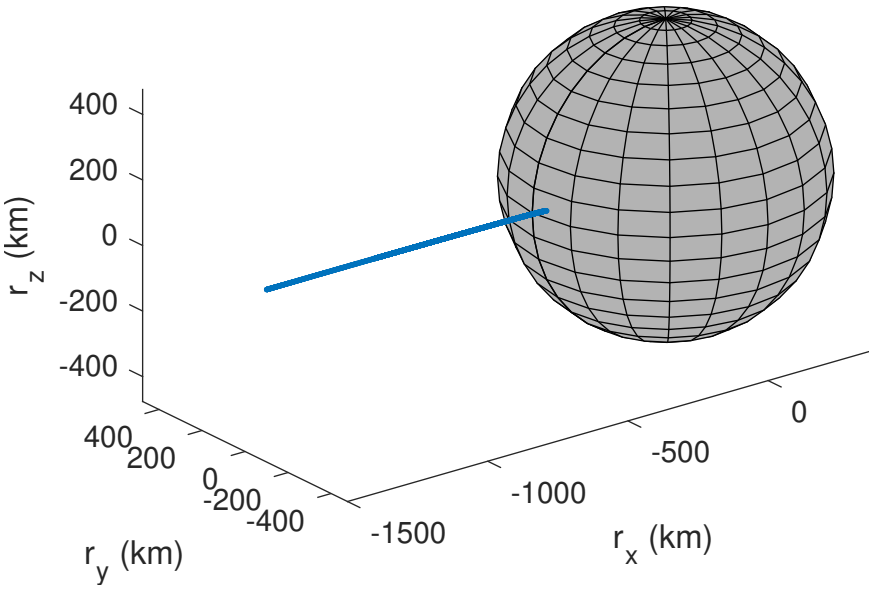

Figure 4.16: Ceres Landing Trajectory. Trajectory of the spacecraft as it lands on Ceres

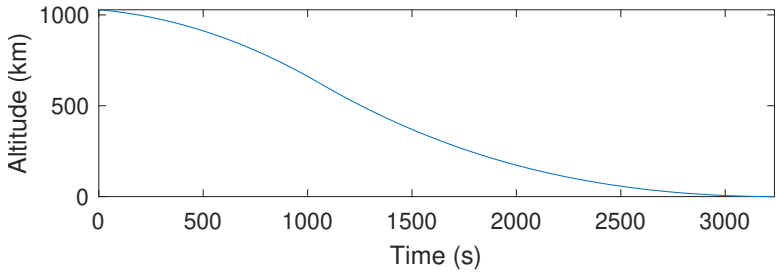

Figure 4.17: Ceres Landing Altitude versus Time. Altitude of the spacecraft as it lands on Ceres

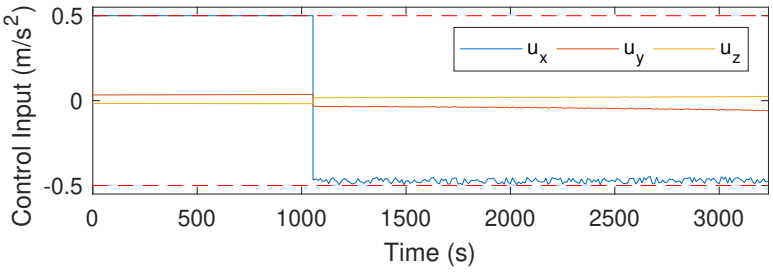

Figure 4.18: Ceres Landing Control Inputs. Control inputs of the spacecraft as it lands on Ceres

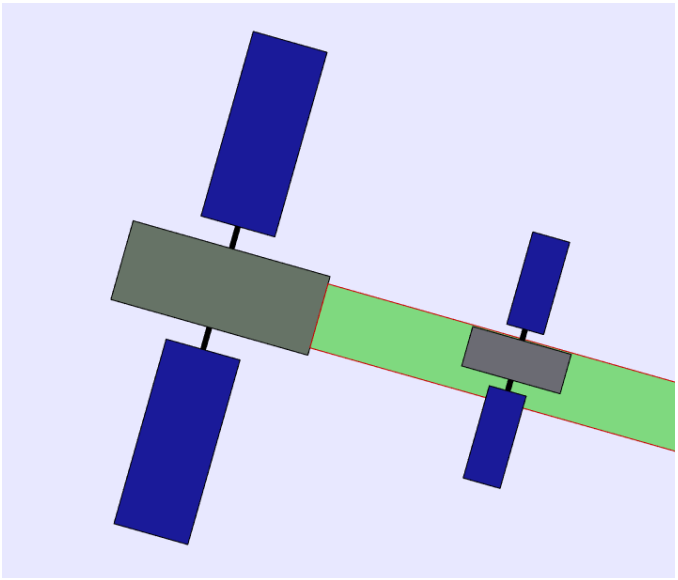

Figure 4.19: Satellite Docking Target and Chaser Visualization. Visualization of the docking cylinder (green) fixed to the target spacecraft (larger spacecraft) and a chaser spacecraft (smaller spacecraft) near the end of its docking maneuver

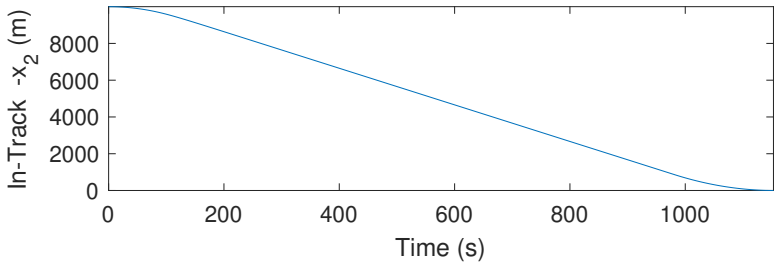

Figure 4.20: Satellite Docking In-Track Distance. Distance between spacecraft along docking axis

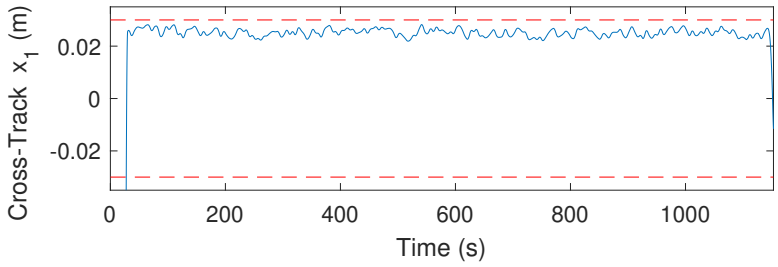

Figure 4.21: Satellite Docking Cross-Track Distance. Distance between spacecraft orthogonal to docking axis

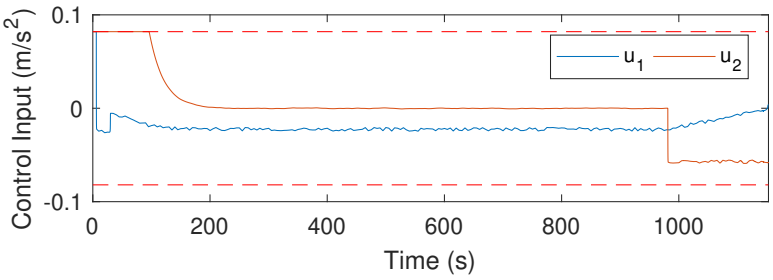

Figure 4.22: Satellite Docking Control Inputs. Control inputs of the chaser spacecraft

where $k_p = 0.1$. I then applied the controller

$$u(t,x) = \begin{cases} \arg\min_{u \in \boldsymbol{\mu}_{\text{rcbf,dock}} \cap \boldsymbol{\mu}_{\text{rcbf,right}} \cap \boldsymbol{\mu}_{\text{rcbf,vel}}} \|u - u_{\text{nom}}(t,x)\|^2 & h_{\text{left}}(t,x) > 0 \\ \arg\min_{u \in \boldsymbol{\mu}_{\text{rcbf,dock}} \cap \boldsymbol{\mu}_{\text{rcbf,right}} \cap \boldsymbol{\mu}_{\text{rcbf,left}} \cap \boldsymbol{\mu}_{\text{rcbf,vel}}} \|u - u_{\text{nom}}(t,x)\|^2 & h_{\text{left}}(t,x) \leq 0 \end{cases}.$$

Note that the controller in (4.98) is broken into two cases because most initial conditions start outside the (narrow) docking cylinder, so only one of the RCBFs $h_{\text{right}}, h_{\text{left}}$ will be initially non-positive. In this case, it held that $h_{\text{right}}(t_0, x(t_0)) \leq 0$. The term $-k_p x_1$ of u_{nom} works to drive the spacecraft close to the docking axis, and once it is sufficiently close, the control law (4.98) applies both the h_{right} and h_{left} RCBF conditions. Meanwhile, the first term of u_{nom} in (4.97) satisfies condition (4.88) to ensure docking occurs in finite time as in Theorem 4.22, and the quadratic program in (4.98) ensures safety and input constraint satisfaction. Note that $\tilde{u}_1$ and $\tilde{u}_0$ were chosen so that (4.70) is *strictly* satisfied for each Φ. This results in the QPs in (4.98) being *strictly* feasible and therefore locally Lipschitz continuous in x [199, Thm. 2.1].

A docking simulation with random bounded disturbances is shown in Figs. 4.20-4.21 and the control inputs are shown in Fig. 4.22. A video of the sequence is available at https://youtu.be/RoByiSD__jo. Docking occurred within the constraints in 1153 seconds with a terminal velocity of $\dot{\kappa}(t_f, x(t_f)) = 0.11$ m/s $\in [\gamma_1, \gamma_2]$. Note that x_1 in Fig. 4.21 converged quickly to $x_1 \in [-\Delta, \Delta]$, and then spent a lot of time near $x_1 = \Delta$. This is because the uncontrolled dynamics in (4.96) tend to cause x_1 to increase. The proportional control law $-k_p x_1$ in (4.97) does not adequately account for these dynamics, so instead the safety constraint $h_{0,r}$ ensures that $x_1 \leq \Delta$ for all time.

4.7.5 Conclusions on Docking and Other Tight Tolerance Specifications

This section has demonstrated how RCBFs introduce a margin on how close system trajectories can come to the boundary of the CBF set. In response to this, I then developed a method for tuning this margin and applied it to guaranteeing the finite-time execution of a landing and

docking maneuver with a terminal velocity inside a specified interval in the presence of bounded matched and unmatched disturbances. While the focus of this case study was spacecraft docking, the observations that

1. RCBFs limit the set of reachable states under certain disturbances, and

2. This shrinking of the CBF set can be tuned

are generally applicable to any system. In addition to demonstrating a solution to this specific problem, I hope that this section encourages the reader to think about problems like satellite docking as set-based specifications. In a conventional docking control design, one would typically design a controller for a particular nominal trajectory and then verify that under expected disturbance bounds, the set of all possible trajectories (or, more likely, a finite number of representative test trajectories) stays within the tolerances of the nominal trajectory. By contrast, the approach in this section did not require any nominal trajectory, and instead worked directly upon the set of trajectories defined as acceptable. This allowed me to prove a priori (without any test trajectories) that trajectories would stay within this set. This perspective, and the tuning of asymptotically convergent surfaces enabled by (4.61), are powerful control design tools.

CHAPTER 5

Sampled-Data Implementation of Control Barrier Functions

This chapter is concerned with implementing CBFs under sampled-data control laws. While the prior chapters considered entirely continuous-time systems, in practice, most modern control laws are implemented on computers, which can only sample the system state at some maximum frequency. Often, this sampling frequency is fast enough to well approximate continuous-time behavior. However, most spacecraft outside low Earth orbit require radiation hardened equipment, which substantially decreases processing speed. Whereas factory robots may recompute control inputs at 100 Hz or quicker, spacecraft are more likely to run between 1 Hz and 10 Hz update cycles. Between controller samples, the system continues to evolve in continuous time, but the controller cannot measure this evolution until the next sample, so the control input is fixed in a Zero-Order-Hold (ZOH) fashion. With such long delays between control input updates, it is important to consider these ZOH dynamics when implementing a safety-critical system.

This chapter is organized into three parts that address different aspects of the "sampled-data CBF problem". In Section 5.1, I present some preliminary work on 1) implementing ZOH controllers similar to (2.9), and 2) reducing the conservatism of these controllers. As I will show, the state of the art at the beginning of this research used extremely conservative approximations for the system evolution between controller updates, and these approximations were poorly suited to applications at 10 Hz or slower. Section 5.1 is intended as an overview of the "sampled-data CBF problem" and various solutions to it. Next, Section 5.2 presents the "robust sampled-data CBF problem" and the "robust high relative-degree sampled-data CBF problem" that builds upon the best-performing of the results in Section 5.1. Section 5.2 also considers a case study of constrained satellite attitude control in detail. Lastly, Section 5.3 presents some tools for the related problem of sampled-data control with impulsive actuators instead of ZOH actuators. While these impulsive actuators lead to a hybrid system model, the principal question is still whether the system is safe between control impulses, and thus the analysis is very similar to the analysis for ZOH actuators.

5.1 Various Methods to Ensure Set Invariance with Zero-Order-Hold Control Laws

5.1.1 Introduction to Sampled-Data CBFs

Quadratic Programs (QPs) utilizing Control Barrier Functions (CBFs) have been used for safety-critical control applications across disciplines, including vehicle control [29, 129], bipedal robots [30, 84], mechanical hands [119], and multi-agent systems [36]. Various authors have developed CBF-based set invariance conditions that apply to both continuous-time [29, 36, 84, 119] and discrete-time [30, 128, 129] systems. In practice, physical systems evolve in continuous time under controllers that are implemented in discrete time, such as Zero-Order-Hold (ZOH) controllers with fixed time-step. One can easily construct counter-examples showing that the control laws developed from the continuous-time CBF condition in [29, 36, 84] are no longer provably safe when the controller is executed in discrete steps, as visualized in Fig. 5.1. On the other hand, a controller implemented using only discrete-time CBFs may not satisfy the continuous safety condition between time steps [78], as illustrated by the red trajectory in Fig. 5.1.

Recently, [119] proposed a method for ensuring satisfaction of the continuous-time CBF condition using a ZOH control law by bounding the time derivative of the CBF between time steps. The method is extended in [124] to multi-agent systems in the presence of adversaries and uncertainty. The authors in [31] propose a similarly motivated approach, which also addresses uncertainty and input delay, using reachable set theory. In all of these papers, certain safe states might be cast unreachable, or excessive control inputs might be used to avoid unsafe regions.

This section studies several alternative conditions for forward invariance of safe sets under ZOH controllers. I begin by defining two types of margins, the *controller margin* and the *physical margin*, to compare the conservatism of the conditions developed. In Section 5.1.3.1, I then present extensions to the approaches in [31, 119, 124] that reduce conservatism as measured by these margins, while similarly relying on proving that the continuous-time CBF condition is always satisfied. Next, in Section 5.1.3.2, I instead approach the problem starting from discrete-time CBF conditions such as in [30, 195], and develop new sufficient conditions on the forward invariance of a CBF set under ZOH controllers. Finally, I present simulations using the existing and new conditions on an obstacle-avoidance problem for a unicycle agent, and on a spacecraft attitude-control problem. The simulations demonstrate how the reduced conservatism of the proposed approaches enables both the achievement of tight tolerance mission objectives (similar to how I tuned margins for disturbances in Section 4.7) and the ZOH application of CBFs under time-steps that were not possible using the method in [119, 124].

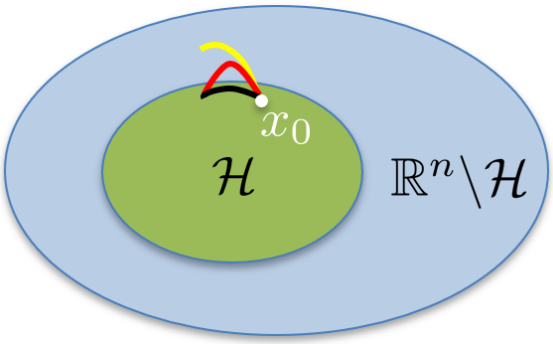

Figure 5.1: Illustration of Common CBF Sampling Outcomes. Illustration of CBF set and possible trajectories over a simple time-step $[0, \Delta t]$ from some point $x(0) = x_0$. If one simply applies the conventional CBF condition $\frac{\partial h(x)}{\partial x}(f(x) + g(x)u) \leq \alpha(-h(x))$ only at the controller sample times (instead of at all times continuously), then any of these three trajectories may occur. The black trajectory remains in the CBF set for the entire interval. The red trajectory is safe at the beginning and end of the interval, but becomes unsafe between samples (this also occurs with conditions such as those in [30, 128, 129]). The yellow trajectory exits the CBF set and does not return before the next sample.

5.1.2 Preliminaries

5.1.2.1 Notations

In addition to the notations in Section 2.1, let $B_r(x)$ denotes the closed ball centered at x of radius r. For a matrix $A \in \mathbb{R}^{n \times m}$, let $\|A\|$ denote the matrix-induced 2-norm of A. The function $\text{wrap}_\pi(\lambda)$ wraps λ to $[-\pi, \pi]$. For a given dynamical system, let $\mathcal{R}(x_0, T)$ denote the set of states reachable from some $x_0 \in \mathbb{R}^n$ in times $0 \leq t < T$. This section also makes use of the Lie-derivative notations explained in Section 3.2.1.

5.1.2.2 Model and Problem Formation

In this section, I return to considering time-invariant systems and CBFs, so consider the model

$$\dot{x} = f(x) + g(x)u\,, \tag{5.1}$$

with state $x \in \mathcal{X} \subseteq \mathbb{R}^n$, control input $u \in \mathcal{U} \subset \mathbb{R}^m$ where $\mathcal{U}$ is compact, and functions $f : \mathcal{X} \to \mathbb{R}^n$ and $g : \mathcal{X} \to \mathbb{R}^{n \times m}$. Assume that f and g are locally Lipschitz continuous. Define $u_{\max} \triangleq \max_{u \in \mathcal{U}} \|u\|$.

Let $h : \mathcal{X} \to \mathbb{R}$ be a continuously differentiable function and further assume that h has locally

Lipschitz continuous derivatives. Define the time-invariant set $\mathcal{H}$ as

$$\mathcal{H} \triangleq \{x \in \mathbb{R}^n \mid h(x) \leq 0\}. \tag{5.2}$$

In this section, I call h a "CBF", and $\mathcal{H}$ a "CBF set", but I note that this section does not make use of any formal definition of CBF. That is, h in this section is simply any function that defines a set $\mathcal{H}$ as in (5.2). For now, I call h a CBF for consistency of terminology, and in Section 5.2, I will present a formal definition of "ZOH-CBF" analogous to the Robust CBF in Definition 4.2.

For a continuous control law $u : \mathcal{X} \to \mathcal{U}$, the problem of rendering $\mathcal{H}$ forward invariant may be solved using any of Theorems 2.2, 2.6, 2.8. When applied to a QP as in (2.9), this leads to a condition of form

$$L_f h(x(t)) + L_g h(x(t))u(x(t)) \leq \alpha(-h(x(t)), \forall t \in \mathcal{T} \tag{5.3}$$

where $\alpha \in \mathcal{K}_e$ and $\mathcal{T} = [t_0, t_f]$ is the time domain of some trajectory. Without loss of generality, let $t_0 = 0$. Here, I write the CBF condition (5.3) in terms of time $t \in \mathcal{T}$ instead of states $x \in \mathcal{X}$ (as was done in (2.5) and (2.6)), because this will better align with the ZOH extensions of (5.3) in this section.

To apply the theorems in Section 2.3.2, one must ensure that (5.3) is satisfied along $x(t)$ for all $t \in \mathcal{T}$. However, suppose instead that the state x is only measured discretely (and thus $u(x)$ is updated discretely too) at times $t_k = kT, k \in \mathbb{Z}_{\geq 0}$ for a fixed time-step $T \in \mathbb{R}_{>0}$. Consider a ZOH control law satisfying

$$u_{\text{zoh}}(t) = u_k \triangleq u(x_k),\ \forall t \in [t_k, t_{k+1}), \tag{5.4}$$

where $u : \mathcal{X} \to \mathcal{U}$ and $x_k \triangleq x(t_k), \forall k \in \mathbb{Z}_{\geq 0}$. Note that uniqueness of the maximal closed-loop solution $x(t)$ (and hence of the sequence x_k) under such a control law u_{zoh} for a compact set $\mathcal{U}$ is guaranteed by [212, Thm. 54]. If one applies a control law similar to (2.9) with sampling as in (5.4), then (5.3) is only guaranteed to be satisfied at t_k, not necessarily on (t_k, t_{k+1}). This discrete satisfaction of (5.3) is not sufficient to guarantee forward invariance of $\mathcal{H}$. Thus, this section seeks a control-affine condition similar to (5.3) under which safety can be guaranteed when the control input is updated as in (5.4). This is summarized in the following problem statement:

Problem 5.1. *For a function $h : \mathcal{X} \to \mathbb{R}$, design a function $\phi : \mathbb{R}_{>0} \times \mathcal{X} \to \mathbb{R}$ such that any bounded, piecewise-constant control input u_{zoh} of the form* (5.4) *satisfying*

$$L_f h(x_k) + L_g h(x_k)u_k \leq \phi(T, x_k), \tag{5.5}$$

at the sampled states x_k for all $k \in \mathbb{Z}_{\geq 0}$ renders $\mathcal{H}$ forward invariant along the closed-loop

trajectories of (5.1).

I call (5.5) the ZOH-CBF condition. The following result, adapted from [119], provides one form of the function ϕ that solves Problem 5.1 (see also [124]).

Lemma 5.1 ([119, Thm. 2])**.** *Let the set $\mathcal{H}$ in* (5.2) *be compact and $\alpha \in \mathcal{K}$ be locally Lipschitz continuous. Let $l_{L_f h}, l_{L_g h}, l_{\alpha(h)}$ be the Lipschitz constants of $L_f h, L_g h, \alpha(-h)$, respectively. Then the function $\phi_0^g : \mathbb{R}_{>0} \times \mathbb{R}^n$, defined as*

$$\phi_0^g(T, x) \triangleq \alpha(-h(x)) - \frac{l_1 \Delta}{l_2} \left(e^{l_2 T} - 1 \right) , \tag{5.6}$$

solves Problem 5.1, where $l_1 = l_{L_f h} + l_{L_g h} u_{max} + l_{\alpha(h)}, l_2 = l_{L_f h} + l_{L_g h} u_{max}$, and $\Delta = \sup_{x \in \mathcal{H}, u \in \mathcal{U}} \| f(x) + g(x) u \|$.

Note that (5.3) and (5.5) are sufficient, not necessary, conditions for forward invariance [28, Rem. 12]. In practice, the form of the function ϕ_0^g in (5.6) is conservative in the sense that many safe trajectories with ZOH controllers may fail to satisfy (5.5) for $\phi = \phi_0^g$, as illustrated in simulation in Section 5.1.4. The work in the Section 5.1.3 is devoted to developing alternative solutions to Problem 5.1 that are less conservative compared to (5.6). I first introduce two metrics to quantify the conservatism of solutions to Problem 5.1 so as to better compare the following solutions.

5.1.2.3 ZOH Comparison Metrics

This section only considers functions ϕ of the form:

$$\phi(T, x) = \alpha(-h(x)) - \nu(T, x) , \tag{5.7}$$

where α is a class-$\mathcal{K}$ function that vanishes as $h(x) \to 0$, and $\nu : \mathbb{R}_{>0} \times \mathbb{R}^n \to \mathbb{R}$ is a function of the discretization time-step T and the state x that does not explicitly depend on h. This motivates the first metric of comparison, defined as follows.

Definition 5.1 (**Controller margin**)**.** *The function ν in* (5.7) *is called the* controller margin.

Note that ν is the difference between the right-hand sides of conditions (5.3) and (5.5), and is a bound on the discretization error that could occur between time steps. At a given state $x \in \mathcal{H}$, a larger controller margin will necessitate a larger control input to satisfy (5.5). A sufficiently large controller margin might also necessitate inadmissible control inputs, and thus make a CBF no longer applicable to a system. Thus, it is desired to design functions ϕ whose controller margins are small. For a given T, I say that a solution ϕ_a less conservative than ϕ_b if the controller margins of ϕ_a and ϕ_b satisfy $\nu_a(T, x) \leq \nu_b(T, x), \forall x \in \mathcal{H}$.

The controller margin is called *local* (denoted as $\nu^l(T,x)$) if ν varies with x, and *global* (denoted as $\nu^g(T)$) if ν is independent of x. The superscripts l and g, respectively, denote the corresponding cases, and ν is denoted with the same sub/superscripts as the corresponding ϕ function. For instance,

$$\nu_0^g(T) = \frac{l_1\Delta}{l_2}(e^{l_2 T} - 1) \tag{5.8}$$

is the controller margin of ϕ_0^g defined in (5.6), and is a global margin because it is independent of x.

Note that the continuous-time CBF condition (5.3) imposes that the time derivative of h vanishes as h approaches the boundary of the safe set. In contrast, the ZOH-CBF condition (5.5) causes the time derivative of h to vanish at a manifold in the interior of the safe set. Inspired from this, I define a second metric of comparison, which captures the maximum distance between this manifold and the boundary of the safe set.

Definition 5.2 (**Physical margin**). *For a solution ϕ of Problem 5.1 with the form* (5.7)*, the* physical margin *is the function $\delta : \mathbb{R}_{>0} \to \mathbb{R}$ defined as*

$$\delta(T) \triangleq \sup_{\{x \in \mathcal{H} \mid \phi(T,x)=0\}} -h(x)\,. \tag{5.9}$$

Intuitively, δ quantifies the effective shrinkage of the safe set due to the error introduced by discrete sampling. The condition (5.5) may exclude closed-loop trajectories from entering the set $\mathcal{H}^\delta \triangleq \{x \in \mathcal{X} \mid -\delta \leq h(x) \leq 0\}$, while the condition (5.3) does not (though condition (5.3) still limits the *rate* at which trajectories can approach this set, as discussed in Section 4.7). A smaller physical margin δ implies a smaller set $\mathcal{H}^\delta$ where system trajectories may not be allowed to enter.

Remark 5.1. *The physical margin δ depends on the choice of $\alpha \in \mathcal{K}$, but is always lower bounded. To capture this, define*

$$\delta^{\inf}(T) \triangleq \inf_{\alpha \in \mathcal{A}} \delta(T)\,, \tag{5.10}$$

where $\mathcal{A} \subseteq \mathcal{K}$ is the set of considered α (e.g. Lischitz continuous α in Lemma 5.1). Note that $\delta^{\inf}$ may be unachievable. For instance, the α which yields the physical margin-infimum *for ϕ_0^g is a linear function with an unbounded slope.*

The goal of Section 5.1.3 is to develop solutions to Problem 5.1 which have lower controller and/or physical margins than ϕ_0^g.

5.1.3 Three New Methods

This section presents three solutions to Problem 5.1, in both local and global forms. Section 5.1.3.1 presents solutions that follow from the continuous-time CBF condition (5.3), while

Section 5.1.3.2 presents conditions inspired by discrete-time CBF conditions as in [30, 195].

5.1.3.1 Extensions to Existing Literature

First, I note that in the proof of Lemma 5.1 in [119], the term $\frac{\Delta}{l_2}(e^{l_2 T}-1)$ serves as an upper bound on $\|x(t)-x_k\|, t \in [kT,(k+1)T)$. The bound is exponential, because x_k is treated as a solution to a dynamical system in [119]. Noting that x_k is a constant, the following lemma presents an alternative upper bound.

Lemma 5.2. *Let $\Delta = \sup_{x\in\mathcal{D},u\in\mathcal{U}} \|f(x)+g(x)u\|$ where $\mathcal{D}\subseteq\mathbb{R}^n$. Then for any $x_k = x(kT)\in\mathcal{D}$, the closed-loop trajectories of* (5.1) *satisfy $\|x(kT+\tau)-x_k\| \leq \tau\Delta$ for all $\tau \in \mathbb{R}_{\geq 0}$ such that $x(kT+\tau)\in\mathcal{D}$.*

Proof. The proof follows by inspection. ■

That is, the exponential upper bound in (5.6) can be immediately replaced with a linear bound. For very small time-steps T, this will make a minor difference, but the impact can be substantial for slower updating systems like spacecraft and/or systems with large l_2.

Second, I note that ν_0^g is a global margin. The ZOH-CBF condition (5.5) with ϕ of the form (5.7) can be made less conservative by using local margins instead of global margins. To this end, let $\mathcal{R}(x_k,T)$ denote the set of states reachable from some $x_k\in\mathcal{H}$ in times $t\in[kT,(k+1)T)$. I now present the first main result of this section.

Theorem 5.3. *Consider the set $\mathcal{H}$ defined in* (5.2) *and let $\alpha\in\mathcal{K}$ be locally Lipschitz continuous. Let $l_{L_fh}(x), l_{L_gh}(x), l_{\alpha(h)}(x)$ be the Lipschitz constants of $L_fh, L_gh, \alpha(-h)$ over the set $\mathcal{R}(x,T)$, respectively. Then the function $\phi_1^l : \mathbb{R}_{>0}\times\mathbb{R}^n$, defined as*

$$\phi_1^l(T,x) \triangleq \alpha(-h(x)) - \underbrace{l_1(x)T\Delta(x)}_{\nu_1^l(T,x)}, \tag{5.11}$$

solves Problem 5.1, where $l_1(x) = l_{L_fh}(x) + l_{L_gh}(x)u_{max} + l_{\alpha(h)}(x)$, and $\Delta(x) = \sup_{z\in\mathcal{R}(x,T),u\in\mathcal{U}}\|f(z)+g(z)u\|$.

Proof. For all $t\in[kT,(k+1)T)$, $k\in\mathbb{Z}_{\geq 0}$, it holds that

$$\begin{aligned} L_fh(x(t)) + L_gh(x(t))u_k &= L_fh(x_k) + L_gh(x_k)u_k - \alpha(-h(x_k)) \\ &\quad + [L_fh(x(t)) - L_fh(x_k) + (L_gh(x(t)) - L_gh(x_k))u_k \\ &\quad - (\alpha(-h(x(t))) - \alpha(-h(x_k)))] + \alpha(-h(x(t))) \\ &\leq L_fh(x_k) + L_gh(x_k)u_k - \alpha(-h(x_k)) + (l_{L_fh}(x_k)+ \end{aligned}$$

$$
\begin{aligned}
&\quad + l_{L_g h}(x_k) u_{\max} + l_{\alpha(h)}(x_k)) \|x - x_k\| + \alpha(-h(x(t))) \\
&\overset{(5.11)}{\leq} L_f h(x_k) + L_g h(x_k) u_k - \phi_1^l(T, x_k) + \alpha(-h(x(t))) \\
&\overset{(5.5)}{\leq} \alpha(-h(x(t)))\,.
\end{aligned}
$$

Thus, under (5.5) with $\phi = \phi_1^l$, it follows that $\dot{h}(x(t)) = L_f h(x(t)) + L_g h(x(t)) u_k \leq \alpha(-h(x(t))$ for all $t \in [kT, (k+1)T)$. Since this holds for all $k \in \mathbb{Z}_{\geq 0}$, it follows that $\dot{h}(x(t)) \leq \alpha(-h(x(t)))$ for all $t \in \mathcal{T}$. By assumption, f, g, and $\frac{\partial h}{\partial x}$ are locally Lipschitz continuous, and u is constant over $[t_k, t_{k+1})$, so all assumptions of Theorem 2.6 are satisfied and thus the set $\mathcal{H}$ is rendered forward invariant. Therefore, the function ϕ_1^l solves Problem 5.1. ■

Note that Theorem 5.3 requires knowledge of the local Lipschitz constants. If these constants are unavailable (e.g. due to computation constraints), it is still possible to improve upon Lemma 5.1 with the global margin function introduced in the following result.

Corollary 5.4. *Under the assumptions of Lemma 5.1, and with l_1, Δ as in Lemma 5.1, the function $\phi_1^g : \mathbb{R}_{>0} \times \mathbb{R}^n$, defined as*

$$
\phi_1^g(T, x) \triangleq \alpha(-h(x)) - \underbrace{l_1 T \Delta}_{\nu_1^g(T)}, \tag{5.12}
$$

solves Problem 5.1. Furthermore, for the same α, it holds that $\nu_1^l(T, x) \leq \nu_1^g(T) < \nu_0^g(T), \forall x \in \mathcal{H}, \forall T \in \mathbb{R}_{>0}$.

Proof. Observe that (5.11) reduces to (5.12) for $l_1 = \sup_{x \in \mathcal{H}} l_1(x)$ and $\Delta = \sup_{x \in \mathcal{H}} \Delta(x)$, so it holds that $\nu_1^l(T, x) \leq \nu_1^g(T), \forall x \in \mathcal{H}, \forall T \in \mathbb{R}_{>0}$ for the same α. It follows that $\phi_1^g(T, x) \leq \phi_1^l(T, x)$. Therefore, satisfaction of (5.5) with ϕ_1^g implies satisfaction of (5.5) with ϕ_1^l, and so by Theorem 5.3, ϕ_1^g also solves Problem 5.1.

From Taylor expansion, it holds that $T < \frac{1}{\lambda}(e^{\lambda T} - 1), \forall \lambda > 0$, so it follows that $\nu_1^g(T) < \nu_0^g(T), \forall T \in \mathbb{R}_{>0}$. ■

Thus, both ϕ_1^l and ϕ_1^g reduce conservatism compared to ϕ_0^g.

The physical margins of ϕ_1^l and ϕ_1^g are then $\delta_1^l(T) = \alpha^{-1}(\sup_{x \in \mathcal{H}, \phi_1^l(T,x)=0} l_1(x) T \Delta(x))$ and $\delta_1^g(T) = \alpha^{-1}(l_1 T \Delta)$, respectively. Since α is assumed locally Lipschitz continuous, there exists $\beta \in \mathbb{R}_{>0}$ and a neighborhood $\mathcal{N} \subseteq \mathbb{R}_{\geq 0}$ of the origin such that $\alpha(\lambda) \leq \beta\lambda, \forall \lambda \in \mathcal{N}$. It follows that $\alpha^{-1}(\lambda) \geq \frac{1}{\beta}\lambda, \forall \lambda \in \mathcal{N}$, so δ_1^l and δ_1^g vary linearly with T, as does δ_0^g.

To reduce conservatism further, I next define the following error term, inspired by [31], representing the difference between (5.3) evaluated at two points $x, z \in \mathcal{X}$ for a given input u:

$$
v(x, z, u) \triangleq L_f h(z) - L_f h(x) + (L_g h(z) - L_g h(x)) u - \alpha(-h(z)) + \alpha(-h(x))\,. \tag{5.13}
$$

The following result immediately follows.

Theorem 5.5. *Consider the set $\mathcal{H}$ defined in* (5.2) *and let $\alpha \in \mathcal{K}$. Then the function $\phi_2^l : \mathbb{R}_{>0} \times \mathbb{R}^n$ as follows solves Problem 5.1:*

$$\phi_2^l(T,x) \triangleq \alpha(-h(x)) - \underbrace{\sup_{z \in \mathcal{R}(x,T), u \in \mathcal{U}} \upsilon(x,z,u)}_{\nu_2^l(T,x)} . \tag{5.14}$$

The proof follows identical logic to the proof of Theorem 5.3. Note that α in Theorem 5.5 is no longer assumed to be locally Lipschitz continuous, as Theorem 5.5 no longer requires knowledge of the Lipzchitz constant of α and as this is not a condition of Theorem 2.6. Using the same approach relating ϕ_1^g and ϕ_1^l, I then define the function ϕ_2^g for which the following result can be easily shown.

Corollary 5.6. *Suppose that the conditions of Theorem 5.5 hold. Then the function $\phi_2^g : \mathbb{R}_{>0} \times \mathbb{R}^n$ as follows solves Problem 5.1:*

$$\phi_2^g(T,x) \triangleq \alpha(-h(x)) - \underbrace{\sup_{y \in \mathcal{H}, z \in \mathcal{R}(y,T), u \in \mathcal{U}} \upsilon(y,z,u)}_{\nu_2^g(T)} . \tag{5.15}$$

Remark 5.2. *Using $l_1(x), \Delta(x)$ as defined in Theorem 5.3, and for the same $\alpha \in \mathcal{K}$, it follows that $\upsilon(x,z,u) \leq l_1(x) T \Delta(x), \forall z \in \mathcal{R}(x,T), \forall u \in \mathcal{U}, \forall x \in \mathcal{H}$. Thus, for any $T \in \mathbb{R}_{>0}$, the controller margins satisfy $\nu_2^l(T,x) \leq \nu_1^l(T,x), \forall x \in \mathcal{H}$, and it follows that $\nu_2^g(T) \leq \nu_1^g(T)$.*

5.1.3.2 Alternative Method Based On Second Order Dynamics

The approaches discussed so far, as well as in [31, 119, 124], have relied on showing satisfaction of (5.3) to prove safety. In this section, rather that enforcing (5.3) between sample times, I instead start from a discrete-time CBF condition and apply it to an approximation of the continuous-time dynamics. One sufficient discrete-time CBF condition, as shown in [30], is

$$h(x_{k+1}) - h(x_k) \leq -\gamma h(x_k), \ \forall k \in \mathbb{Z}_{\geq 0} \tag{5.16}$$

for some $\gamma \in (0,1]$. In general, this condition is not control-affine. However, its linear approximation is control-affine and thus amenable to inclusion in a QP. The error of a linear approximation of a twice differentiable function is bounded by the function's second derivative. For brevity, define

$$\psi(x,u) \triangleq \frac{\partial \dot{h}(x,u)}{\partial x} \left(f(x) + g(x)u \right) , \tag{5.17}$$

which represents the second derivative of h between time steps. Since $f, g, \frac{\partial h}{\partial x}$ are assumed locally Lipschitz continuous, ψ is defined almost everywhere. Define the bound

$$\zeta(T,x) \triangleq \max\left\{\left(\sup_{z\in\mathcal{R}(x,T)\setminus\mathcal{Z},u\in\mathcal{U}}\psi(z,u)\right),0\right\}, \tag{5.18}$$

where $\mathcal{Z}$ is any set of Lebesgue measure zero (to account for CBFs that are not twice differentiable everywhere). I now state the first solution to Problem 5.1 in this chapter that does not rely on satisfying (5.3) along $x(t), \forall t \in \mathcal{T}$.

Theorem 5.7. *The function $\phi_3^l : \mathbb{R}_{>0} \times \mathbb{R}^n$, defined as*

$$\phi_3^l(T,x) \triangleq -\frac{\gamma}{T}h(x) - \underbrace{\frac{1}{2}T\zeta(T,x)}_{\nu_3^l(T,x)} \tag{5.19}$$

solves Problem 5.1, for any $\gamma \in (0,1]$.

Proof. For $k \in \mathbb{Z}_{\geq 0}$, let $t = kT + \tau, \tau \in [0,T]$. For any $x(t) \in \mathcal{R}(x_k,T)$, u_{zoh} as in (5.4) with $u_k \in \mathcal{U}$, the time derivative $\dot{h}$ satisfies

$$\begin{aligned}
\dot{h}(x(t),u_k) =&\ \dot{h}(x(t_k),u_k) + \int_{kT}^{kT+\tau}\ddot{h}(x(\sigma),u_k)d\sigma\\
=&\ \dot{h}(x(t_k),u_k) + \int_{kT}^{kT+\tau}\psi(x(\sigma),u_k)d\sigma\\
\overset{(5.18)}{\leq}&\ \dot{h}(x_k,u_k) + \int_{kT}^{kT+\tau}\zeta(T,x_k)d\sigma\\
=&\ \dot{h}(x_k,u_k) + \tau\zeta(T,x_k).
\end{aligned} \tag{5.20}$$

Similarly, h satisfies

$$\begin{aligned}
h(x(t)) =&\ h(x(t_k)) + \int_{kT}^{kT+\tau}\dot{h}(x(\sigma),u_k)d\sigma\\
\overset{(5.20)}{\leq}&\ h(x(t_k)) + \int_{kT}^{kT+\tau}\left(\dot{h}(x_k,u_k) + \tau\zeta(T,x_k)\right)d\sigma\\
=&\ h(x_k) + \dot{h}(x_k,u_k)\tau + \frac{1}{2}\tau^2\zeta(T,x_k)\\
\overset{(5.19)}{\leq}&\ h(x_k) + \phi_3^l(T,x)\tau + \frac{1}{2}\tau^2\zeta(T,x_k)\\
=&\ h(x_k) - \frac{\gamma\tau}{T}h(x_k) - \frac{\tau T}{2}\zeta(T,x_k) + \frac{\tau^2}{2}\zeta(T,x_k)\\
=&\ \left(1-\frac{\gamma\tau}{T}\right)h(x_k) + \frac{\tau}{2}\zeta(T,x_k)\left(\tau - T\right).
\end{aligned} \tag{5.21}$$

By definition in (5.18), $\zeta(T, x_k) \geq 0$. Suppose $h(x_k) \leq 0$. Then both terms of (5.21) are non-positive for any $\tau \in [0, T]$, so $h(x(t)) \leq 0, \forall t \in [kT, kT + T]$, and thus, $h(x_{k+1}) \leq 0$. Hence, given $x(0)$ such that $h(x(0)) \leq 0$ and applying (5.5) at every time step with $\phi = \phi_3^l$, it follows by induction that $h(x(t)) \leq 0, \forall t \in \mathcal{T}$, and thus $\mathcal{H}$ is forward invariant along the closed-loop trajectories of (5.1). Therefore, the function ϕ_3^l solves Problem 5.1. ■

Note that Theorem 5.7 did not invoke any of Theorems 2.2, 2.6, 2.8. One advantage of working with ZOH controllers (and discrete-time systems in general) is that many of the peculiarities of existence and uniqueness in continuous-time systems no longer matter.

Similar to the previous cases, I define the global version ϕ_3^g as follows.

Corollary 5.8. *Under the assumptions of Theorem 5.7, the function $\phi_3^g : \mathbb{R}_{>0} \times \mathbb{R}^n$ as follows solves Problem 5.1:*

$$\phi_3^g(T, x) \triangleq -\frac{\gamma}{T} h(x) - \underbrace{\frac{1}{2} T \sup_{z \in \mathcal{H}} \zeta(T, z)}_{\nu_3^g(T)} . \tag{5.22}$$

Next, I study how the solutions ϕ_3^l, ϕ_3^g compare to prior methods, by first comparing the controller margins as follows.

Theorem 5.9. *Under the assumptions of Theorem 5.7, the controller margins for ϕ_3^l, ϕ_3^g and ϕ_1^l, ϕ_1^g satisfy $\nu_3^l(T, x) \leq \frac{1}{2}\nu_1^l(T, x)$ and $\nu_3^g(T) \leq \frac{1}{2}\nu_1^g(T)$, $\forall x \in \mathcal{H}, \forall T \in \mathbb{R}_{>0}$.*

Proof. Since f and g are differentiable almost everywhere, their Lipschitz constants are the norms of their gradients. Thus,

$$\begin{aligned}
\nu_3^l(T, x) &= \frac{T}{2}\zeta(T, x) = \frac{T}{2} \max \left\{ \sup_{z \in \mathcal{R}(x,T) \setminus \mathcal{Z}, u \in \mathcal{U}} \psi(z, u), 0 \right\} \\
&= \frac{T}{2} \max \left\{ \sup_{z \in \mathcal{R}(x,T) \setminus \mathcal{Z}, u \in \mathcal{U}} \frac{\partial [L_f h(z) + L_g h(z) u]}{\partial z} \dot{z}, 0 \right\} \\
&\leq \frac{T}{2} \sup_{z \in \mathcal{R}(x,T) \setminus \mathcal{Z}, u \in \mathcal{U}} \left(\left\| \frac{\partial [L_f h(z)]}{\partial z} \right\| + \left\| \frac{\partial [L_g h(z)]}{\partial z} \right\| u_{\max} \right) \|\dot{z}\| \\
&\leq \frac{T}{2} \left(l_{L_f h}(x) + l_{L_g h}(x) u_{\max} \right) \Delta(x) \\
&= \frac{1}{2}\nu_1^l(T, x) - \frac{1}{2} l_{\alpha(h)}(x) T \Delta(x)
\end{aligned} \tag{5.23}$$

The inequality for the global margins follows immediately. ■

Thus, solutions ϕ_1^g, ϕ_2^g are provably less conservative than the existing solution ϕ_0^g, and ϕ_3^g is provably half as conservative as ϕ_1^g (and similarly for the local margins). It is difficult to analytically compare ϕ_2^l, ϕ_2^g with ϕ_3^l, ϕ_3^g, so I address this via simulations in Section 5.1.4.

Lastly, I consider the physical margins. Since $\alpha \in \mathcal{K}$ from (5.7) is specified as $\alpha(\lambda) = \frac{\gamma}{T}\lambda$ in (5.19),(5.22), the physical margin of ϕ_3^g is $\delta_3^g(T) = \frac{T}{\gamma}\nu_3^g(T) = \frac{T^2}{2\gamma}\sup_{x\in\mathcal{H}\setminus\mathcal{Z},u\in\mathcal{U}}\psi(x,u)$, and similarly $\delta_3^l(T) = \frac{T^2}{2\gamma}\sup_{x\in\mathcal{H}\setminus\mathcal{Z},\phi_3^l(T,x)=0,u\in\mathcal{U}}\psi(x,u)$. This implies δ_3^l, δ_3^g vary quadratically with T, while $\delta_0^g, \delta_1^l, \delta_1^g$ vary only linearly with T. Note that choosing $\alpha(\lambda) = \frac{\gamma}{T}\lambda$ does not similarly reduce $\delta_0^g, \delta_1^l, \delta_1^g, \delta_2^l, \delta_2^g$, because $l_{\alpha(h)}$ would increase inversely with T. Thus, reducing step size is far more effective at reducing physical margin when ϕ_3^l or ϕ_3^g is used.

5.1.4 Simulation Results and Comparison

I implemented the methods in Section 5.1.3 on two systems. First, I tested the unicycle system, described by

$$\dot{x}_1 = u_1\cos(x_3),\ \dot{x}_2 = u_1\sin(x_3),\ \dot{x}_3 = u_2, \tag{5.24}$$

where $[x_1,\ x_2]^{\mathrm{T}}$ is the position, x_3 is the orientation, and u_1,u_2 are the linear and angular velocity of the agent; the agent's task was to move around an obstacle at the origin using the CBF [34]

$$h = \rho - \sqrt{x_1^2 + x_2^2 - (\mathrm{wrap}_\pi(x_3 - \sigma\arctan 2(x_2, x_1)))^2}\,, \tag{5.25}$$

where ρ is the radius to be avoided, and σ is a shape parameter. Second, I tested a spacecraft pointing system (assuming unit moment of inertia in every direction), described by

$$\dot{p} = \omega \times p,\ \dot{\omega} = u, \tag{5.26}$$

where $p \in \mathbb{R}^3, \|p\| \equiv 1$, is a pointing vector, $\omega \in \mathbb{R}^3$ is the angular velocity, and $u \in \mathbb{R}^3$ is the angular acceleration. The system was tasked with reorienting an instrument while pointing away from an inertially-fixed vector using the CBF

$$h = s\cdot p - \cos(\theta) + \mu(s\cdot(\omega\times p))|s\cdot(\omega\times p)|\,, \tag{5.27}$$

where $s \in \mathbb{R}^3$, $\|s\| \equiv 1$, is a constant vector pointing to an object to be avoided, θ is the smallest allowable angle, and μ is a shape parameter. I also constrained $\|\omega\|_\infty \leq 0.2$ (this can be done easily by requiring $\omega_i + Tu_i \leq 0.2$ for each $i \in \{1,2,3\}$ in the QP), because otherwise the global controller margins are unbounded.

Both systems were tested for $T = 0.1$. For functions $\phi_0^g, \phi_1^l, \phi_1^g, \phi_2^l, \phi_2^g$, I used $\alpha(\lambda) = \lambda$, and for ϕ_3^l, ϕ_3^g, I used $\gamma = 1$. Notable parameters and the controller margins for the selected time-step for both systems are listed in Table 5.1. The physical margins for various time-steps are listed in Table 5.2. Note that $\delta_3^{g,\inf}$ is less than $\delta_0^{g,\inf}, \delta_1^{g,\inf}, \delta_2^{g,\inf}$, which means that ϕ_3^l and ϕ_3^g will allow the system trajectories to get closer to the boundary of the safe set than any of the other

Parameter	Unicycle	Spacecraft
Exclusion Zone	$\rho = 10$	$\theta = \pi/5$
Shape Parameter	$\sigma = 1$	$\mu = 100$
$\mathcal{U}$	$u_1 \in [0, 5]$ $u_2 \in [-0.25, 0.25]$	$\|u\|_\infty \leq 0.01$
$\nu_0^g(0.1)$	$1.316(10)^{50}$	14.20
$\nu_1^g(0.1)$	570.3	2.946
$\nu_2^g(0.1)$	0.6908	0.8815
$\nu_3^g(0.1)$	0.1319	0.1194

Table 5.1: Simulation parameters and global controller margins

	Unicycle			Spacecraft		
T	0.1	0.01	0.001	0.1	0.01	0.001
$\delta_0^{g,\text{inf}}$	$1.2(10)^{42}$	420	0.010	9.8	0.23	0.021
$\delta_1^{g,\text{inf}}$	0.54	0.054	0.0054	2.0	0.20	0.020
$\delta_2^{g,\text{inf}}$	0.53	0.053	0.0053	0.81	0.082	0.0082
$\delta_3^{g,\text{inf}}$	0.013	$1.3(10)^{-4}$	$1.3(10)^{-6}$	0.013	$1.3(10)^{-4}$	$1.3(10)^{-6}$

Table 5.2: Global physical margins for selected time-steps T

methods. Moreover, for the smaller values of T in Table 5.2, $\delta_3^{g,\text{inf}}$ varies quadratically with T, while $\delta_0^{g,\text{inf}}, \delta_1^{g,\text{inf}}, \delta_2^{g,\text{inf}}$ vary linearly with T. The agents used a controller of the form

$$u = \underset{u \in \mathcal{U}_{\text{zoh}}}{\arg\min} \|u - u_{\text{nom}}\| \tag{5.28}$$

where u_{nom} is a nominal control law that ignores the obstacle, and $\mathcal{U}_{\text{zoh}} \subseteq \mathcal{U}$ is the set of control inputs satisfying (5.5).

For the unicycle agent, the exact reachable sets $\mathcal{R}(x_k, T)$ were computed, and $\nu_1^l, \nu_2^l, \nu_3^l$ were computed using online maximizations of $l_1(x), \Delta(x), \upsilon(x, z, u), \psi(x, u)$ over these sets. For the spacecraft system (and in general for nonlinear systems), these reachable sets are harder to compute online, so I note that all preceding results still hold when $\mathcal{R}(x_k, T)$ is replaced with any superset of $\mathcal{R}(x_k, T)$ (though this in principle increases conservatism). Also, by Lemma 5.2, $\mathcal{R}(x_k, T) \subseteq B_{T\Delta}(x_k)$. To this end, given Lipschitz constants l_f, l_g for functions f, g, respectively, an upper bound for Δ is

$$\Delta_0(x_k) \triangleq \frac{\|f(x_k)\| + \|g(x_k)\| u_{\max}}{1 - (l_f + l_g u_{\max})T}, \tag{5.29}$$

assuming that the denominator of (5.29) is positive. Thus, the margins ν_1^l,ν_2^l,ν_3^l for the spacecraft were computed using online maximizations over the superset $B_{T\Delta_0(x_k)}(x_k)$. These maximizations

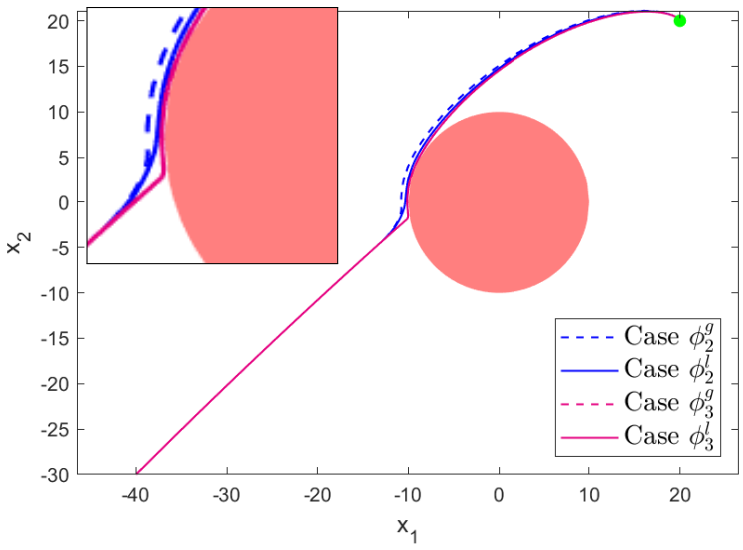

Figure 5.2: Unicycle Trajectories. The trajectories of the unicycle for 4 of the margin functions

took approximately 0.028, 0.026, and 0.018 seconds for $\nu_1^l, \nu_2^l, \nu_3^l$, respectively, for the unicycle, and 0.058, 0.071, and 0.045 seconds, respectively, for the spacecraft on a 3.5 GHz computer using MATLAB R2019b. For higher-dimensional systems, these online computations could limit the applications of the local methods. Each global margin took under a minute to compute. I then computed the states using the exact dynamics, and solved (5.28) using OSQP [196]. In total, 7 solutions (ϕ_0^g,ϕ_1^g,ϕ_1^l,ϕ_2^g,ϕ_2^l,ϕ_3^g,ϕ_3^l) to Problem 5.1 were tested. All simulation code for this section may be found at https://github.com/jbreeden-um/phd-code/tree/main/2021/L-CSS%20CBFs%20for%20Sampled%20Data%20Systems.

The trajectories for the two systems are plotted in Figs. 5.2-5.3, where the green markers are the target locations. As expected, certain methods took wider arcs around the obstacles than others based on the relative values of ν and δ. For the unicycle, only four methods are shown because using ϕ_0^g,ϕ_1^g,ϕ_1^l resulted in the agent turning away from the target. Similarly for the spacecraft, using ϕ_0^g,ϕ_1^g,ϕ_2^g eventually resulted in divergence from the target attitude as the QP was unable to satisfy (5.5).

The instantaneously required controller margins ν for every method, computed for $x(t)$ along the ϕ_3^l trajectories from Figs. 5.2-5.3, are plotted in Figs. 5.4-5.5. As predicted by Theorem 5.9, the green solid and dashed lines for controller margins ν_1^l, ν_1^g are always at least double (and generally an order of magnitude greater than) the equivalent pink lines for ν_3^l, ν_3^g, respectively. The controller margins ν_2^l, ν_2^g were also always larger than ν_3^l, ν_3^g, though this is not guaranteed by Theorem 5.9. Interestingly, for the unicycle, the global margin ν_3^g was generally similar to or smaller than the local margin ν_2^l, whereas for the spacecraft, ν_3^g was larger than both ν_1^l and ν_2^l. However, the trajectories corresponding to ϕ_3^g still approached closer to the obstacles than those under ϕ_1^l and ϕ_2^l

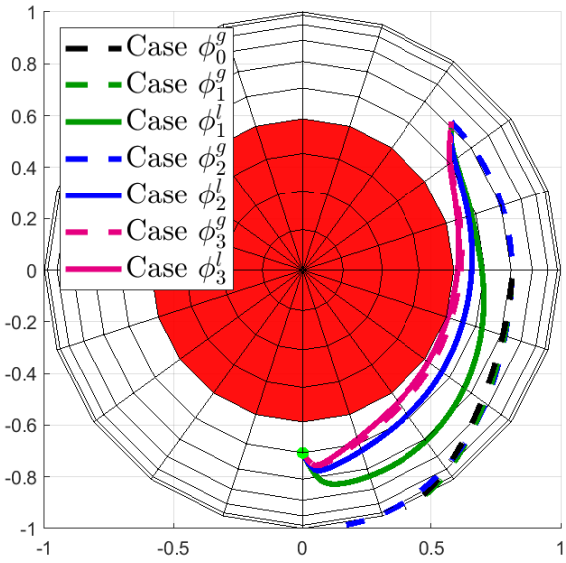

Figure 5.3: Satellite Orientation Trajectories. The trajectories of the spacecraft for all 7 margin functions

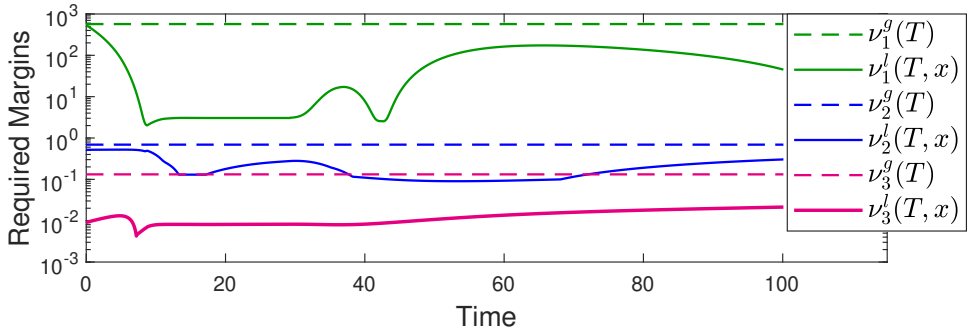

Figure 5.4: Unicycle Controller Margins. Controller margins for the unicycle system

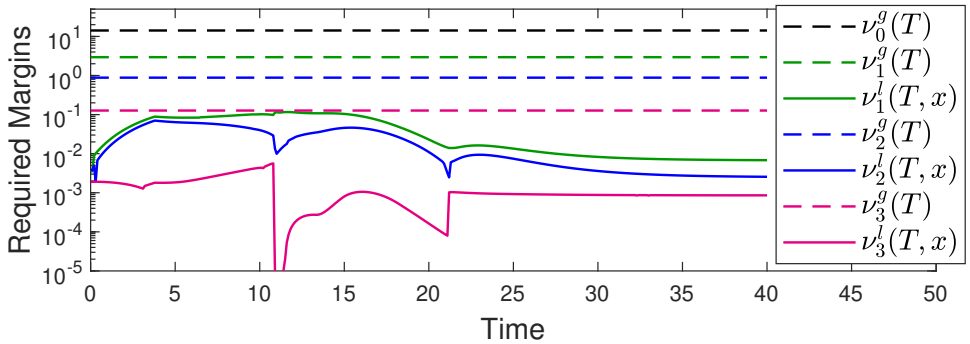

Figure 5.5: Satellite Orientation Controller Margins. Controller margins for the spacecraft system

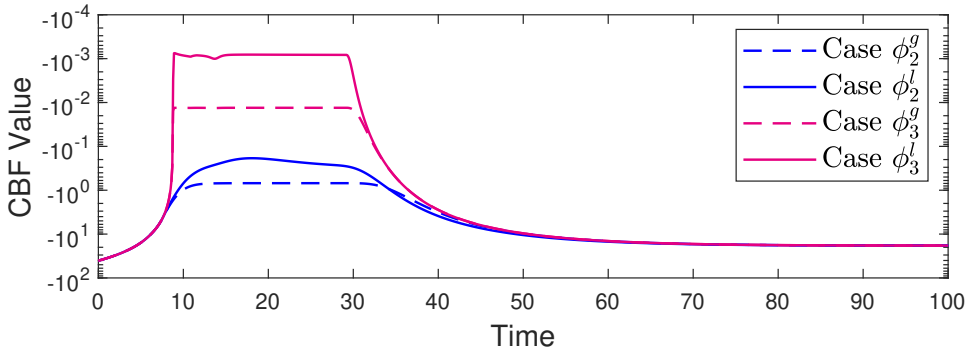

Figure 5.6: Unicycle CBF Values. CBF values along the 4 unicycle trajectories in Fig. 5.2

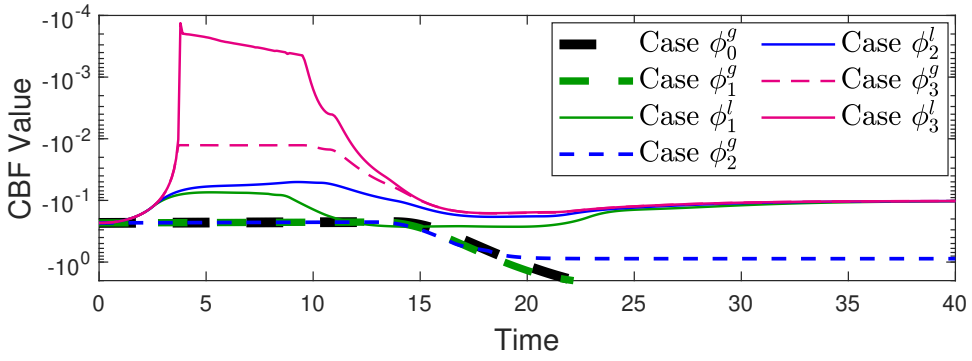

Figure 5.7: Satellite Orientation CBF Values. CBF values along the 7 spacecraft trajectories in Fig. 5.3

in both Figs. 5.2-5.3 because ϕ_3^g has an order of magnitude smaller physical margin.

Finally, the CBF values during every simulation are shown in Figs. 5.6-5.7. These plots show that the trajectories corresponding to ϕ_3^l and ϕ_3^g come within an order of magnitude closer to the boundary than those for any of the other methods. The dashed lines in Figs. 5.6-5.7 also agree with the theoretical physical margins listed in Table 5.2.

Noting these physical margins, I added a second constraint to the unicycle system that forced the unicycle to navigate through a narrow corridor only 0.3 units wide, shown in Fig. 5.8. The unicycle operating under ϕ_3^g or ϕ_3^l made it through the obstacles, while the best of the other methods (ϕ_2^l) could not, illustrating the importance of the physical margin.

5.1.5 Conclusions

This section has presented new conditions for ensuring safety in sampled-data systems that provably reduce conservatism compared to earlier results. I introduced two metrics for quantifying the margin in both the control input and in the effective shrinkage of the safe set. I then showed that the proposed conditions have smaller margins compared to those in earlier studies, and demon-

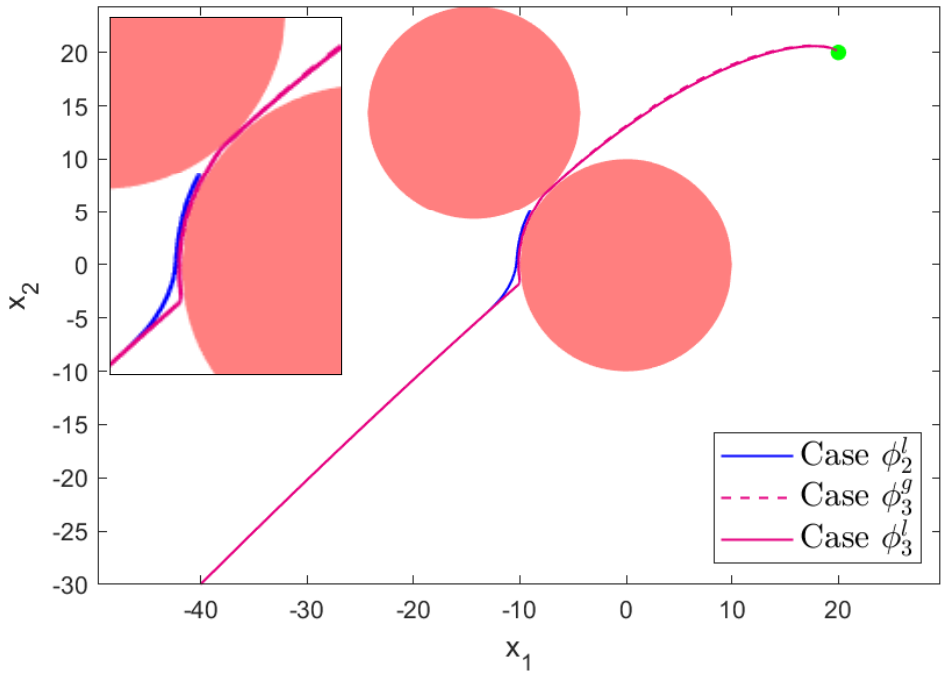

Figure 5.8: Unicycle Trajectories with Two Obstacles. A simulation with two tightly-spaced obstacles, in which controllers using margins ϕ_3^l and ϕ_3^g permit passage through the obstacles, while the other functions force the agent to stop.

strated the improved performance of the proposed results via numerical case studies. Conservatism in the numerical studies was reduced by at least as much as the theoretical results, and often by multiple orders of magnitude compared to the prior work. In particular, the physical margin of the last condition proposed varied quadratically with the discretization time-step, while that of the existing approaches varied linearly. This allowed completion of objectives that were not possible using other methods under the same time-step. Future work includes studying whether higher-order approximations can further decrease conservatism, as will be partially elaborated upon in the next section.

5.1.5.1 Remark About Class-$\mathcal{K}$ Functions

Theorem 5.9 only guarantees that the controller margin of ν_3^g will be at most half the controller margin ν_1^g, yet in both simulations it is actually many magnitudes lower (the difference in the local margins is not quite as stark, with the solid green and pink lines in Fig. 5.5 approaching a ratio of 2 to 1 around $t = 40$). The reason for this is largely due to the second term involving $l_{\alpha(h)}$ in (5.23). If one chooses $\gamma = 1$ in (5.16), then (5.16) reduces to simply $h(x_{k+1}) \leq 0$. That is, there is no class-$\mathcal{K}$ function in this expression. In continuous-time, the class-$\mathcal{K}$ function is important to ensure a continuous control signal (as discussed in Section 4.4) and the associated implications for

uniqueness of system solutions, but in discrete time, this function is unnecessary. The solutions ϕ_3^l and ϕ_3^g are the only solutions in this section that make use of this fact. Since these solutions do not include a class-$\mathcal{K}$ function (provided that $\gamma = 1$; one can still choose other $\gamma \in (0, 1]$ if one seeks to limit the system rate for other reasons), they also do not include a Lipschitz constant for the class-$\mathcal{K}$ function. The elimination of this term is the most significant source of improvement between ν_1^g and ν_3^g. For this reason, the following section will only discuss solutions that originate from the condition $h(x_{k+1}) \leq 0$ without any additional bounds on the rate of increase of h.

5.2 Robust Zero-Order-Hold for Spacecraft Attitude Control

Now that I have presented various solutions to the "sampled-data CBF problem", I now expand upon the best of these solutions (i.e. (5.19) with $\gamma = 1$) to create a more complete framework for CBFs under ZOH control laws and in the presence of disturbances. The ZOH controller margin discussed in the prior section acts as a margin for what is not known (or rather what would be too computationally expensive to predict) between time-steps, so adding an unknown disturbance to the model (5.1) as in (4.1) is a natural extension. This leads to the "robust sampled-data CBF problem", which is the main topic of Section 5.2.4.

Another issue highlighted in Section 5.1 is that the condition (5.16) may demand control inputs outside the allowable control set $\mathcal{U}$. This was observed particularly for ϕ_0^g under the satellite simulation in Section 5.1.4. However, while this error did not occur for ϕ_3^l and ϕ_3^g along the particular trajectory simulated, Example 5.1 will show that even these improved solutions are subject to the same deficiency. This is because (5.27) is an application of (4.17) to the high relative-degree constraint $\kappa = s \cdot p - \cos(\theta)$. Thus, Section 5.2.3 will also address the "robust high relative-degree sampled-data CBF problem". The solution to this problem is presented first, because the "robust sampled-data CBF problem" is essentially a special case of the high relative-degree version of this problem. At the end of both Section 5.2.3 and Section 5.2.4, I present formal definitions of the properties of a "ZOH CBF", analogous to Definition 2.14.

The following subsections contain a frightening amount of basic algebra and calculus. To ground all of this theory in a practical application, a case study for spacecraft attitude control (a more general version of (5.26)) is presented in several steps alongside the steps of the theory. The section concludes with extensive simulations surrounding this case study.

5.2.1 Introduction to Spacecraft Attitude Control

This section extends the theory of CBFs to solve the problem of constrained spacecraft attitude reorientation. At present, most spacecraft reorientations are accomplished either via shortest-path

maneuvers, which can be easily implemented onboard a spacecraft, or else are pre-planned by ground operators when more complex maneuvers are required. As the number of active spacecraft increases, there is potential for reducing operating costs in the latter case by increasing spacecraft autonomy, i.e. by computing maneuvers onboard without consulting ground operators. A common scenario in which shortest-past maneuvers are not allowable is when a spacecraft is not permitted to point sensitive instruments (body fixed vectors) at bright objects (inertially fixed vectors), or equivalently, when a spacecraft is required to keep an instrument pointed in a specified direction.

The problem of constrained reorientation has been studied extensively, using methods including path planners [221–230], model predictive controllers (MPC) [45, 231–234], sliding mode controllers (SMC) [235–237], reference governors [44], and barrier functions [26, 27, 238–240]. It has also been studied using CBFs combined with path planning in [47], along with cursory treatment using CBFs with controllers computed online in [83, 203, 241]. Compared to prior approaches, this work develops a method that provably guarantees both state constraint (i.e. instrument pointing requirements are obeyed) and input constraint (i.e. maximum allowable torques are not exceeded) satisfaction in the presence of bounded disturbances and under a sampled-data control law. The final control law is the output of a 4-dimensional quadratic program that is computationally lightweight. These guarantees are particularly useful when designing SmallSat attitude controllers, which often operate with infrequent ground contact, using undersized actuators (i.e. tight input constraints), at low altitudes (i.e. large disturbances), at low control sampling frequencies, and with limited computational capabilities.

To employ standard CBF terminology, I refer to the set of states with allowable separations between all instruments and all bright objects, and with allowable angular rates, as the *safe set*, which I assume to be nonempty at all times. The central problem is that of rendering trajectories always inside the safe set from some *viable set* (see Definition 2.8) of initial conditions where this problem is well-posed.

Early work on constrained reorientation in [26] developed a Lyapunov function for safe reorientation in terms of Euler angles, though this Lyapunov function may be nonconvex. The authors in [221] noted that this same constraint could be expressed as a convex set of quaternions, and [27, 238] developed a strictly convex Lyapunov function in terms of quaternions. The work in [236, 237] added an angular velocity constraint and actuator-allocation algorithm to the same technique. The work in [240] expanded the technique to Modified Rodrigues Parameters, and proposed a method for ensuring input constraint satisfaction. Note that while these Lyapunov functions resulted in simple control laws that could be implemented online, none of these approaches consider controller sampling, and these controllers can result in slow trajectories, as I will show in simulation in Section 5.2.5.

An early path planning technique utilized a variant of Rapidly Exploring Random Trees to find

safe paths in $SO(3)$ space [230]. Later, path planning techniques using direct optimization along with the quaternion constraint identified in [221] were developed in [221, 222, 226] and combined with translational planning in [225], though these methods are potentially too computationally intensive to implement online on a spacecraft processor. Related work in [223, 224, 228] discretized the safe set to a finite set of nodes and used graph search techniques to plan paths between the nodes. The maneuvers resulting from these techniques are safe but possibly inefficient due to the discretization. The planners in [227, 229] add additional refinements to improve efficiency, whereas the controller proposed in [47] executes a faster transition between the path nodes and uses CBFs to keep the trajectory within a safe region around the pre-planned path. By comparison, the approach employed in this work and in [27, 238] only keeps the state away from unsafe states rather than in a neighborhood of a precomputed safe path as in [47].

MPC approaches to constrained reorientation, such as [45] and its extensions in [231, 232], are generally special applications of path planning techniques. Similarly, the SMC approach in [235–237] and the approximate optimal control via reinforcement learning in [239] are special applications of the barrier functions used in [27, 238]. While MPC and optimal control can provide safety guarantees, in this work I seek a method that is less computationally intensive. The reference governor approach in [44] is notable because it developed an explicit control law without path planning that is guaranteed to satisfy input constraints. However, few of the aforementioned approaches explicitly consider disturbances, whereas there is extensive CBF literature on disturbance rejection [117, 118], and Chapter 4 developed multiple results on simultaneous disturbance rejection and input constraint satisfaction. Finally, spacecraft often operate with digital controllers with slow update cycles. Path planners and MPC can account for controller sampling given sufficiently sophisticated models, while most Lyapunov methods cannot. On the other hand, margins for controller sampling have also been considered in prior CBF literature such as [119] and Section 5.1, which this section will now extend to also account for relative-degree 2 state constraints, input constraints, and disturbance rejection.

Control Barrier Functions are a Lyapunov-like method for determining safe control inputs, i.e. control inputs that generate trajectories that provably satisfy the state constraints. For an overview of CBFs, see [66] or Chapter 2. Following this methodology, I assume that each requirement that the system trajectories must satisfy is expressed as the state belonging to a given *constraint set* (e.g. the set of states such that a particular instrument is sufficiently far away from a particular bright object). The safe set is then the intersection of all constraint sets (see also [136, 140] and Chaoter 6). For each constraint set, I then construct a corresponding CBF (e.g. as in [85, 86] and Chapters 3-4) and associated zero-sublevel set, herein called a *CBF set*. Each CBF then provides a pointwise condition on the control input that is sufficient to ensure that state trajectories always belong to the corresponding CBF set. Multiple CBFs and CBF sets may then be combined to establish forward

invariance of a subset of the safe set (see Chapter 6 for more considerations regarding combining CBFs). Application of CBFs to attitude control was first suggested in [83], and in fact, it would be simple to express the quaternion constraint developed in [221] as a CBF. However, such a CBF would suffer from the same challenges with input constraints, disturbances, and controller sampling as the related Lyapunov approaches in [27, 236–238]. These challenges are amplified when some of the constraint functions are of relative-degree 2 with respect to the system dynamics, as is the case for spacecraft pointing constraints. That said, extensions of [66] in the CBF literature provide several general tools for addressing these challenges (see [66, 87, 117, 118] and the remainder of this dissertation), as well as other potentially relevant phenomena not presently considered. This dissertation has already individually addressed input constraint satisfaction, robustness to disturbances, and zero-order-hold (ZOH) controller sampling with CBFs, and this section will now incorporate and extend all of these preceding results. In particular, I will show in Example 5.1 that the ZOH discretization method in Section 5.1 is not immediately compatible with the input constraint work in Chapters 3-4 and [87], so the bulk of Section 5.2.3 is devoted to reconciling these two approaches while minimizing conservatism. I then apply all the CBF conditions herein derived together online using an m-dimensional quadratic program (QP), where m is the number of control inputs and is generally far smaller than the dimension of the optimizations in planning or MPC approaches.

The rest of this section is organized into both 1) a general method accomplishing the above foci for arbitrary systems and constraints, and 2) a case study that applies this method to the constrained reorientation problem. The case study is presented in parallel as each step of the theory is developed for numerical motivation. Section 5.2.2 presents the formulation of the general problem, and of the specific system and constraints used in the case study. Section 5.2.3 presents the main result combining ZOH control inputs with input constraints and disturbances for relative-degree 2 constraints (e.g. pointing constraints), while Section 5.2.4 presents a related result for relative-degree 1 constraints (e.g. angular rate constraints). Section 5.2.5 presents the real-time QP controller and simulations both in MATLAB and in a NASA-developed attitude control simulator. Section 5.2.6 presents concluding remarks.

5.2.2 Preliminaries for General Problem and Case Study

5.2.2.1 Model

Let $q \in \mathbb{Q} \subseteq \mathbb{R}^{n_1}$ be the coordinates and $v \in \mathbb{V} \subseteq \mathbb{R}^{n_2}$ the velocities of a second-order system of the form

$$\dot{q} = f_1(t, q, v) \tag{5.30a}$$

$$\dot{v} = f_2(t, q, v) + g_1(t, q, v)u + g_2(t, q, v)\xi \tag{5.30b}$$

with time $t \in \mathcal{T} \subseteq \mathbb{R}$, state $x \triangleq (q, v) \in \mathcal{X} \triangleq \mathbb{Q} \times \mathbb{V} \subseteq \mathbb{R}^{n_1+n_2}$, control $u \in \mathcal{U} \subset \mathbb{R}^m$ where $\mathcal{U}$ is compact, and disturbance $\xi \in \Xi \subset \mathbb{R}^p$ where Ξ is bounded. Assume function f_1 is twice continuously differentiable in all arguments, functions f_2, g_1, g_2 are continuously differentiable in all arguments, and that $f_1, f_2, g_1, g_2, u, \xi$ are sufficiently regular so as to admit unique system trajectories for the entire time domain $\mathcal{T}$. The results of this section hold for general f_1, f_2, g_1, g_2, but for the purposes of the attitude control case study, suppose the following specific system.

Case Study Part i (System Definition)**.** *Assume a single rigid-body spacecraft. Let F_N be an inertial frame and F_B a spacecraft-fixed frame. For this case study, let $\mathbb{Q} = \{q \in \mathbb{R}^4 \mid \|q\| = 1\}$ be the quaternion space and let $q = [q_0,\, q_1,\, q_2,\, q_3]^{\mathrm{T}} \in \mathbb{Q}$ be the quaternion (with scalar element q_0 first) that rotates from F_N to F_B. Let $\omega \in \mathbb{R}^3$ be the angular velocity of F_B with respect to F_N expressed in frame F_B. Suppose the spacecraft has m reaction wheels. Let $a_i, i = 1, \cdots, m$, $a_i \in \mathbb{R}^3$, $\|a_i\| = 1$ denote the spin axes of the wheels in frame F_B, and define $A \in \mathbb{R}^{3\times m}$ as $A \triangleq [a_1,\, \cdots,\, a_m]$. Let $w_i, i = 1, \cdots, m$, $w_i \in \mathbb{R}$ denote the angular velocity of the wheels with respect to F_B, and define $w \in \mathbb{R}^m$ as $w = [w_1,\, \cdots,\, w_m]^{\mathrm{T}}$. The system velocities as in* (5.30b) *are $v = (\omega, w) \in \mathbb{V} = \mathbb{R}^{3+m}$. Assume each wheel is axially symmetric and let $J_{w,i} \in \mathbb{R}_{>0}$ be the axial moment of inertia of the ith wheel, and let $J_w \in \mathbb{R}^{m\times m}$ be a diagonal matrix whose ith row and column element is $J_{w,i}$. Let J_b be the moment of inertia of the spacecraft without wheels plus the transverse moments of inertia of the wheels (e.g. see [14, Eq. 3.140, Ch. 3.3.5.1]) expressed in frame F_B, and let $J_{tot} \triangleq J_b + \sum_{i=1}^m J_{w,i}(a_i a_i^{\mathrm{T}})$ denote the total moment of inertia of the spacecraft. Assume that J_b and J_w are constant. The spacecraft state is then $x = (q, \omega, w) \in \mathcal{X} = \mathbb{Q} \times \mathbb{R}^{3+m}$ and the dynamics [242] are*

$$\dot{q} = \frac{1}{2}\begin{bmatrix} 0 & -\omega_1 & -\omega_2 & -\omega_3 \\ \omega_1 & 0 & \omega_3 & -\omega_2 \\ \omega_2 & -\omega_3 & 0 & \omega_1 \\ \omega_3 & \omega_2 & -\omega_1 & 0 \end{bmatrix} q \tag{5.31a}$$

$$\dot{v} = \begin{bmatrix} \dot{\omega} \\ \dot{w} \end{bmatrix} = \underbrace{\begin{bmatrix} J_{tot} & AJ_w \\ J_w A^{\mathrm{T}} & J_w \end{bmatrix}^{-1}}_{\triangleq Z} \begin{bmatrix} -\omega \times (J_{tot}\omega + AJ_w w) + \xi \\ u \end{bmatrix} \tag{5.31b}$$

where $u \in \mathcal{U} \subset \mathbb{R}^m$ is the commanded wheel torque. The maximum wheel torque is limited to $u_{\max}$, so $\mathcal{U} = \{u \in \mathbb{R}^m \mid \|u\|_\infty \leq u_{\max}\}$. For this particular case study, I suppose a 6U CubeSat with parameters given in Table 5.3 and visualized in Fig. 5.9. Note that I have chosen

Parameter	Value
m	4
A	$\begin{bmatrix} 0 & 0 & 0.8165 & -0.8165 \\ 0 & -0.9428 & 0.4714 & 0.4714 \\ -1 & 0.3333 & 0.3333 & 0.3333 \end{bmatrix}$
$J_{w,i}$	$1.722(10)^{-5}$ kg-m^2, $i = 1, 2, 3, 4$
$u_{\max,i}$	$7(10)^{-4}$ N-m, $i = 1, 2, 3, 4$
J_b	$\begin{bmatrix} 0.1672 & 0 & 0 \\ 0 & 0.1259 & 0 \\ 0 & 0 & 0.06121 \end{bmatrix}$ kg-m^2
P	$\begin{bmatrix} 0.1672 & P_{12} & P_{13} \\ P_{21} & 0.1259 & P_{23} \\ P_{31} & P_{32} & 06121 \end{bmatrix}$ kg-m^2 where $\lvert P_{ij}\rvert < 10^{-20}$ for $i \neq j$
$e_{\max}$	$5.092(10)^{-5}$ kg-m^2/s^2
$w_{\max}$	628.3 rad/s
$\xi_{\max}$	$1.00(10)^{-5}$ Nm

Table 5.3: Physical Parameters of the Spacecraft

a configuration with 4 wheels in Table 5.3 rather than a more typical 3 wheel configuration in order to demonstrate the general applicability of these results. The wheel moments of inertia and maximum torques in Table 5.3 are based off a commercially available wheel package[a], with the maximum per-wheel torque reduced to be comparable to a 3 wheel configuration. Let $\Xi = \{\xi \in \mathbb{R}^3 \mid \|\xi\| \leq \xi_{\max}\}$ *for* $\xi_{\max}$ *in Table 5.3, which comes from approximate values of aerodynamic drag on a 6U CubeSat at 500 km altitude.*

[a]CubeWheel Medium: www.cubespace.co.za/products/adcs-components/cubewheel/#cubewheel-specifications

5.2.2.2 Safety Constraints

Next, suppose that the trajectories of (5.30) are required to lie in the intersection of several *constraint sets*, each defined by the zero sublevel set of some *constraint function* as in Chapters 3-4. Let $\kappa_i : \mathcal{T} \times \mathbb{Q} \to \mathbb{R}$ for $i = 1, \cdots, N_1$ denote the relative-degree 2 constraint functions, and let $\eta_i : \mathcal{T} \times \mathbb{V} \to \mathbb{R}$ for $i = N_1 + 1, \cdots, N_1 + N_2$ denote the relative-degree 1 constraint functions. The constraint sets are

$$\mathcal{Q}_i(t) \triangleq \{x = (q, v) \in \mathcal{X} \mid \kappa_i(t, q) \leq 0\} \tag{5.32a}$$

$$\mathcal{V}_i(t) \triangleq \{x = (q, v) \in \mathcal{X} \mid \eta_i(t, v) \leq 0\} \tag{5.32b}$$

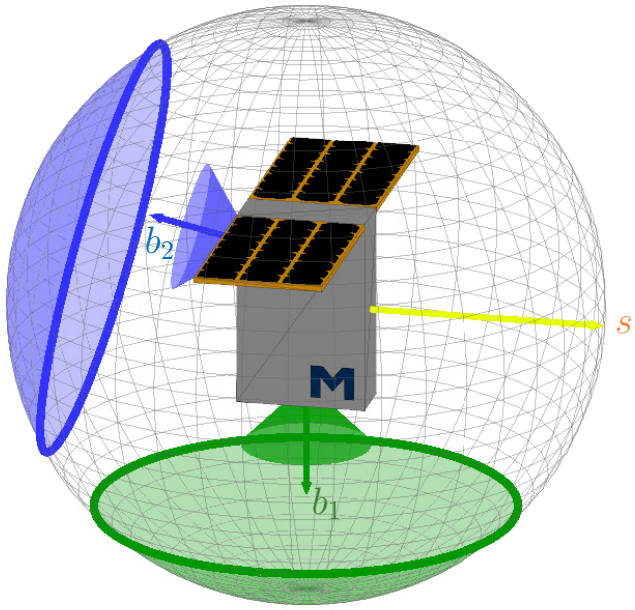

Figure 5.9: Visualization of the Spacecraft. A 6U CubeSat with two spacecraft-fixed keep-out zones centered about b_1, b_2, and an inertially-fixed vector s that must be kept outside these zones

and the resultant *safe set* is

$$\mathcal{S}(t) \triangleq \left(\bigcap_{i=1}^{N_1} \mathcal{Q}_i(t) \right) \cap \left(\bigcap_{i=N_1+1}^{N_1+N_2} \mathcal{V}_i(t) \right) \tag{5.33}$$

where $\mathcal{Q}_i$, $\mathcal{V}_i$, and $\mathcal{S}$ are permitted to be time-varying. As an abuse of notation, I will generally write $\kappa_i(t,x)$ and $\eta_i(t,x)$ in place of $\kappa_i(t,q)$ and $\eta_i(t,v)$ in order to match the CBF notation for $h(t,x)$ in Section 5.2.2.3. Some constraint functions that are common in attitude control are as follows; these constraints are also the basis of the simulations in Section 5.2.5.

Case Study Part ii (Constraints)**.** *For the spacecraft system in* (5.31)*, let* $b \in \mathbb{R}^3$*,* $\|b\| = 1$*, be a body-fixed vector, such as an instrument boresight vector (e.g. the green or blue vectors in Fig. 5.9). Let* $s(t)$*,* $\|s(t)\| = 1$*, be a vector, potentially time-varying (provided* s *is thrice continuously differentiable). Suppose that one requires that the angle between* $s(t)$ *and* b *is always at least* θ *(e.g. the local sun vector, represented by the yellow vector in Fig. 5.9). This leads to a constraint function of the form*

$$\kappa_b(t,q) = s(t)^{\mathrm{T}} R(q) b - \cos\theta \tag{5.34}$$

where $R(q)$ is

$$R(q) \triangleq \begin{bmatrix} 1-2q_2^2-2q_3^2 & 2q_1q_2-2q_0q_3 & 2q_0q_2+2q_1q_3 \\ 2q_0q_3+2q_1q_2 & 1-2q_1^2-2q_3^2 & 2q_2q_3-2q_0q_1 \\ 2q_1q_3-2q_0q_2 & 2q_0q_1+2q_2q_3 & 1-2q_1^2-2q_2^2 \end{bmatrix}. \tag{5.35}$$

This is a relative-degree 2 constraint function, since $\dot{\kappa}_b$ is not a function of u, ξ. Note that (5.34) *can be used to express both keep-out and keep-in zones. Also note that κ_b in* (5.34) *is equivalent to $\kappa_b^*(t,x) = q^{*\mathrm{T}} M q^*$ in [221, Eq. 2.5] where M is given in [221, Eq. 2.6] and $q^* = [-q_1, -q_2, -q_3, q_0]^{\mathrm{T}}$ is the conjugate of q with the scalar element q_0 last (the conjugate arises because of notational differences with [221]). Next, suppose that one also requires that the maximum angular rate of the spacecraft is bounded for safety of the spacecraft structure. This leads to the constraint function*

$$\eta_\omega(t,v) = \omega^{\mathrm{T}} P \omega - e_{\max} \tag{5.36}$$

where $e_{\max} \in \mathbb{R}$ and $P \in \mathbb{R}^{3\times 3}$ are given in Table 5.3. The values of $e_{\max}$ and P in Table 5.3 are constructed so that the safe set allows for angular rates of up to 1 deg/s *on the largest principal axis and up to* 2.730 deg/s *on the smallest principal axis, and will be elaborated upon in Case Study Parts xi-xii. This is a relative-degree 1 constraint, since $\dot{\eta}_\omega$ is a function of u, ξ. Finally, suppose that the wheel angular velocities are limited by the constraint functions*

$$\eta_{w_i}(t,v) = |w_i| - w_{\max},\ i = 1, \cdots, m \tag{5.37}$$

for $i \in \{1, \cdots, m\}$, where $w_{\max}$ is a constant. This work will assume that a suitable momentum dumping control law (e.g. scheduled thruster or magnetorquer application) has been developed so that the constraints encoded by $\eta_{w_i}(t,x)$ are always satisfied without impacting the rest of the control design. Thus, the following subsections only focus on the relative-degree 1 constraint in η_ω and the relative-degree 2 constraint in κ_b, though I still incorporate the wheel rate bounds in (5.37) *in the safe set construction in* (5.33)*. Finally, for this case study, suppose there are two constraints of the form* (5.34) *for body fixed vectors b_1 and b_2, so the safe set is $\mathcal{S} = \mathcal{Q}_{b_1} \cap \mathcal{Q}_{b_2} \cap \mathcal{V}_\omega \cap \mathcal{V}_{w_1} \cap \mathcal{V}_{w_2} \cap \mathcal{V}_{w_3} \cap \mathcal{V}_{w_4}$.*

5.2.2.3 Notes Regarding Continuous-Time RCBFs

Recall the definition of an RCBF on a set as in Definition 4.2 (where now $w_u = \xi$ and $w_x \equiv 0$). One interpretation of this definition is that a function h_i is an RCBF on $\mathcal{S}$ if there is sufficient

control authority given the set $\mathcal{U}$ that the total derivative of h_i can be appropriately upper bounded on $\mathcal{S}$ regardless of the disturbance value ξ in the considered set Ξ. Similarly, this section will develop a suitable "ZOH robustness margin" that serves as a combination of W in (4.7) and ν in (5.7). I then say that h_i is a "ZOH CBF" in Sections 5.2.3-5.2.4 if there exists sufficient control authority over $\mathcal{U}$ at every state in $\mathcal{H}_i$ to satisfy these margins.

Also, while Chapter 4 was concerned with only one RCBF at a time, Theorems 4.1, 4.3, 4.4 can be applied to any number of CBFs, so the following work seeks a collection of CBFs $\{h_i\}_{i=1}^M$ such that the intersection of the CBF sets $\cap_{i=1}^M \mathcal{H}_i$ is a subset of $\mathcal{S}$ in (5.33). As will be shown in Chapter 6, this can be achieved with an arbitrary number of CBFs, but in this chapter, I will aim to construct one CBF h_i for each constraint function κ_i or η_i (equivalently, one CBF set $\mathcal{H}_i$ for each constraint set $\mathcal{Q}_i$ or $\mathcal{V}_i$) in order to leverage the work in Chapter 4, though such a one-to-one correspondence is not necessary.

For each relative-degree 1 constraint η_i in (5.32b), I will choose the CBF $h_i \equiv \eta_i$ so $\mathcal{H}_i \equiv \mathcal{V}_i$. This choice will be further justified at the end of Section 5.2.4. For the relative-degree 2 constraints κ_i in (5.32a), the following work will extend the method in Section 4.3.1 specifically. Recall from Section 4.3.1 that for a constraint function κ_i satisfying certain properties, one possible choice of RCBF is

$$h_i(t,x) = \kappa_i(t,x) + \frac{\dot{\kappa}_i(t,x)|\dot{\kappa}_i(t,x)|}{2\mu}, \tag{5.38}$$

$$\mathcal{H}_i(t) = \{x \in \mathcal{X} \mid h(t,x) \leq 0\}, \tag{5.39}$$

for some parameter $\mu > 0$. Note that in this chapter, I replace $a_{\max}$ in (4.22) with μ in (5.38), as μ will be a tunable parameter rather than a specific value as in (4.24). The choice of CBF in (5.38) does not work for all systems, but is particularly useful for systems similar to the double integrator, such as a double integrator with small nonlinearities. I hypothesize that (5.38) can be used for pointing constraints as in (5.34), so this is the only CBF for relative-degree 2 constraint functions κ_i considered in the remainder of this section. Possible extensions of the other RCBFs in Chapter 4 to zero-order-hold control inputs would be an excellent area for future work, partially discussed in Section 5.2.6.2.

Let $\mu_1 > \mu_2 > 0$, and let h_{i,μ_1} and h_{i,μ_2} be two corresponding CBFs. Note that $\mathcal{H}_{i,\mu_1} \supset \mathcal{H}_{i,\mu_2}$. Thus, the least conservative CBF of the form (5.38) will have the largest allowable parameter μ. Also recall from Lemma 4.9 and Theorem 4.10 that, with the RCBF (5.38) on $\mathcal{Q}_i$, the RCBF condition (4.8) is sufficient to render $\mathcal{H}_i \cap \mathcal{Q}_i$ forward invariant rather than just $\mathcal{H}_i$, as visualized in Fig. 5.10. For simplicity, I restate this result as follows.

Lemma 5.10. *Let $[t_0, t_f] \subseteq \mathcal{T}$. Let $\kappa : \mathcal{T} \times \mathcal{X} \to \mathbb{R}$ be of relative-degree 2, and let $h : \mathcal{T} \times \mathcal{X} \to \mathbb{R}$ be as in* (5.38)*. If $x(t_0) \in \mathcal{Q}(t_0)$ and $x(t) \in \mathcal{H}(t)$ for all $t \in [t_0, t_f]$, then $x(t) \in \mathcal{Q}(t)$ for all $t \in [t_0, t_f]$.*

Thus, given one CBF $h_i, i = 1, \cdots N_1$ in (5.38) for each relative-degree 2 constraint function κ_i, and one CBF $h_i = \eta_i, i = N_1 + 1, N_1 + N_2$ for each relative-degree 1 constraint function η_i, leads to an effective operating subset $\mathcal{A}(t) \subseteq \mathcal{S}(t)$ where

$$\mathcal{A}(t) = \left(\bigcap_{i=1}^{N_1} (\mathcal{Q}_i(t) \cap \mathcal{H}_i(t)) \right) \cap \left(\bigcap_{i=N_1+1}^{N_1+N_2} \mathcal{H}_i(t) \right) . \tag{5.40}$$

To begin the analysis, consider the application of the continuous-time CBF (5.38) to the case study as follows.

Case Study Part iii (Continuous-Time CBF)**.** *For the constraint function κ_b in* (5.34)*, the function h_b in* (5.38) *is an RCBF on $\mathcal{S} \cap \mathcal{H}_b$ as in Definition 4.2 for any parameter $0 < \mu \leq 0.0025$.*

5.2.2.4 Robust Sampled-Data Formulation

The preceding subsection discussed results for continuous controller updates, but the goal of this work is to apply the CBFs (5.38) and (5.36) when the controller is instead updated at a fixed frequency. Now suppose that the control input u is updated at discrete times $t_k, k \in \mathbb{Z}_{\geq 0}$ where $t_{k+1} - t_k = T$ for fixed time-step $T > 0$, and that u is fixed between time steps k and $k + 1$. That is, consider a control law of the form (5.4), repeated here as

$$u_{\text{zoh}}(t) = u_k \triangleq u(t_k, x_k), \ \forall t \in [t_k, t_{k+1}) \, , \tag{5.41}$$

where $u : \mathcal{T} \times \mathcal{X} \to \mathcal{U}$ and $x_k = x(t_k)$. As discussed in Section 5.1, since the control input is updated only at the times t_k, it is difficult to ensure that the continuous-time CBF condition (2.5),(2.6),(3.9),(5.3) (or RCBF condition (4.8)) is satisfied at every time instant (i.e. including between time steps). This was partially solved in Section 5.1. However, the methods in Section 5.1 do not easily apply to CBFs constructed from relative-degree 2 constraint functions, such as in (5.38). This is demonstrated by way of the following example.

Example 5.1. *Given a relative-degree 2 constraint function κ_i, one possible CBF is that in* (5.38) *for some constant $\mu > 0$. According to Theorem 5.7, this CBF can be rendered safe in a ZOH fashion if for all $x \in \mathcal{H}_i(t), t \in \mathcal{T}$ there exists $u \in \mathcal{U}$ such that*

$$\dot{h}_i(t, x) = \dot{\kappa}_i(t, x) \left(1 + \frac{\ddot{\kappa}_i(t, x, u, \xi)}{\mu} \right) \leq -\frac{1}{T} \kappa_i(t, x) - \frac{1}{2} \nu_3^l(T, x) T \tag{5.42}$$

for ν_3^l in (5.19)*. Here, ν_3^l represents possible values of $\ddot{\kappa}_i$, and for that reason is usually lower bounded by a positive number, here denoted $\nu^* > 0$ (or one could use the global case of Corol-*

lary 5.8 and let $\nu^* = \nu_3^g(T)$).

The issue that arises here is that for any arbitrarily small $\delta > 0$, there exist $x \in \mathcal{H}_i(t), t \in \mathcal{T}$ such that $\dot{\kappa}_i(t,x) = \delta$ and $\kappa_i(t,x) = -\frac{\delta^2}{2\mu}$. For such x, (5.42) *simplifies to $\ddot{\kappa}_i(t,x,u,\xi) \leq \epsilon(\delta) \triangleq \frac{\delta}{2T} - \frac{\mu\nu_3^l T}{2\delta}$. Because $\nu_3^l \geq \nu^* > 0$, it follows that $\lim_{\delta\to 0^+} \epsilon(\delta) = -\infty$. That is, the ZOH sampling margin ν_3^l causes the required $\ddot{\kappa}_i$ to go to $-\infty$ near the boundary of $\mathcal{H}_i$, which also causes the required u to become unbounded. Thus, the methods in Section 5.1 cannot be applied to the CBF* (5.38) *(except along specific trajectories that avoid states where both h and $\dot{\kappa}$ are close to zero, as happened to occur in Section 5.1.4), or any of the relative-degree 2 strategies in Chapter 4, if there are also input constraints.*

Thus, even the improved method in Section 5.1 suffers from the possible infeasibility of the condition (5.5) when the control input u is constrained to a compact set. This is why that section did not yet present a definition of "ZOH CBF" and instead only presented the preliminary condition (5.5). The interested reader can examine this problem further by downloading the code linked in Section 5.1.4 and increasing the value of the constant μ in Table 5.1. Thus, the central problem of the present section is as follows.

Problem 5.2. *Given the safe set $\mathcal{S}$ in* (5.33)*, focus on a single constraint function $\eta_i : \mathcal{T} \times \mathbb{V} \to \mathbb{R}$ or $\kappa_i : \mathcal{T} \times \mathbb{Q} \to \mathbb{R}$ that is of relative-degree 1 or 2, respectively, with respect to the dynamical model* (5.30)*. Assume that $x(t_k) \in \mathcal{S}(t_k)$ in* (5.33) *at the current sample time k, and that all other constraints $x(t) \in \mathcal{Q}_j(t)$ and $x(t) \in \mathcal{V}_j(t)$ for $j \neq i$ are satisfied for all t in the inter-sample period $[t_k, t_{k+1})$. Sections 5.2.3-5.2.4 seek to derive a set $\mathcal{Z}_i(t) \subseteq \mathcal{Q}_i(t)$ or $\mathcal{Z}_i(t) \subseteq \mathcal{V}_i(t)$ for all $t \in \mathcal{T}$ and a set $\mathcal{U}_{\text{zoh},i}(t_k, x(t_k))$ such that, 1) given $x(t_k) \in \mathcal{Z}_i(t_k)$ and $u(t_k, x(t_k)) \in \mathcal{U}_{\text{zoh},i}(t_k, x(t_k))$ it is provably guaranteed that i) $x(t_{k+1}) \in \mathcal{Z}_i(t_{k+1})$ and ii) $x(t) \in \mathcal{Q}_i(t)$ or $x(t) \in \mathcal{V}_i(t)$ for all $t \in [t_k, t_{k+1})$ for any allowable disturbance $\xi \in \Xi$, and 2) the set $\mathcal{U} \cap \mathcal{U}_{\text{zoh},i}(t_k, x(t_k))$ is nonempty for all $x(t_k) \in \mathcal{Z}_i(t_k) \cap \mathcal{S}(t_k), k \in \mathcal{Z}_{\geq 0}$.*

I refer to set $\mathcal{Z}_i$ as the ith *robust CBF set*. Similar to how Chapter 4 restricted the constraint set $\mathcal{Q}_i$ (in that chapter, I used simply $\mathcal{S} \equiv \mathcal{Q}_i$ since there was only one constraint) to the CBF set $\mathcal{H}_i$ to account for input constraints, leading to the safe control set $\mathcal{U}_{\text{rcbf},i}$, in the following subsections, I will further restrict the allowable sampled states to the new set $\mathcal{Z}_i$ to account for disturbances and controller sampling, leading to the new safe control set $\mathcal{U}_{\text{zoh},i}$. Fig. 5.10 shows the relation between $\mathcal{Q}_i$ (cyan), $\mathcal{H}_i$ (hashed), and $\mathcal{Z}_i$ (gray) for a relative-degree 2 constraint function. Section 5.2.3 will address the relative-degree 2 case of Problem 5.2, while Section 5.2.4 will address the simpler relative-degree 1 case.

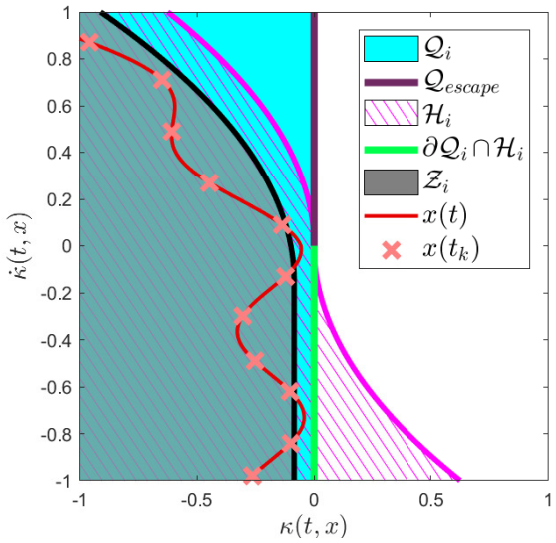

Figure 5.10: Visualization of Constraint Set, CBF Set, and ZOH CBF Set. Diagram of a constraint set $\mathcal{Q}_i$, corresponding CBF set $\mathcal{H}_i$, robust CBF set $\mathcal{Z}_i$, and a safe trajectory $x(t) \in \mathcal{Q}_i \cap \mathcal{H}_i$ with controller samples $x(t_k) \in \mathcal{Z}_i$ at the red 'x' marks. In this case, $\mathcal{H}_i \not\subseteq \mathcal{Q}_i$, which is allowed since Lemma 5.10 ensures that state trajectories cannot cross the green line $(\partial\mathcal{Q}_i) \cap \mathcal{H}_i$. The robust CBF set $\mathcal{Z}_i$ is a subset of $\mathcal{Q}_i \cap \mathcal{H}_i$ with additional margin. The trajectory $x(t)$ lies within the set $\mathcal{Z}_i(t_k)$ at every sample time $t_k, k \in \mathbb{Z}_{\geq 0}$, and thus lies within $\mathcal{Q}_i(t) \cap \mathcal{H}_i(t)$ for all $t \in \mathcal{T}$. $\mathcal{Q}_{escape}$ is the set of points on $\partial\mathcal{Q}_i$ that immediately leave $\mathcal{Q}_i$ and therefore are excluded from $\mathcal{H}_i \cap \mathcal{Q}_i$ and from $\mathcal{Z}_i$.

5.2.3 Method for Relative-Degree Two

5.2.3.1 Strategy

I begin by addressing the relative-degree 2 case of Problem 5.2, as the relative-degree 1 case easily follows. In this subsection, I drop the subscript i, so let κ denote any relative-degree 2 constraint function (e.g. (5.34)), and h the corresponding CBF as in (5.38). The core idea of this method is that given $h(t_k, x(t_k)) \leq 0$, I want to identify a formula for a worst case value of $h(t_k + \tau, x(t_k + \tau))$, denoted $h_{\text{bound}}(t_k, x(t_k), \tau)$, for $\tau \in [0, T]$ and find a suitable control input to ensure that $h_{\text{bound}}(t_k, x(t_k), \tau)$ is nonpositive for all $\tau \in [0, T]$. However, the problem highlighted in Example 5.1 is that the worst case formulas h_{bound} following from all the methods in Section 5.1 and [119] rely upon linear approximations of h on the interval $[t_k, t_{k+1}]$. The obvious extension is to use a higher-order approximation of the worst case trajectory that h could follow between time steps. However, when using a higher order approximation, it is no longer sufficient to only check that $h_{\text{bound}}(t_k, x(t_k), T) \leq 0$, as there may exist $\tau \in (0, T)$ such that $h_{\text{bound}}(t_k, x(t_k), \tau) > h_{\text{bound}}(t_k, x(t_k), T)$, as visualized by the red and cyan points in Fig. 5.11. Thus, unlike in Section 5.1, one must instead check that $h_{\text{bound}}(t_k, x(t_k), \tau) \leq 0$ for all $\tau \in [0, T]$, which adds complexity to the problem. To address this possibility of exiting and returning to the CBF set, I seek local maximizers σ (e.g. the red circle in Fig. 5.11) such that $h_{\text{bound}}(t_k, x(t_k), \sigma) \geq h_{\text{bound}}(t_k, x(t_k), \tau)$ for all $\tau \in [0, T]$. I then identify a bound Δ on the differences $h_{\text{bound}}(t_k, x(t_k), \sigma) - h(t_k, x(t_k))$ and $h_{\text{bound}}(t_k, x(t_k), \sigma) - h_{\text{bound}}(t_k, x(t_k), T)$ and define the sets

$$\mathcal{H}^{\Delta}(t) \triangleq \{x \in \mathcal{X} \mid h(t, x) \leq -\Delta\}, \tag{5.43}$$

$$\mathcal{Q}^{\delta}(t) \triangleq \{x \in \mathcal{X} \mid \kappa(t, x) \leq -\delta\}. \tag{5.44}$$

It follows that if $h(t_k, x(t_k)) \leq -\Delta$ and $h_{\text{bound}}(t_k, x(t_k), T) \leq -\Delta$, which are relatively simple conditions to enforce (e.g. see (5.93)), then $h(t_k + \tau, x(t_k + \tau)) \leq h_{\text{bound}}(t_k, x(t_k), \tau) \leq h_{\text{bound}}(t_k, x(t_k), \sigma) \leq 0$ for all $\tau \in [0, T]$. This is visualized in Fig. 5.10, where the red sample trajectory is always safe, because at the sample times t_k, the trajectory meets the stricter condition of being in the gray set.

To define a set $\mathcal{Z}$ as in Problem 5.2, I will derive expressions for suitable Δ in (5.43) and δ in (5.44), from which it will follow that $\mathcal{Z} = \mathcal{H}^{\Delta} \cap \mathcal{Q}^{\delta}$. I seek to minimize conservatism, i.e. to choose the smallest δ, Δ for which I can still provably demonstrate safety between each t_k and t_{k+1}. To this end, Sections 5.2.3.3-5.2.3.4 study possible expressions for δ, Δ that work well for the system (5.31), and that lead to a final control strategy summarized in Theorem 5.16.

I begin by presenting a naive approach to determining Δ, δ. Assuming a second order h_{bound} function (such as that in (5.92)), the required margin Δ can be determined entirely by the values

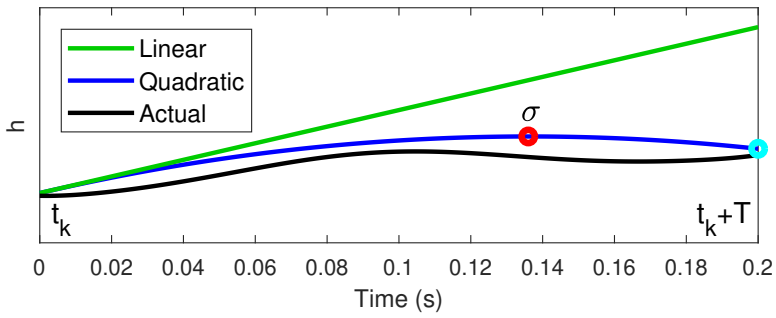

Figure 5.11: Diagram of Linear and Quadratic Upper Bounds. Diagram of a linear and quadratic upper bound on the trajectory of $h(t, x(t))$ between two sampled times, and the maximizer σ of the quadratic curve

of the second derivative $\ddot{h}$. Thus, consider the following (very conservative) baseline example with numbers derived from the case study.

Case Study Part iv (A Naive Approach to Computing Maximum Overshoot)**.** *Let the time-step for the controller of* (5.31) *be* $T = 0.2$ s. *Let* h_b *as in* (5.38) *be a CBF for* κ_b *in* (5.34) *and suppose* $\mu = 0.00167$ *as in Table 5.4 (which I will justify later). Suppose* $\xi \equiv 0$ *for this example. Let* $r = \min_{x\in\mathcal{S}(t),t\in\mathcal{T},u\in\mathcal{U}} \ddot{h}_b(t, x, u, 0) = -0.550$. *It follows that one possible upper bound on the overshoot of* h_b *between time steps is* $\Delta = -\frac{1}{8}T^2 r = 2.75(10)^{-3}$. *I will show in Sections 5.2.3.3-5.2.3.4 that this is over 200 times as conservative as necessary for this system.*

5.2.3.2 Sampling and Robustness Constants

I now proceed similarly to Section 5.1 by defining several constants of the system, analogous to the Lipschitz constants in Section 5.1, and then using these constants to bound system behavior. First, define

$$M_2^- \triangleq \inf_{t\in\mathcal{T},x\in\mathcal{S}(t),\xi\in\Xi} \frac{\partial\dot{\kappa}(t,x)}{\partial v} g_2(t,x)\xi \tag{5.45a}$$

$$M_2^+ \triangleq \sup_{t\in\mathcal{T},x\in\mathcal{S}(t),\xi\in\Xi} \frac{\partial\dot{\kappa}(t,x)}{\partial v} g_2(t,x)\xi \tag{5.45b}$$

The constants M_2^- and M_2^+ represent bounds on the uncertainty in $\ddot{\kappa}$ because of the unknown disturbance. I assume that (5.45) and (5.48) are well-defined. I represent the component of $\ddot{\kappa}$ that is certain using the function ψ (similar to (5.17), now for the model (5.30)) as follows

$$\psi(t,x,u) \triangleq \frac{\partial\dot{\kappa}(t,x)}{\partial t} + \frac{\partial\dot{\kappa}(t,x)}{\partial q} f_1(t,x) + \frac{\partial\dot{\kappa}(t,x)}{\partial v}\left(f_2(t,x) + g_1(t,x)u\right) \tag{5.46}$$

so that

$$\ddot{\kappa}(t,x,u) \leq \psi(t,x,u) + M_2^+ \tag{5.47a}$$

$$\ddot{\kappa}(t,x,u) \geq \psi(t,x,u) + M_2^- \tag{5.47b}$$

In practice, the value of ψ is exactly known only at the sampling times t_k, so I also define the constants

$$M_3^- \triangleq \inf_{t\in\mathcal{T},x\in\mathcal{S}(t),u\in\mathcal{U},\xi\in\Xi} \Bigg[\frac{\partial\psi(t,x,u)}{\partial t} + \frac{\partial\psi(t,x,u)}{\partial q} f_1(t,x) + \frac{\partial\psi(t,x,u)}{\partial v}\left(f_2(t,x) + g_1(t,x)u + g_2(t,x)\xi\right)\Bigg] \tag{5.48a}$$

$$M_3^+ \triangleq \sup_{t\in\mathcal{T},x\in\mathcal{S}(t),u\in\mathcal{U},\xi\in\Xi} \Bigg[\frac{\partial\psi(t,x,u)}{\partial t} + \frac{\partial\psi(t,x,u)}{\partial q} f_1(t,x) + \frac{\partial\psi(t,x,u)}{\partial v}\left(f_2(t,x) + g_1(t,x)u + g_2(t,x)\xi\right)\Bigg] \tag{5.48b}$$

to describe the uncertainty in the evolution of ψ between time steps due to both the ZOH sampling and the disturbance. That is, for $\tau > 0$,

$$\psi(t+\tau, x(t+\tau), u) \leq \psi(t, x(t), u) + M_3^+\tau \tag{5.49a}$$

$$\psi(t+\tau, x(t+\tau), u) \geq \psi(t, x(t), u) + M_3^-\tau \tag{5.49b}$$

Note that the control input u is the same on both sides of the inequalities in (5.49), so these inequalities are only useful during a single ZOH time step. Also, note that now, unlike in Section 5.1, I assume the bounds $M_2^-, M_2^+, M_3^-, M_3^+$ are global (i.e. are computed over all of $\mathcal{S}$) for simplicity. Extensions of this work for local bounds computed online as in Section 5.1 could also be developed, but I found this computation to be one of the more difficult elements of Section 5.1 to implement. If the global bounds (5.45), (5.48) are undefined, then more involved analysis than is presently considered may be required. Note that I defined the lower bounds M_2^-, M_3^- and upper bounds M_2^+, M_3^+ separately to cover cases such as when the dynamics and/or disturbance environment are known to tend to increase/decrease h (e.g. if the unsafe state is of higher/lower potential energy than other states, such as would occur if gravity gradient were included in (5.31)). In other cases, it may occur that $M_2^- = -M_2^+$ and $M_3^- = -M_3^+$.

In the upcoming theorems, I will need the following relations. Let σ be some time in $[t_k, t_{k+1}]$ where u is constant on $[t_k, t_{k+1})$, and let $\tau \in \mathbb{R}_{\geq 0}$ be such that $\sigma + \tau$ (or $\sigma - \tau$) is also within

Symbol	Value
T	0.2 s
M_2^+	$1.64(10)^{-4}$
M_2^-	$-1.64(10)^{-4}$
M_3^+	$6.2(10)^{-3}$
M_3^-	$-6.2(10)^{-3}$
δ_1	$1.10(10)^{-5}$
δ_2	$9.7(10)^{-6}$
Δ_2	$1.3(10)^{-5}$
Δ_3	$1.09(10)^{-5}$
μ	0.00167
M_1	$5.79(10)^{-7}$
M_2^{alt}	$1.95(10)^{-5}$

Table 5.4: System Constants for Case Study

$[t_k, t_{k+1}]$. Then, using only the time argument for brevity, it holds that

$$\ddot{\kappa}(\sigma) \overset{(5.47a)}{\leq} \psi(\sigma) + M_2^+ \overset{(5.49b)}{\leq} \psi(\sigma+\tau) - M_3^-\tau + M_2^+ \overset{(5.47b)}{\leq} \ddot{\kappa}(\sigma+\tau) - M_2^- - M_3^-\tau + M_2^+ \quad (5.50a)$$

$$\ddot{\kappa}(\sigma) \overset{(5.47b)}{\geq} \psi(\sigma) + M_2^- \overset{(5.49a)}{\geq} \psi(\sigma+\tau) - M_3^+\tau + M_2^- \overset{(5.47a)}{\geq} \ddot{\kappa}(\sigma+\tau) - M_2^+ - M_3^+\tau + M_2^- \quad (5.50b)$$

$$\ddot{\kappa}(\sigma) \overset{(5.47b)}{\geq} \psi(\sigma) + M_2^- \overset{(5.49b)}{\geq} \psi(\sigma-\tau) + M_3^-\tau + M_2^- \overset{(5.47a)}{\geq} \ddot{\kappa}(\sigma-\tau) - M_2^+ + M_3^-\tau + M_2^- \quad (5.50c)$$

$$\ddot{\kappa}(\sigma) \overset{(5.47a)}{\leq} \psi(\sigma) + M_2^+ \overset{(5.49a)}{\leq} \psi(\sigma-\tau) + M_3^+\tau + M_2^+ \overset{(5.47b)}{\leq} \ddot{\kappa}(\sigma-\tau) - M_2^- + M_3^+\tau + M_2^+ \quad (5.50d)$$

Case Study Part v (Constants)**.** *For the system* (5.31) *and constraint* κ_b *in* (5.34)*, the values of* $M_2^-, M_2^+, M_3^-, M_3^+$ *are given in Table 5.4. Note that these values hold for all* $\theta \leq \pi/2$ *in* (5.34)*, and are larger than the resultant values (i.e. are overly conservative) when* $\theta > \pi/2$*.*

5.2.3.3 Determining Δ, δ when h and κ share maximizers

Using the above constants, I now determine suitable values of Δ, δ for (5.43),(5.44) in two parts. First, I note that a necessary condition for a maximizer σ of h occurring between time steps t_k and t_{k+1} is $\dot{h}(\sigma, x(\sigma), u_k, \xi) = 0$. Because of the form of h in (5.38), a sufficient condition for $\dot{h} = 0$ is $\dot{\kappa} = 0$, so maximizers of h will often be co-located with maximizers of κ, as illustrated by the blue lines in Fig. 5.12. Thus, this subsection determines appropriate margins Δ, δ specifically when the maximizers of κ and h are co-located, while the following subsection determines these margins when this is not the case (green lines in Fig. 5.12). I begin with the following lemma.

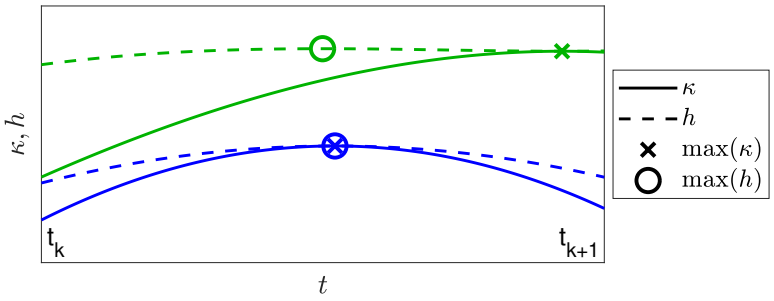

Figure 5.12: Illustration of Co-Located and Distinct Maximizers of h and κ. Illustration of trajectories where the maximizers of κ and h on $[t_k, t_{k+1}]$ are co-located (blue), and where the maximizer of h precedes that of κ (green)

Lemma 5.11. *Suppose κ is thrice continuously differentiable and of relative-degree 2 with respect to* (5.30)*, and u is constant on $[t_k, t_{k+1})$. If $\sigma \in (t_k, t_{k+1})$ is the time of a local maximizer of h in* (5.38) *on (t_k, t_{k+1}) and $\dot{\kappa}(\sigma, x(\sigma)) = 0$, then σ is also a local maximizer of κ on (t_k, t_{k+1}) and it must hold that $0 \geq \ddot{\kappa}(\sigma, x(\sigma), u, \xi) \geq -\mu$.*

Proof. For brevity, in this proof, I only write out the time argument of κ, h, and their derivatives. Note that on the open interval (t_k, t_{k+1}), the functions κ, $\dot{\kappa}$, $\ddot{\kappa}$, and $\dddot{\kappa}$ are continuous due to the assumptions on f_1, f_2, g_1, g_2, ξ in (5.30) and how u is constant. I divide this proof into two cases depending on whether $\dot{\kappa}$ changes signs at σ.

First, suppose that $\dot{\kappa}$ does not change signs at σ (including the case where $\dot{\kappa}(t) = 0$ for all $t \in (t_k, t_{k+1})$). This can only occur if $\ddot{\kappa}$ changes signs at σ or if $\ddot{\kappa}(t)$ is zero for all $t \in (t_k, t_{k+1})$. Since $\ddot{\kappa}$ is continuous, this implies $\ddot{\kappa}(\sigma) = 0$. Since $\mu > 0$, it follows that there exists sufficiently small $\tau > 0$ such that $|\ddot{\kappa}(t)| < \mu$ for all t in a neighborhood $t \in (\sigma - \tau, \sigma + \tau)$. Note that $\dot{h}$ is given by

$$\dot{h}(t) = \dot{\kappa}(t) + \frac{1}{\mu}|\dot{\kappa}(t)|\ddot{\kappa}(t)\,. \tag{5.51}$$

It follows from (5.51) that $\dot{h}(t)$ has the same sign as $\dot{\kappa}(t)$ for all $t \in (\sigma - \tau, \sigma + \tau)$. Since $\dot{h}(\sigma)$ does not change signs, σ cannot be a local maximizer of h on (t_k, t_{k+1}) unless $\dot{\kappa}(t) = \dot{h}(t) = \ddot{\kappa}(t) = 0$ for all $t \in (t_k, t_{k+1})$, in which case the lemma is trivially true.

Second, suppose that $\dot{\kappa}$ does change signs at σ. The second derivative of h is

$$\ddot{h}(t) = \ddot{\kappa}(t) + \frac{1}{\mu}\operatorname{sign}(\dot{\kappa}(t))\ddot{\kappa}(t)^2 + \frac{1}{\mu}|\dot{\kappa}(t)|\,\dddot{\kappa}(t) \tag{5.52}$$

so h is twice continuously differentiable for almost all $t \in [t_k, t_{k+1}]$. Therefore, a necessary condition for σ to be local maximizer of h is for $\ddot{h}$ to be nonpositive in a neighborhood of σ. At $t = \sigma$ exactly, $\ddot{h}(\sigma)$ is undefined since the lemma assumed $\dot{\kappa}(\sigma)$ to be zero, but the limits of $\ddot{h}(t)$ as t approaches σ from the left and right are well-defined and must both be nonpositive for σ to be

a maximizer of h. These limits are

$$\ddot{h}^-(\sigma) = \lim_{\dot{\kappa}(t)\to 0^-} \ddot{h}(t) = \ddot{\kappa}(\sigma) - \frac{\dddot{\kappa}(\sigma)^2}{\mu}, \tag{5.53a}$$

$$\ddot{h}^+(\sigma) = \lim_{\dot{\kappa}(t)\to 0^+} \ddot{h}(t) = \ddot{\kappa}(\sigma) + \frac{\dddot{\kappa}(\sigma)^2}{\mu}. \tag{5.53b}$$

A necessary condition for both (5.53a) and (5.53b) to be nonpositive simultaneously is for $0 \geq \ddot{\kappa}(\sigma) \geq -\mu$. Since $\ddot{\kappa}(\sigma) \leq 0$, and this case assumed $\dot{\kappa}(t)$ changed sign at σ, it follows that σ is necessarily also a maximizer of κ, so the lemma holds in this case as well. ■

The consequence of Lemma 5.11 is that, provided the stated condition holds, it is now possible to use knowledge about κ to upper bound the variation in h between time steps. Lemma 5.11 is particularly helpful because analysis of κ is generally simpler than analysis of h, and because μ is a tunable parameter. Note that maximizers of the CBF h can also occur when $\dot{\kappa}(\sigma, x(\sigma)) \neq 0$ and in these cases Lemma 5.11 would no longer apply, thus motivating Lemma 5.14 in Section 5.2.3.4. However, when Lemma 5.11 does hold, one can substantially reduce the required conservatism to prevent $x(t)$ from leaving $\mathcal{H}(t)$ between sample times, as illustrated using the case study as follows.

Case Study Part vi (Application of Lemma 5.11)**.** *Suppose the same setup as in Case Study Part iv and suppose that the conditions of Lemma 5.11 hold. It follows that* $h(\sigma, x(\sigma)) = \kappa(\sigma, x(\sigma))$*, so* Δ *can be computed as a bound on the possible overshoot of* κ *(instead of overshoot of* h*) between time steps. Here, let* $r = \min_{x\in\mathcal{S}(t),t\in\mathcal{T},u\in\mathcal{U}} \ddot{\kappa}(t, x, u, 0) = -0.0262$ *(recall Case Study Part iv assumed* $\xi \equiv 0$*), which leads to the new bound* $\Delta = -\frac{1}{8}T^2 r = 1.31(10)^{-4}$*. Finally, since Lemma 5.11 also says that* $\ddot{\kappa}(\sigma, x(\sigma), u, \xi) \geq -\mu$*, and since* $\min_{x\in\mathcal{S}(t),t\in\mathcal{T},u\in\mathcal{U}} \dddot{\kappa}(t, x, u, 0) = -0.0062$*, it follows that* $\ddot{\kappa}(t, x(t), u, 0) \geq r = -\mu - 0.0062T = -0.00291$ *for all* $t \in [t_k, t_{k+1}]$ *assuming* $\sigma \in [t_k, t_{k+1}]$*, which leads to* $\Delta = -\frac{1}{8}T^2 r = 1.46(10)^{-5}$*. Thus, Lemma 5.11 reduces the conservatism* Δ *on* $\mathcal{H}$ *needed to ensure safety during the between-sample interval by a factor of 188 compared to Case Study Part iv.*

I now apply Lemma 5.11 to calculate a general formula for appropriate margins Δ, δ in (5.43),(5.44) on $h(t_k, x_k), \kappa(t_k, x_k)$ to ensure x remains safe between sampling times.

Theorem 5.12. *Suppose* κ *is thrice continuously differentiable and of relative-degree 2 with respect to* (5.30)*, and* u *is constant on* $[t_k, t_{k+1})$*. Suppose that all maximizers* σ *of* h *in* (5.38) *on the*

interval (t_k, t_{k+1}) *satisfy* $\dot{\kappa}(\sigma, x(\sigma)) = 0$. *Define* δ_1 *as*

$$\delta_1 \triangleq \max_{\tau \in [0,T]} \Bigg[\min \Bigg\{ \frac{1}{2}(\mu + M_2^+ - M_2^-)(T-\tau)^2 - \frac{1}{6} M_3^- (T-\tau)^3, \\ \frac{1}{2}(\mu + M_2^+ - M_2^-)\tau^2 + \frac{1}{6} M_3^+ \tau^3 \Bigg\} \Bigg] \quad (5.54)$$

If $x(t_k) \in \mathcal{H}^{\delta_1}(t_k) \cap \mathcal{Q}(t_k)$ *in* (5.43),(5.32) *and* $x(t_{k+1}) \in \mathcal{H}(t_{k+1}) \cap \mathcal{Q}^{\delta_1}(t_{k+1})$ *in* (5.39),(5.44)*, then* $x(t) \in \mathcal{H}(t)$ *for all* $t \in [t_k, t_{k+1}]$.

Proof. For brevity, in this proof, I only write out the time argument of κ, h, and their derivatives. The trajectory $x(t)$ belongs to $\mathcal{H}(t)$ for all $t \in [t_k, t_{k+1}]$ if the maximum value $h(\sigma)$ for some maximizer $\sigma \in [t_k, t_{k+1}]$ satisfies $h(\sigma) \leq 0$, so I proceed by trying to bound $h(\sigma)$ using Lemma 5.11 and the system constants (5.45), (5.48). By assumption, $h(t_k) \leq -\delta_1 \leq 0$ and $h(t_{k+1}) \leq 0$, so the theorem is immediately true in the case where σ is either endpoint. By Lemma 5.11, if σ is a local maximizer of h on the open interval (t_k, t_{k+1}), then σ must also be a local maximizer of κ. This implies $h(\sigma) = \kappa(\sigma)$. Thus, I focus on κ instead of h going forward.

Suppose there exists a local maximizer σ of κ on (t_k, t_{k+1}) for which $\kappa(\sigma) = h(\sigma) > \max\{h(t_k), h(t_{k+1})\}$. The largest possible value of $\kappa(\sigma)$ occurs when $\dot{\kappa}(t)$ is positive for all $t \in [t_k, \sigma)$ and negative for all $t \in (\sigma, t_{k+1}]$, so without loss of generality, suppose that the sign of $\dot{\kappa}(t)$ follows this partitioning. By assumption, $h(t_k) \leq -\delta_1$, and since this proof assumed that $\dot{\kappa}(t_k) > 0$, it follows from (5.38) that $\kappa(t_k) \leq -\delta_1$ as well. Thus, for the worst case value of $\kappa(\sigma)$, both $\kappa(t_k)$ and $\kappa(t_{k+1})$ are at most $-\delta_1$. Also, by Lemma 5.11, $\ddot{\kappa}(\sigma) \geq -\mu$. It follows from (5.50a) that

$$\ddot{\kappa}(\sigma + \tau) \geq -\mu + M_2^- + M_3^- \tau - M_2^+ \,. \quad (5.55)$$

Thus, $\dot{\kappa}(t)$ can be lower bounded for $t \in (\sigma, t_{k+1}]$ as

$$\begin{aligned} \dot{\kappa}(\sigma + \tau) &= \underbrace{\dot{\kappa}(\sigma)}_{=0} + \int_{\sigma}^{\sigma+\tau} \ddot{\kappa}(t) dt \\ &\overset{(5.55)}{\geq} \int_{\sigma}^{\sigma+\tau} \left(-\mu - M_2^+ + M_3^-(t-\sigma) + M_2^- \right) dt \\ &= (-\mu - M_2^+ + M_2^-)\tau + \frac{1}{2} M_3^- \tau^2 \end{aligned} \quad (5.56)$$

and thus $\kappa(t)$ can be lower bounded for $t \in (\sigma, t_{k+1}]$ as

$$\kappa(\sigma + \tau) = \kappa(\sigma) + \int_{\sigma}^{\sigma+\tau} \dot{\kappa}(t) dt$$

$$\overset{(5.56)}{\geq} h(\sigma) + \int_{\sigma}^{\sigma+\tau} \left((-\mu - M_2^+ + M_2^-)(t-\sigma) + \frac{1}{2} M_3^- (t-\sigma)^2 \right) dt$$
$$= \kappa(\sigma) + \frac{1}{2}(-\mu - M_2^+ + M_2^-)\tau^2 + \frac{1}{6} M_3^- \tau^3 \,. \tag{5.57}$$

Similarly, it follows from (5.50d) that

$$\ddot{\kappa}(\sigma - \tau) \geq -\mu + M_2^- - M_3^+ \tau - M_2^+ \tag{5.58}$$

Thus, $\dot{\kappa}(t)$ can be upper bounded for $t \in [t_k, \sigma)$ as

$$\begin{aligned}\dot{\kappa}(\sigma - \tau) &= \underbrace{\dot{\kappa}(\sigma)}_{=0} - \int_{\sigma-\tau}^{\sigma} \dot{\kappa}(t) dt \\ &\overset{(5.58)}{\leq} - \int_{\sigma-\tau}^{\sigma} \left[-\mu - M_2^+ - M_3^+ (\sigma - t) + M_2^- \right] dt \\ &= (\mu + M_2^+ - M_2^-)\tau + \frac{1}{2} M_3^+ \tau^2 \end{aligned} \tag{5.59}$$

and thus $\kappa(t)$ can be lower bounded for $t \in [t_k, \sigma)$ as

$$\begin{aligned}\kappa(\sigma - \tau) &= \kappa(\sigma) - \int_{\sigma-\tau}^{\sigma} \dot{\kappa}(t) dt \\ &\overset{(5.59)}{\geq} \kappa(\sigma) - \int_{\sigma-\tau}^{\sigma} \left(- (-\mu - M_2^+ + M_2^-)(\sigma - t) + \frac{1}{2} M_3^+ (\sigma - t)^2 \right) dt \\ &= \kappa(\sigma) + \frac{1}{2}(-\mu - M_2^+ + M_2^-)\tau^2 - \frac{1}{6} M_3^+ \tau^3 \,. \end{aligned} \tag{5.60}$$

I then rearrange (5.57) and (5.60) to

$$\kappa(\sigma) \leq \kappa(\sigma + \tau) - \frac{1}{2}(-\mu - M_2^+ + M_2^-)\tau^2 - \frac{1}{6} M_3^- \tau^3 \text{ and} \tag{5.61}$$
$$\kappa(\sigma) \leq \kappa(\sigma - \tau) - \frac{1}{2}(-\mu - M_2^+ + M_2^-)\tau^2 + \frac{1}{6} M_3^+ \tau^3 \,. \tag{5.62}$$

Let $\tau = t_{k+1} - \sigma$ in (5.61) and $\tau = \sigma - t_k$ in (5.62). Note that the bounds in (5.61) and (5.62) must both apply simultaneously, or equivalently, whichever bound is tighter must apply. Thus, it follows that

$$\begin{aligned}\kappa(\sigma) \overset{(5.61),(5.62)}{\leq} \min \Bigg\{ &-\delta_1 + \frac{1}{2}(\mu + M_2^+ - M_2^-)(t_{k+1} - \sigma)^2 - \frac{1}{6} M_3^- (t_{k+1} - \sigma)^3, \\ &-\delta_1 + \frac{1}{2}(\mu + M_2^+ - M_2^-)(\sigma - t_k)^2 + \frac{1}{6} M_3^+ (\sigma - t_k)^3 \Bigg\} \overset{(5.54)}{\leq} 0 \,. \end{aligned} \tag{5.63}$$

Thus, because of the choice of δ_1 in (5.54), it is guaranteed that $\kappa(\sigma) \leq 0$ in (5.63). It follows that $h(t) \leq h(\sigma) = \kappa(\sigma) \leq 0$ for all $t \in [t_k, t_{k+1}]$, or equivalently $x(t) \in \mathcal{H}(t)$ for all $t \in [t_k, t_{k+1}]$. ■

Case Study Part vii (Application of Theorem 5.12)**.** *Using the values of $M_2^-, M_2^+, M_3^-, M_3^+, \mu$ in Table 5.4, it follows that $\delta_1 = 1.10(10)^{-5}$ in* (5.54)*. This is of similar magnitude to the value of Δ in Case Study Part vi, as expected, and is equivalent to 2.27 arc-seconds of shrinkage of the CBF set.*

Thus, in the case where the maximizers of κ and h are consistent, I have presented an explicit formula for how much to further restrict the set $\mathcal{H}$ at the sample times to ensure that the state never leaves the set $\mathcal{H}$ between the sample times. Having established this, I note that the requirements of Theorem 5.12 are still overly conservative. This is because I assumed that $\mathcal{H}^\Delta$ and $\mathcal{Q}^\delta$ were defined using the same margin parameter $\Delta = \delta = \delta_1$. For certain systems, applying different margins Δ_2 on $h(t_k, x_k)$ and δ_2 on $\kappa(t_k, x_k)$ may reduce this margin, as presented in the following theorem.

Theorem 5.13. *Suppose κ is thrice continuously differentiable and of relative-degree 2 with respect to* (5.30)*, and u is constant on $[t_k, t_{k+1})$. Suppose that all maximizers σ of h in* (5.38) *on the interval (t_k, t_{k+1}) satisfy $\dot{\kappa}(\sigma, x(\sigma)) = 0$. Suppose there exists constants $\delta_2 \geq 0$ and $\Delta_2 \geq 0$ for which it holds that*

$$\max_{\tau \in [0,T]} \left[\min \left\{ -\Delta_2 + \frac{1}{2}(\mu + M_2^+ - M_2^-)(T-\tau)^2 - \frac{1}{6}M_3^-(T-\tau)^3, \right. \right. \\ \left. \left. -\delta_2 + \frac{1}{2}(\mu + M_2^+ - M_2^-)\tau^2 + \frac{1}{6}M_3^+\tau^3 \right\} \right] \leq 0\,. \quad (5.64)$$

If $x(t_k) \in \mathcal{H}^{\Delta_2}(t_k) \cap \mathcal{Q}(t_k)$ in (5.43),(5.32) *and $x(t_{k+1}) \in \mathcal{H}^{\Delta_2}(t_{k+1}) \cap \mathcal{Q}^{\delta_2}(t_{k+1})$ in* (5.43),(5.44)*, then $x(t) \in \mathcal{H}(t)$ in* (5.39) *for all $t \in [t_k, t_{k+1}]$.*

Proof. The proof follows almost identical logic to that of Theorem 5.12, but instead of having $\kappa(t_k) \leq -\delta_1$ and $\kappa(t_{k+1}) \leq -\delta_1$, one ends up with $\kappa(t_k) \leq -\Delta_2$ and $\kappa(t_{k+1}) \leq -\delta_2$. Thus, in place of (5.63), it follows that

$$\kappa(\sigma) \leq \min \left\{ -\delta_2 + \frac{1}{2}(\mu + M_2^+ - M_2^-)(t_{k+1}-\sigma)^2 - \frac{1}{6}M_3^-(t_{k+1}-\sigma)^3, \right. \\ \left. -\Delta_2 + \frac{1}{2}(\mu + M_2^+ - M_2^-)(\sigma - t_k)^2 + \frac{1}{6}M_3^+(\sigma - t_k)^3 \right\} \overset{(5.64)}{\leq} 0\,. \quad (5.65)$$

Similar to in Theorem 5.12, the condition on Δ_2 and δ_2 in (5.64) ensures that $\kappa(\sigma) \leq 0$ regardless of the actual maximizer location $\sigma \in (t_k, t_{k+1})$. By the same logic as in Theorem 5.12, it follows

that $x(t) \in \mathcal{H}(t)$ for all $t \in [t_k, t_{k+1}]$. ■

The primary difference between Theorem 5.12 and Theorem 5.13 is that in Theorem 5.12, the form for δ_1 was provided explicitly. On the other hand, in Theorem 5.13, neither δ_2 nor Δ_2 is uniquely defined. If one fixes either δ_2 or Δ_2, then one can use condition (5.64) to compute the other constant. It follows from Theorem 5.12 that one valid combination is $\Delta_2 = \delta_2 = \delta_1$. Another helpful strategy is to set $\Delta_2 = \Delta_3$, where Δ_3 is presented in the next subsection, and to then compute the smallest allowable δ_2. I also note that, unlike Theorem 5.12, the conditions of Theorem 5.13 are recursively feasible. That is, the ending condition $x(t_{k+1}) \in \mathcal{H}^{\Delta_2}(t_{k+1}) \cap \mathcal{Q}^{\delta_2}(t_{k+1})$ at time t_{k+1} implies the starting condition $x(t_k) \in \mathcal{H}^{\Delta_2}(t_k) \cap \mathcal{Q}(t_k)$ when k advances by one step.

Case Study Part viii (Application of Theorem 5.13)**.** *Using the values of $M_2^-, M_2^+, M_3^-, M_3^+, \mu$ in Table 5.4, one possible combination satisfying* (5.64) *besides $\Delta_2 = \delta_2 = \delta_1$ is $\Delta_2 = 1.3(10)^{-5}$ and $\delta_2 = 9.7(10)^{-6}$.*

5.2.3.4 Determining Δ, δ when h and κ have distinct maximizers

Now that I have thoroughly covered excursions outside the sets $\mathcal{H}^{\Delta}, \mathcal{Q}^{\delta}$ when Lemma 5.11 applies, I finally discuss the behavior between sampling times when this is not the case, as is illustrated by the green lines in Fig. 5.12.

Lemma 5.14. *Suppose κ is thrice continuously differentiable and of relative-degree 2 with respect to* (5.30)*, and u is constant on $[t_k, t_{k+1})$ where $t_{k+1} = t_k + T$. Suppose that $M_3^+ > 0$, $M_3^- < 0$, and $\mu \geq M_2^+ - M_2^- + (\max\{|M_3^+|, |M_3^-|\})T$. Define Δ_3 as in* (5.88)*. Suppose there exists a maximizer time $\sigma \in (t_k, t_{k+1})$ for which $h(\sigma, x(\sigma)) \geq h(t, x(t))$ for all $t \in [t_k, t_{k+1}]$ at which $\dot{\kappa}(\sigma, x(\sigma)) \neq 0$. If $x(t_k) \in \mathcal{H}^{\Delta_3}(t_k) \cap \mathcal{Q}(t_k)$ and $x(t_{k+1}) \in \mathcal{H}^{\Delta_3}(t_{k+1}) \cap \mathcal{Q}(t_{k+1})$, then $x(t) \in \mathcal{Q}(t)$ for all $t \in [t_k, t_{k+1}]$. Moreover, if there exists a time $t_s \in (t_k, t_{k+1})$ at which $\dot{\kappa}(t_s, x(t_s)) = 0$, then t_s is unique.*

Proof. For brevity, in this proof, I only write out the time argument of ψ, κ, h, and their derivatives. By assumption, $x(t_k) \in \mathcal{Q}(t_k)$ and $x(t_{k+1}) \in \mathcal{Q}(t_{k+1})$, so the trajectory can only leave $\mathcal{Q}$ if there exists a local maximizer $t_s \in (t_k, t_{k+1})$ of κ such that $\kappa(t_s) > 0$. Thus, the rest of this proof proceeds by analyzing whether such a maximizer t_s can exist.

Note that on the open interval (t_k, t_{k+1}), the functions κ, $\dot{\kappa}$, and $\ddot{\kappa}$ are continuous in time, and therefore $\dot{h}$ in (5.51) is continuous as well. By assumption, σ is a maximizer of h, so it follows that $\dot{h}(\sigma) = 0$. Because of the form of $\dot{h}$ in (5.51), if $\dot{h}(\sigma) = 0$ and $\dot{\kappa}(\sigma) \neq 0$, it must be that $\ddot{\kappa}(\sigma) = -\mu \operatorname{sign}(\dot{\kappa}(\sigma))$. That is, for a critical point of h to occur at σ when $\dot{\kappa}(\sigma) \neq 0$, the second

derivative of κ (e.g. angular acceleration) at σ must pass through one of two critical values, $\pm\mu$, depending on the sign of $\dot{\kappa}(\sigma)$. This yields the two cases below.

First, suppose that $\dot{\kappa}(\sigma) < 0$, which implies $\ddot{\kappa}(\sigma) = \mu$. Thus, I derive an expression for $\ddot{\kappa}(t)$ for t in a neighborhood of σ. Let $\tau \geq 0$ and it follows from (5.50a) that

$$\ddot{\kappa}(\sigma + \tau) \geq \mu + M_2^- + M_3^- \tau - M_2^+ \,. \tag{5.66}$$

Because of the assumed lower bound on μ, (5.66) implies that $\ddot{\kappa}(\sigma + \tau) > 0$ for any $\tau \in (0, T)$. Similarly, it follows from (5.50d) that

$$\ddot{\kappa}(\sigma - \tau) \geq \mu + M_2^- - M_3^+ \tau - M_2^+ \,, \tag{5.67}$$

which implies $\ddot{\kappa}(\sigma - \tau) > 0$ for any $\tau \in (0, T)$. Thus, by (5.66) and (5.67), $\ddot{\kappa}(t) > 0$ for all $t \in (t_k, t_{k+1})$ and therefore there can be no local maximizers of κ on (t_k, t_{k+1}). Moreover, if there exists a time $t_s \in (t_k, t_{k+1})$ at which $\dot{\kappa}(t_s) = 0$, then t_s is unique and is a local minimizer of κ since $\ddot{\kappa}$ is strictly positive.

Second, suppose that $\dot{\kappa}(\sigma) > 0$, which implies $\ddot{\kappa}(\sigma) = -\mu$. It follows from (5.50c) that

$$\ddot{\kappa}(\sigma - \tau) \leq -\mu + M_2^+ - M_3^- \tau - M_2^- \,, \tag{5.68}$$

which implies that $\ddot{\kappa}(\sigma - \tau) < 0$ for all $\tau \in (0, T)$. Since $\dot{\kappa}(\sigma) > 0$ and $\ddot{\kappa}(\sigma - \tau) < 0$, it follows that $\dot{\kappa}(t) > 0$ for all $t \in [t_k, \sigma]$ and therefore there can be no local maximizers of κ on $[t_k, \sigma]$. Next, it follows from (5.50b) that

$$\ddot{\kappa}(\sigma + \tau) \leq -\mu + M_2^+ + M_3^+ \tau - M_2^- \,, \tag{5.69}$$

which again implies that $\ddot{\kappa}(\sigma + \tau) < 0$ for all $\tau \in (0, T)$, and thus by (5.68) and (5.69), $\ddot{\kappa}(t) < 0$ for all $t \in (t_k, t_{k+1})$. Thus, in this case, there may exist a local maximizer t_s of κ but only for $t_s \in (\sigma, t_{k+1}]$, such as shown by the green "x" in Fig. 5.12. If such a t_s exists, then $\dot{\kappa}(t_s) = 0$, and t_s is the unique maximizer of κ on $[t_k, t_{k+1}]$ since $\ddot{\kappa}$ is strictly negative. Going forward, I assume such a t_s exists and now seek to ascertain its value.

From here, there are several possible ways to ensure $\kappa(t) \leq 0$ for all $t \in [t_k, t_{k+1}]$, and I only present one method. In my approach, I now shift focus from the values of κ to the values of h. I will show that for the choice of Δ_3 in (5.88), it holds that $h(t) \leq 0$ for all $t \in [t_k, t_{k+1}]$ and conclude that $x(t) \in \mathcal{Q}(t)$ for all $t \in [t_k, t_{k+1}]$ by applying Lemma 5.10.

Since $\dot{h}(\sigma) = 0$, I am interested in how positive $\dot{h}(t)$ can be for $t \in [t_k, \sigma]$ and how negative $\dot{h}(t)$ can be for $t \in [\sigma, t_{k+1}]$, from which I can derive a margin that ensures that $h(\sigma)$ does not

exceed zero. To this end, it follows from (5.50d) that

$$\ddot{\kappa}(\sigma - \tau) \geq -\mu + M_2^- - M_3^+ \tau - M_2^+ \,. \tag{5.70}$$

Let $\dot{\kappa}(\sigma) = \gamma$ for $\gamma \in \mathbb{R}_{>0}$ at the maximizer σ of h, where the existence of $t_s \in (\sigma, t_{k+1})$ implies that γ is close to zero. First, for $t \in [t_k, \sigma]$, $\dot{\kappa}(t)$ is upper bounded as

$$\begin{aligned}
\dot{\kappa}(\sigma - \tau) &= \dot{\kappa}(\sigma) - \int_{\sigma-\tau}^{\sigma} \ddot{\kappa}(t) dt \\
&\overset{(5.70)}{\leq} \gamma - \int_{\sigma-\tau}^{\sigma} -\mu + M_2^- - M_2^+ - M_3^+ (\sigma - t) dt \\
&= \gamma + (\mu - M_2^- + M_2^+)\tau + \frac{1}{2} M_3^+ \tau^2 \,.
\end{aligned} \tag{5.71}$$

Since I showed that $\dot{\kappa}(t) > 0$ for all $t \in [t_k, \sigma]$, it follows that $\dot{h}(t)$ is upper bounded by

$$\begin{aligned}
\dot{h}(\sigma - \tau) &\overset{(5.51)}{=} \dot{\kappa}(\sigma - \tau) \left(1 + \frac{\ddot{\kappa}(\sigma - \tau)}{\mu} \right) \\
&\overset{(5.71),(5.68)}{\leq} \left(\gamma + (\mu - M_2^- + M_2^+)\tau + \frac{1}{2} M_3^+ \tau^2 \right) \left(\frac{M_2^+ - M_3^- \tau - M_2^-}{\mu} \right) \,.
\end{aligned} \tag{5.72}$$

Next, similar to (5.70), for $t \in [\sigma, t_{k+1}]$, $\dot{\kappa}(t)$ is upper bounded by

$$\begin{aligned}
\dot{\kappa}(\sigma + \tau) &= \dot{\kappa}(\sigma) + \int_{\sigma}^{\sigma+\tau} \ddot{\kappa}(t) dt \\
&\overset{(5.69)}{\leq} \gamma + \int_{\sigma}^{\sigma+\tau} -\mu + M_2^+ - M_2^- + M_3^+ (t - \sigma) dt \\
&= \gamma - (\mu + M_2^- - M_2^+)\tau + \frac{1}{2} M_3^+ \tau^2 \,.
\end{aligned} \tag{5.73}$$

I now divide the interval $[\sigma, t_{k+1}]$ into two intervals $[\sigma, t_s)$ and $(t_s, t_{k+1}]$ because, unlike the upper bound in (5.72), the lower bounds for $\dot{h}$ in (5.51) will be different before and after $\dot{\kappa}$ changes from positive to negative at t_s. For the interval $t \in [\sigma, t_s)$ where $\dot{\kappa}(t) > 0$, $\dot{h}(t)$ is minimized by minimizing $\ddot{\kappa}(t)$, so I note that (5.50a) lower bounds $\ddot{\kappa}(t)$ as

$$\ddot{\kappa}(\sigma + \tau) \geq -\mu + M_2^- + M_3^- \tau - M_2^+ \,. \tag{5.74}$$

Thus, during the interval $t \in [\sigma, t_s)$ where $\dot{\kappa}(t) > 0$, it follows that $\dot{h}(t)$ is lower bounded by

$$\dot{h}(\sigma + \tau) \overset{(5.51)}{=} \dot{\kappa}(\sigma + \tau) \left(1 + \frac{\ddot{\kappa}(\sigma - \tau)}{\mu} \right)$$

$$\overset{(5.73),(5.74)}{\geq} \left(\gamma - (\mu + M_2^- - M_2^+)\tau + \frac{1}{2}M_3^+\tau^2\right)\left(\frac{M_2^- + M_3^-\tau - M_2^+}{\mu}\right). \tag{5.75}$$

To take into account the interval $t \in (t_s, t_{k+1}]$ where $\dot{\kappa}(t) < 0$, instead of the upper bound (5.69), I will utilize the following lower bound for $\dot{\kappa}$

$$\begin{aligned}\dot{\kappa}(\sigma+\tau) &= \dot{\kappa}(\sigma) + \int_{\sigma}^{\sigma+\tau} \ddot{\kappa}(t)dt \\ &\overset{(5.74)}{\geq} \gamma + \int_{\sigma}^{\sigma+\tau} -\mu + M_2^- - M_2^+ + M_3^-(t-\sigma)dt \\ &= \gamma - (\mu - M_2^- + M_2^+)\tau + \frac{1}{2}M_3^-\tau^2\,. \end{aligned} \tag{5.76}$$

Finally, during the interval $t \in (t_s, t_{k+1}]$ where $\dot{\kappa}(t) < 0$, it follows that $\dot{h}(t)$ is lower bounded as

$$\begin{aligned}\dot{h}(\sigma+\tau) &\overset{(5.51)}{=} \dot{\kappa}(\sigma+\tau)\left(1 - \frac{\ddot{\kappa}(\sigma-\tau)}{\mu}\right) \\ &\overset{(5.76),(5.74)}{\geq} \left(\gamma - (\mu - M_2^- + M_2^+)\tau + \frac{1}{2}M_3^-\tau^2\right)\left(2 - \frac{M_2^- + M_3^-\tau - M_2^+}{\mu}\right). \end{aligned} \tag{5.77}$$

Next, given the bounds for $\dot{h}(t)$ in (5.72),(5.75),(5.77), I seek to upper bound $h(\sigma) - h(t_k)$ and $h(\sigma) - h(t_{k+1})$. Starting with the latter, it follows that

$$\begin{aligned}h(\sigma) - h(t_{k+1}) &= -\int_{\sigma}^{t_s} \dot{h}(t)dt - \int_{t_s}^{t_{k+1}} \dot{h}(t)dt \\ &\leq -\int_{\sigma}^{t_s} [\dot{h}(t) \text{ as in (5.75)}]dt - \int_{t_s}^{t_{k+1}} [\dot{h}(t) \text{ as in (5.77)}]dt \end{aligned} \tag{5.78}$$

where $t_s \in (\sigma, t_{k+1}]$ is an unknown parameter. Because of the disturbance, it is impossible to develop an exact expression for t_s as a function of γ (this is why τ_1, τ_2 will be free optimization parameters in (5.88)), but it is possible to develop lower and upper bounds on t_s for the computation of (5.78). Possible trajectories of $\dot{\kappa}(\sigma+\tau)$ are visualized in Fig. 5.13. At t_s, $\dot{\kappa}(t_s) = 0$, where $\dot{\kappa}$ is bounded by (5.73) and (5.76), so all candidate values of t_s must lie between the roots of the bounding functions (5.73) and (5.76). Since the lemma assumed that $M_3^- < 0$, the red line (5.76) is a concave downwards quadratic polynomial. Since $\dot{\kappa}(t) > 0$ for $t \in [\sigma, t_s)$, it follows that t_s must lie to the right of the second root of (5.76) (dashed red vertical line in Fig. 5.13), denoted $r_1 : \mathbb{R}_{\geq 0} \to \mathbb{R}$:

$$r_1(\gamma) \triangleq \frac{1}{M_3^-}\left[(\mu + M_2^- - M_2^+) - \sqrt{(\mu + M_2^- - M_2^+)^2 - 2M_3^-\gamma}\right]. \tag{5.79}$$

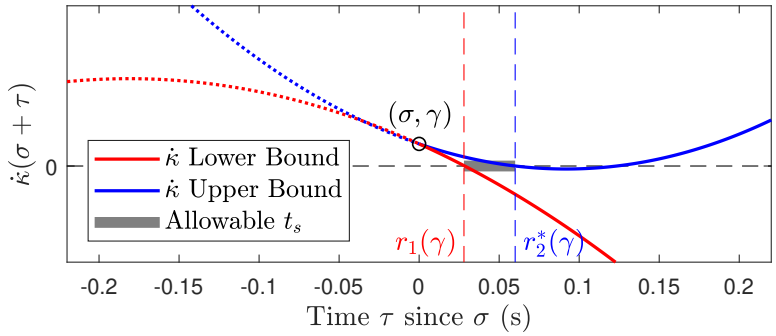

Figure 5.13: Visualization of Bounds on $\dot{\kappa}$. Visualization of the lower and upper bounds on $\dot{\kappa}(\sigma+\tau)$ imposed by (5.76) (red solid line) and (5.73) (blue solid line), respectively, and how this results in a finite interval of possible roots of $\dot{\kappa}$

That is, it holds that

$$t_s - \sigma \overset{(5.76)}{\geq} r_1(\gamma)\,. \tag{5.80}$$

Similarly, since the lemma assumed that $M_3^+ > 0$, the blue line (5.73) is a concave upwards quadratic polynomial, and since $\dot{\kappa}(t) < 0$ for $t \in (t_s, t_{k+1})$, it follows that t_s must lie to the left of the first root of (5.73) (dashed blue vertical line in Fig. 5.13), denoted $r_2^* : \mathbb{R}_{\geq 0} \to \mathbb{R}$:

$$r_2^*(\gamma) \triangleq \frac{1}{M_3^+}\left[(\mu - M_2^- + M_2^+) - \sqrt{(\mu - M_2^- + M_2^+)^2 - 2M_3^+\gamma}\right]. \tag{5.81}$$

The time t_s must also occur inside the present time step, so I define the bound $r_2 : \mathbb{R}_{\geq 0} \times \mathbb{R}_{\geq 0} \to \mathbb{R}$

$$r_2(\gamma,\tau) \triangleq \begin{cases} \min\{r_2^*(\gamma),\, T-\tau\} & \text{if } \gamma \leq (\mu - M_2^- + M_2^+)^2/(2M_3^+) \\ T-\tau & \text{else} \end{cases} \tag{5.82}$$

and conclude that

$$t_s - \sigma \overset{(5.73)}{\leq} r_2(\gamma, \sigma - t_k)\,. \tag{5.83}$$

The second case of (5.82) occurs when γ is such that (5.81) is nonreal, in which case t_s is instead upper bounded by the length of the time-step. Note that the bounds (5.79) and (5.82) greatly simplify the complete relationship between t_s and γ, but accounting for all possible curves of κ in a neighborhood of t_s would require more assumptions about the disturbance and would make this proof far more complex. Instead, I choose to treat t_s and γ as free parameters with minimal coupling to each other except that in (5.79) and (5.82), so that (5.78) is computable. I then simplify the right hand side of (5.78) and assign the result to the function $d_{right} : \mathbb{R}_{\geq 0} \times \mathbb{R}_{\geq 0} \times \mathbb{R}_{\geq 0} \to \mathbb{R}$

defined as

$$d_{right}(\gamma,\tau_1,\tau_2) \triangleq \frac{(M_3^-)^2}{8\mu}(\tau_1^4-\tau_2^4) - \frac{M_3^-M_3^+}{8\mu}\tau_2^4 + \left[\frac{M_3^-(M_2^- - M_2^+)}{2\mu} - \frac{2M_3^-}{3}\right]\left(\tau_1^3-\tau_2^3\right)$$
$$+ \frac{1}{6\mu}\left[2M_3^-(M_2^- - M_2^+ + \mu) - M_3^+(M_2^- - M_2^+)\right]\tau_2^3 - \frac{1}{\mu}(\gamma(M_2^- - M_2^+))\tau_2$$
$$+ \frac{1}{2\mu}\left[\left(M_2^+ - M_2^- + 2\mu\right)(M_2^+ - M_2^- + \mu) + M_3^-\gamma\right]\left(\tau_1^2-\tau_2^2\right)$$
$$+ \frac{1}{2\mu}\left[(M_2^- - M_2^+)(M_2^- - M_2^+ + \mu) - M_3^-\gamma\right]\tau_2^2 + \frac{\gamma}{\mu}(M_2^- - M_2^+ - 2\mu)(\tau_1-\tau_2) \quad (5.84)$$

so that (5.78) simplifies to

$$h(\sigma) - h(t_{k+1}) \leq d_{right}(\gamma, t_{k+1}-\sigma, t_s-\sigma) \quad (5.85)$$

where the third argument $t_s - \sigma$ is bounded by (5.80) and (5.83).

Similar to (5.78), $h(\sigma) - h(t_k)$ can be bounded as

$$h(\sigma) - h(t_k) = \int_{t_k}^{\sigma} \dot{h}(t) \leq \int_{t_k}^{\sigma} [\dot{h}(t) \text{ as in (5.72)}]dt = d_{left}(\gamma, \sigma - t_k) \quad (5.86)$$

where it is not necessary to break the integral into two parts here because $\dot{\kappa}(t)$ does not change signs on $[t_k, \sigma]$. I then define the function $d_{left} : \mathbb{R}_{\geq 0} \times \mathbb{R}_{\geq 0} \to \mathbb{R}$ as the simplification of the integral in (5.86) as follows

$$d_{left}(\gamma,\tau) \triangleq -\frac{\tau^4}{8\mu}(M_3^-M_3^+) - \frac{\tau^3}{6\mu}\left(2M_3^-(M_2^+ - M_2^- + \mu) + M_3^+(M_2^- - M_2^+)\right)$$
$$- \frac{\tau^2}{2\mu}\left((M_2^- - M_2^+)(M_2^+ - M_2^- + \mu) + M_3^-\gamma\right) - \frac{\gamma\tau}{\mu}(M_2^- - M_2^+)\,. \quad (5.87)$$

Both (5.85) and (5.86) must apply simultaneously, so I define Δ_3 below as a maximization of the lesser of d_{left} and d_{right}, subject to the constraints on $t_s - \sigma$ in (5.79) and (5.82). Furthermore, the maximizer σ of h must occur in the present time step, so $\sigma - t_k \in [0, T]$, and γ must be positive. Let $\tau_1 = \sigma - t_k$ and $\tau_2 = t_s - \sigma$, and finally define

$$\Delta_3 \triangleq \max_{\substack{\gamma\in[0,\infty)\\ \tau_1\in[0,T]\\ \tau_2\in[r_1(\gamma),r_2(\gamma,\tau_1)]}} \left(\min\left\{d_{left}(\gamma,\tau_1),\, d_{right}(\gamma, T-\tau_1, \tau_2)\right\}\right). \quad (5.88)$$

Note that although γ is not upper bounded in (5.88), in practice there is a maximum value of γ for which the interval $[r_1(\gamma), r_2(\gamma, \sigma - t_k)]$ is nonempty. Finally, using both bounds (5.85),(5.86), the

maximum value of h is bounded by

$$h(\sigma) \overset{(5.85),(5.86)}{\leq} \min\left\{h(t_k) + d_{left}(\gamma, \sigma - t_k),\ h(t_{k+1}) + d_{right}(\gamma, t_{k+1} - \sigma, t_s - \sigma)\right\}$$
$$\leq -\Delta_3 + \min\left\{d_{left}(\gamma, \sigma - t_k),\ d_{right}(\gamma, t_{k+1} - \sigma, t_s - \sigma)\right\} \overset{(5.88)}{\leq} 0 \quad (5.89)$$

so $x(t) \in \mathcal{H}(t)$ for all $t \in [t_k, t_{k+1}]$. By Lemma 5.10, $x(t) \in \mathcal{Q}(t)$ for all $t \in [t_k, t_{k+1}]$ too. In summary, I have shown that 1) when $\dot{\kappa}(\sigma) < 0$, no maximizer t_s of κ can occur, and 2) when $\dot{\kappa}(\sigma) > 0$, only one maximizer t_s of κ can occur and by (5.89) and Lemma 5.10, $\kappa(t_s) < 0$, so $x(t) \in \mathcal{Q}(t)$ for all $t \in [t_k, t_{k+1}]$ in all cases where $\dot{\kappa}(\sigma) \neq 0$. ■

Note that in Lemmas 5.11 and 5.14, I supposed existence of a local maximizer of h. If a local maximizer of h does not occur on $[t_k, t_{k+1}]$, then it is trivial to show that $x(t) \in \mathcal{H}(t)$ for all $t \in [t_k, t_{k+1}]$, and thus by Lemma 5.10, $x(t) \in \mathcal{Q}(t)$ for all $t \in [t_k, t_{k+1}]$. Also, it should be emphasized that Lemma 5.14 does not guarantee that $x(t) \in \mathcal{H}(t)$ for all $t \in [t_k, t_{k+1}]$ as in Theorems 5.12-5.13, because the value of Δ_3 required to guarantee that result could be larger. Rather, Lemma 5.14 only guarantees that $x(t)$ stays in the original constraint set $\mathcal{Q}(t)$ for all $t \in [t_k, t_{k+1}]$, and the proof further shows that $x(t)$ stays in $\mathcal{H}(t)$ in the special case where a local maximizer $t_s \in (t_k, t_{k+1})$ of κ also exists.

Case Study Part ix (Application of Lemma 5.14)**.** *Using the values of $M_2^-, M_2^+, M_3^-, M_3^+, \mu$ in Table 5.4, it follows that $\Delta_3 = 1.09(10)^{-5}$ in* (5.88)*. This occurs for $\gamma = 3.6(10)^{-6}$, $\sigma - t_k = 0.13$ s, and $t_s - \sigma = 0.0023$ s. In this case, $\Delta_3 < \delta_1$, but this is not guaranteed in general.*

Remark 5.3. *Note that the necessary condition $\ddot{\kappa}(\sigma, x(\sigma), u, \xi) = -\mu$ in the proof of Lemma 5.14 (preceding* (5.68)*) is very specific, so in my experience, maximizers of h meeting the conditions of Lemma 5.14 are rarer than maximizers meeting the conditions of Lemma 5.11. However, the conditions of Lemma 5.14 occur more frequently if μ is chosen very small.*

Thus, I have now identified bounds on the overshoot of κ between time steps both when κ and h share maximizers (Lemma 5.11) and when the maximizer of κ is distinct from the maximizer of h (Lemma 5.14). I thus have all the tools required to define the ZOH robust CBF set $\mathcal{Z}$. I now combine Theorem 5.13 and Lemma 5.14 to state my main theorem for relative-degree 2 constraints.

Theorem 5.15. *Suppose κ is thrice continuously differentiable and of relative-degree 2 with respect to* (5.30)*, and u is constant on $[t_k, t_{k+1})$ where $t_{k+1} = t_k + T$. Suppose that $M_3^+ > 0$, $M_3^- < 0$, and $\mu \geq M_2^+ - M_2^- + \max\{|M_3^+|, |M_3^-|\}T$. Suppose there exists δ_2 and Δ_2 satisfying condition* (5.64)*, and that $\Delta_2 \geq \Delta_3$ in* (5.88)*. If $x(t_k) \in \mathcal{H}^{\Delta_2}(t_k) \cap \mathcal{Q}(t_k)$ and $x(t_{k+1}) \in \mathcal{H}^{\Delta_2}(t_{k+1}) \cap \mathcal{Q}^{\delta_2}(t_{k+1})$, then $x(t) \in \mathcal{Q}(t)$ for all $t \in [t_k, t_{k+1}]$.*

Proof. For brevity, in this proof, I only write out the time argument of κ, h, and their derivatives. By assumption, $\kappa(t_k) \leq 0$, $h(t_k) \leq -\Delta_2 \leq 0$, $\kappa(t_{k+1}) \leq -\delta_2 \leq 0$, and $h(t_{k+1}) \leq -\Delta_2 \leq 0$. Thus, $x(t)$ can only exit $\mathcal{Q}$ if there is a local maximizer t_s of κ for $t_s \in (t_k, t_{k+1})$. As a result of Lemma 5.10, it is only possible for $\kappa(t_s) > 0$ to occur if there also exists a local maximizer σ of h for $\sigma \in (t_k, t_{k+1})$ such that $h(\sigma) > 0$, where it is possible that $\sigma = t_s$. Suppose the existence of both t_s and σ, where neither is necessarily unique. If there exists a maximizer σ of h such that $\dot{\kappa}(\sigma) \neq 0$, then Lemma 5.14 implies that t_s is unique and that $\kappa(t_s) \leq 0$.

Next, if every maximizer σ of h satisfies $\dot{\kappa}(\sigma) = 0$, then Theorem 5.13 implies that $\kappa(t_s) \leq 0$ for every t_s, where $t_s = \sigma$. Finally, if there is one or more maximizers σ_1 of h such that $\dot{\kappa}(\sigma_1) = 0$, and one maximizer σ_2 of h such that $\dot{\kappa}(\sigma_2) \neq 0$, then by the first paragraph, t_s is unique and $\kappa(t_s) \leq 0$, and it follows that σ_1 is unique and $\sigma_1 = t_s$. That is, the conditions presented so far do not preclude the possibility of the cases described in Lemma 5.11 and Lemma 5.14 both occurring in the same time step, but in this case, safety is ensured by Lemma 5.14 alone. Since $\kappa(t_s) \leq 0$ for every maximizer t_s of κ, it follows that $\kappa(t) \leq 0$ for all $t \in [t_k, t_{k+1}]$, and thus $x(t) \in \mathcal{Q}(t)$ for all $t \in [t_k, t_{k+1}]$. ■

It follows from Theorem 5.15 that the ZOH robust CBF set in Fig. 5.10 is

$$\mathcal{Z}(t_k) = \mathcal{Q}^{\delta_2}(t_k) \cap \mathcal{H}^{\Delta_2}(t_k) \cap \mathcal{H}^{\Delta_3}(t_k)\,. \tag{5.90}$$

Remark 5.4. *Note that δ_1 in* (5.54) *decreases with decreasing μ while Δ_3 in* (5.88) *tends to increase with decreasing μ. Although μ is a tunable variable, this tradeoff suggests that there is some minimum amount of margin required when using a ZOH controller, regardless of the choice of μ. Note that both δ_1 and Δ_3 decrease with decreasing T.*

5.2.3.5 Determining the set of safe controls

Now that I have thoroughly addressed the problem of overshoot between time steps, I seek a condition on u that guarantees $h(t_{k+1}, x(t_{k+1})) \leq -\Delta_2$ and $\kappa(t_{k+1}, x(t_{k+1})) \leq -\delta_2$ so that I can practically apply Theorem 5.15. Moreover, I seek a choice of parameters δ_2, Δ_2, μ such that this condition is always feasible with respect to the input constraints everywhere in $\mathcal{Z}$ in (5.90). To this end, define the following polynomials in τ:

$$p_\kappa(t, x(t), u(t), \tau) \triangleq \kappa(t, x(t)) + \dot{\kappa}(t, x(t))\tau + \frac{1}{2}\psi(t, x(t), u(t))\tau^2 + \frac{1}{2}M_2^+\tau^2 + \frac{1}{6}M_3^+\tau^3 \tag{5.91}$$

$$p_h(t, x(t), u(t), \tau) \triangleq p_\kappa(t, x(t), u(t), \tau) + \frac{1}{2\mu}\operatorname{ssq}\Big(\dot{\kappa}(t, x(t)) +$$

$$\psi(t, x(t), u(t))\tau + M_2^+\tau + \frac{1}{2}M_3^+\tau^2\Big) \quad (5.92)$$

which I will show represent upper bounds on $\kappa(t+\tau, x(t+\tau))$ and $h(t+\tau, x(t+\tau))$, respectively, given $x(t)$ and a ZOH $u(t)$. Here, $\mathrm{ssq}(\lambda) \triangleq \lambda|\lambda|$ for brevity.

Theorem 5.16. *Suppose κ is thrice continuously differentiable and of relative-degree 2 with respect to* (5.30)*, $M_3^-, M_3^+, \delta_2, \Delta_2, \Delta_3, \mu$ satisfy the conditions of Theorem 5.15, $\mathcal{Z}$ is as given in* (5.90)*, and u satisfies* (5.41) *for every $k \in \mathbb{Z}_{\geq 0}$. If $x(t_0) \in \mathcal{Z}(t_0)$ and*

$$p_\kappa(t_k, x(t_k), u(t_k, x(t_k)), T) \leq -\delta_2 \quad (5.93a)$$

$$p_h(t_k, x(t_k), u(t_k, x(t_k)), T) \leq -\Delta_2 \quad (5.93b)$$

both hold for every $k \in \mathbb{Z}_{\geq 0}$, then $x(t) \in \mathcal{Q}(t)$ for all $t \in \mathcal{T}$.

Proof. For brevity, I only write the time arguments of ψ, p_κ, p_h, κ, h, and their derivatives in this proof. First, note that the evolution of $\dot{\kappa}$ and κ between time steps can be upper bounded as follows:

$$\begin{aligned}
\dot{\kappa}(t_k+\tau) &= \dot{\kappa}(t_k) + \int_{t_k}^{t_k+\tau} \ddot{\kappa}(t)dt \\
&\overset{(5.49a)}{\leq} \int_{t_k}^{t_k+\tau} \psi(t_k) + M_2^+ + M_3^+(t-t_k)dt \\
&= \dot{\kappa}(t_k) + \psi(t_k)\tau + M_2^+\tau + \frac{1}{2}M_3^+\tau^2\,, \qquad (5.94) \\
\kappa(t_k+\tau) &= \kappa(t_k) + \int_{t_k}^{t_k+\tau} \dot{\kappa}(t)dt \\
&\overset{(5.94)}{\leq} \kappa(t_k) + \int_{t_k}^{t_k+\tau} \dot{\kappa}(t_k) + \left[\psi(t_k) + M_2^+ + \frac{1}{2}M_3^+(t-t_k)\right](t-t_k)dt \\
&= p_\kappa(t_k, \tau)\,. \qquad (5.95)
\end{aligned}$$

Thus, p_κ in (5.91) is an upper bound on $\kappa(t_k+\tau)$, and (5.94) is an upper bound on $\dot{\kappa}(t_k+\tau)$. Since h in (5.38) is monotonically increasing in both κ and $\dot{\kappa}$, it follows that p_h in (5.92) is an upper bound on $h(t_k+\tau)$. Since $t_{k+1} = t_k + T$, it follows that (5.93a) implies $x(t_{k+1}) \in \mathcal{Q}^{\delta_2}(t_{k+1})$ and (5.93b) implies $x(t_{k+1}) \in \mathcal{H}^{\Delta_2}(t_{k+1})$, or equivalently $x(t_{k+1}) \in \mathcal{Z}(t_{k+1})$. Since this holds for every $k \in \mathbb{Z}_{\geq 0}$, Theorem 5.15 implies that $x(t) \in \mathcal{Q}(t)$ for all $t \in \mathcal{T}$. ■

Note that while the upper bounds p_κ and p_h are valid for any $\tau \geq 0$, Theorem 5.16 only considers the values of p_κ and p_h at $\tau = T$, and thus relies on the analysis leading up to Theorem 5.15 (which was not dependent on p_κ, p_h) to guarantee that κ remains nonpositive between sampling

times, i.e. for $\tau \in (0, T)$. Based on Theorem 5.16, I conclude with the following definition of a CBF for ZOH applications.

Definition 5.3 (Degree 2 Zero-Order-Hold Control Barrier Function)**.** *For a thrice continuously differentiable constraint function κ_i, the function $h_i : \mathcal{T} \times \mathcal{X} \to \mathbb{R}$ in* (5.38) *with parameter μ is a* Degree 2 Zero-order-hold Control Barrier Function (D$_2$ZohCBF) *on the set $\mathcal{S}$ for time-step T if there exist constants δ_2, Δ_2 satisfying* (5.64) *and $\Delta_2 \geq \Delta_3$ in* (5.88) *such that*

$$\min_{u \in \mathcal{U}} \left(\max\left\{p_{\kappa_i}(t, x, u, T) + \delta_2,\ p_{h_i}(t, x, u, T) + \Delta_2\right\}\right) \leq 0,\ \forall x \in \mathcal{Z}_i(t) \cap \mathcal{S}(t), \forall t \in \mathcal{T} \quad (5.96)$$

where $\mathcal{Z}_i$, p_{κ_i}, and p_{h_i} are given in (5.90)*,* (5.91)*, and* (5.92)*, respectively.*

I revert to using the i indexing notation in Definition 5.3 for completeness (recall that this entire section, and thus Definition 5.3 too, are for one constraint at a time). Similar to (2.4),(4.5) with continuous control, (5.96) accounts for the allowable control set $\mathcal{U}$, so if h is a D$_2$ZohCBF, then the conditions (5.93) are feasible in the presence of input constraints for all $x(t) \in \mathcal{Z}(t), t \in \mathcal{T}$. Equivalently, if h is a D$_2$ZohCBF then the set $\mathcal{U}_{\text{zoh}}(t, x) \cap \mathcal{U}$ is nonempty for all $x \in \mathcal{Z}(t) \cap \mathcal{S}(t)$, $t \in \mathcal{T}$, where

$$\mathcal{U}_{\text{zoh}}(t, x) = \left\{u \in \mathbb{R}^m \mid p_\kappa(t, x, u, T) \leq -\delta_2 \text{ and } p_h(t, x, u, T) \leq -\Delta_2\right\}. \quad (5.97)$$

The only remaining component is to determine a valid triple $(\delta_2, \Delta_2, \mu)$. One such triple is $\delta_2 = \delta_1$ in (5.54), $\Delta_2 = \Delta_1$ where $\Delta_1 \triangleq \max\{\delta_1, \Delta_3\}$ in (5.54),(5.88), and $\mu = \mu^*$ as follows

$$\mu^*(\delta_2, \Delta_2) \triangleq \max_{\mu \in (0, \infty)} \mu \quad (5.98)$$
$$\text{such that} \max_{\substack{x \in \mathcal{Z}(t) \cap \mathcal{S}(t) \\ t \in \mathcal{T}}} \left(\min_{u \in \mathcal{U}} \left(\max\left\{p_\kappa(t, x, u, T) + \delta_2,\ p_h(t, x, u, T) + \Delta_2\right\}\right)\right) \leq 0$$

assuming μ^* exists. One can also choose $\mu \leq \mu^*(\delta_2, \Delta_2)$. Note that for large ξ or T, δ_2 and Δ_2 will also be large, and there may be no μ^* satisfying (5.98) and the conditions of Theorem 5.15, indicating that (5.30) cannot be safely controlled at such a sampling time T. A plot of μ^* using $\delta_2 = \delta_1$ and $\Delta_2 = \Delta_1$ for dynamics (5.31) is shown in Fig. 5.14, where the black region is where μ^* does not exist or is less than $M_2^+ - M_2^- + \max\{|M_3^+|, |M_3^-|\}T$. Note that the range of valid μ^* will also vary with $u_{\max}$, though that variation is not plotted here.

Case Study Part x (Selection of μ for Input Constraints)**.** *Using the values of $M_2^-, M_2^+, M_3^-, M_3^+$ in Table 5.4, the choice $(\delta_1, \delta_1, \mu^*(\delta_1, \delta_1))$ where $\mu^*(\delta_1, \delta_1) = 0.00167$ as in* (5.98) *is one valid triple. Alternatively, $(\delta_2, \Delta_2, \mu^*(\delta_2, \Delta_2))$ is another such triple. Note that*

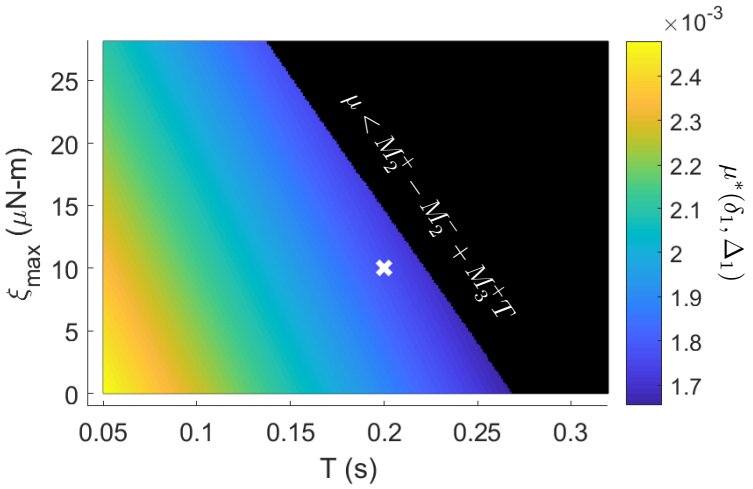

Figure 5.14: Plot of μ^* with Sample Time and Disturbance Bounds. Plot of μ^* in (5.98) variation with the disturbance bound $\xi_{\max}$ and sampling period T for the system (5.31), where 'x' marks the case study parameters. The black region represents μ that violate the assumptions of Theorem 5.15.

$\mu^*(\delta_2, \Delta_2)$ is slightly larger than $\mu^*(\delta_1, \delta_1)$, but the difference is only in the fourth significant digit of μ for this particular system. I observed a greater difference between $\mu^*(\delta_2, \Delta_2)$ and $\mu^*(\delta_1, \delta_1)$ for problems where $u_{\max}$ was greater. Thus, ZOH discretization has led to a more conservative result than the continuous-time case with $\mu = 0.0025$ in Case Study Part iii.

Note that the polynomial p_κ is linear in ψ, and therefore affine in u, so one can encode (5.93a) in a QP-based control law as in [66, Sec. II-C]. The polynomial p_h has nonlinear dependence on ψ (because of the ssq function), but p_h is still monotone increasing in ψ, and thus one can write $p_h \leq -\Delta_2$ in (5.93b) equivalently as $\psi \leq \psi_{\max}$ for some number $\psi_{\max}$ (the expression for $\psi_{\max}$ is excessively long and is omitted here, but can be easily derived; the interested reader may also refer to the function `get_PhiQ` in https://github.com/jbreeden-um/phd-code/blob/main/2022/AIAA%20Autonomous%20Attitude%20Reorientation/42/Source/42fsw.c). Thus, one can also encode (5.93b) in a QP, and the set $\mathcal{U}_{\text{zoh}}$ in (5.97) is a polytope. In conclusion, the sets $\mathcal{Z}$ in (5.90) and $\mathcal{U}_{\text{zoh}}$ in (5.97) solve the relative-degree 2 case of Problem 5.2.

5.2.4 Method for Relative Degree One

5.2.4.1 Preliminary Method

I now extend the method in Section 5.2.3 to constraint functions that are of relative-degree 1 with respect to the dynamics (5.30). As before, I drop the subscript i and assume that η represents any relative-degree 1 constraint function. In this subsection, I assume that η is also a CBF, so I will

not need to employ the intermediary step of defining h and $\mathcal{H}$, as was done for relative-degree 2 constraints. Similar to (5.46), define the function

$$\phi(t,x,u) \triangleq \frac{\partial \eta(t,x)}{\partial t} + \frac{\partial \eta(t,x)}{\partial v}\left(f_2(t,x) + g_1(t,x)u\right) , \tag{5.99}$$

which represents the component of $\dot{\eta}$ that is known to the controller. Likewise, define the constant

$$M_1 \triangleq \sup_{t\in\mathcal{T}, x\in\mathcal{S}(t), \xi\in\Xi} \frac{\partial \eta(t,x)}{\partial v} g_2(t,x)\xi \, , \tag{5.100}$$

which represents the uncertainty in $\dot{\eta}$ because of the unknown disturbance. It then holds that

$$\dot{\eta}(t,x,u) \leq \phi(t,x,u) + M_1 \, . \tag{5.101}$$

Next, to account for the ZOH controller, define

$$\begin{aligned} M_2 \triangleq \sup_{t\in\mathcal{T}, x\in\mathcal{S}(t), u\in\mathcal{U}, \xi\in\Xi} \Bigg[& \frac{\partial \phi(t,x,u)}{\partial t} + \frac{\partial \phi(t,x,u)}{\partial q} f_1(t,x) \\ & + \frac{\partial \phi(t,x,u)}{\partial v}\left(f_2(t,x) + g_1(t,x)u + g_2(t,x)\xi\right) \Bigg] \end{aligned} \tag{5.102}$$

so that for $\tau \geq 0$ it holds that

$$\phi(t+\tau, x(t+\tau), u) \leq \phi(t, x(t), u) + M_2 \tau \, . \tag{5.103}$$

In this subsection, I only require the upper bounds on (5.100) and (5.102), so I omit the superscripts $+$ and $-$ used in the prior section. Here, M_1 and M_2 are analogous to M_2^+ and M_3^+, respectively, from Section 5.2.3.2. Now define the following polynomial in τ

$$p_\eta(t,x,u,\tau) \triangleq \eta(t,x) + \phi(t,x,u)\tau + M_1\tau + \frac{1}{2}M_2\tau^2 \, . \tag{5.104}$$

which serves as an upper bound on the evolution of η and is employed in the following theorem.

Theorem 5.17. *Suppose η is twice continuously differentiable and of relative-degree 1 with respect to* (5.30) *and u satisfies* (5.41) *for every $k \in \mathbb{Z}_{\geq 0}$. Suppose $M_2 \geq 0$ in* (5.102)*. If $x(t_0) \in \mathcal{V}(t_0)$ and*

$$p_\eta(t_k, x(t_k), u(t_k, x(t_k)), T) \leq 0 \tag{5.105}$$

for every $k \in \mathbb{Z}_{\geq 0}$, then $x(t) \in \mathcal{V}(t)$ for all $t \in \mathcal{T}$.

Proof. For brevity, I only write the time arguments of ϕ, p_η, η, and their derivatives in this proof.

First, note that the evolution of η can be upper bounded between time steps as follows

$$\begin{aligned}\eta(t_k+\tau) &= \eta(t_k) + \int_{t_k}^{t_k+\tau}\left[\dot\eta(t_k) + \int_{t_k}^{\tau_1}\ddot\eta(t_k+\tau_2)d\tau_2\right]d\tau_1 \\ &\overset{(5.102)}{\leq} \eta(t_k) + \int_{t_k}^{t_k+\tau}\left[\dot\eta(t_k) + \int_{t_k}^{\tau_1} M_2 d\tau_2\right]d\tau_1 \\ &\overset{(5.100)}{\leq} \eta(t_k) + \int_{t_k}^{t_k+\tau}\left[\phi(t_k) + M_1 + M_2\tau_1\right]d\tau_1 \\ &\overset{(5.104)}{\leq} p_\eta(t_k,\tau)\,. \end{aligned} \tag{5.106}$$

It follows that if $p_\eta(t_k,\tau) \leq 0$, then $\eta(t_k+\tau) \leq 0$ and thus $x(t_k+\tau) \in \mathcal{V}(t_k+\tau)$. Note that p_η in (5.104) is a concave upwards quadratic in τ (since M_2 is assumed to be nonnegative), so if $p_\eta(t_k,0) = \eta(t_k) \leq 0$ and $p_\eta(t_k,T) \leq 0$, then $\eta(t_k+\tau) \leq p_\eta(t_k,\tau) \leq p_\eta(t_k,T) \leq 0$ for all $\tau \in [0,T]$. Since the theorem assumed $x(t_0) \in \mathcal{V}(t_0)$, or equivalently $p_\eta(t_0,0) = \eta(t_0) \leq 0$, and since (5.106) implies $p_\eta(t_k,T) \leq 0$ for all $k \in \mathbb{Z}_{\geq 0}$, it follows that $\eta(t) \leq 0$ for all $t \in \mathcal{T}$, or equivalently, $x(t) \in \mathcal{V}(t)$ for all $t \in \mathcal{T}$. ■

Note that Theorem 5.17 is a straightforward extension of Corollary 5.8 to systems with disturbances, while the insights in the following subsection are more novel and are motivated specifically by the system in (5.31).

Case Study Part xi (Application of Theorem 5.17)**.** *For the system* (5.31)*, in order for the constants* $M_2^-, M_2^+, M_3^-, M_3^+$ *for* κ_b *in* (5.34) *to be well defined (i.e. for* $\mathcal{S}$ *in* (5.45),(5.48) *to be compact), the maximum system angular velocity must be bounded. There are various ways to encode such a bound. First, if one desires that* $\|\omega\| \leq \omega_{\max}$ *for some* $\omega_{\max} \in \mathbb{R}_{>0}$*, then one could use either* $\eta_1(t,x) = \|\omega\| - \omega_{\max}$ *or* $\eta_2(t,x) = \|\omega\|^2 - \omega_{\max}^2$*. Note that* M_2 *is undefined for the constraint function* η_1*, so Theorem 5.17 does not apply. Instead, consider the function* η_2*. For* η_2 *with* $\omega_{\max} = 0.0175$ rad/s*, it follows that* $M_2 = 0.00153$*. While* η_2 *satisfies the definition of D*$_1$*ZohCBF, this leads to an effective margin of* $\left(\frac{1}{2}M_2T^2\right)/\omega_{\max}^2 \approx 10\%$*, which is rather large. While this does not directly impact the robust CBF set* $\mathcal{Z}$ *in* (5.113)*, this margin in effect makes certain states in the safe set inaccessible (see Definitions 5.1-5.2), so I would like to reduce this margin.*

Next, suppose the matrix Z *in* (5.31) *is of the form*

$$Z = \begin{bmatrix} Z_{11} & Z_{12} \\ Z_{21} & Z_{22} \end{bmatrix} \tag{5.107}$$

where $Z_{11} \in \mathbb{R}^{3\times 3}, Z_{12} \in \mathbb{R}^{3\times m}, Z_{21} \in \mathbb{R}^{m\times 3}, Z_{22} \in \mathbb{R}^{m\times m}$*. Note that, under the dynamics*

in (5.31)*, $\|\omega\|^2$ is not a conserved quantity, so if the spacecraft is not spinning about a principal axis, it will take active control effort to keep the state within a level set of η_2. On the other hand, kinetic energy is a conserved quantity, which takes no control effort to maintain (unless the disturbance adds energy to the system). For this reason, define P in η_ω in* (5.36) *as $P = Z_{11}^{-1}$, so that η_ω encodes a maximum kinetic energy constraint. Then, using η_ω in* (5.36) *with $e_{\max}$ in Table 5.3, one finds $M_2 = 8.30(10)^{-5}$, leading to a smaller controller margin of $\left(\frac{1}{2}M_2T^2\right)/e_{\max} \approx 3.3\%$.*

5.2.4.2 Reducing Conservatism

Before I present a definition for a valid CBF for the relative-degree 1 case, I present an extension of Theorem 5.17 that reduces conservatism for certain systems and constraint functions, and in particular the constraint function η_ω in the case study in (5.36). In developing this work, I noticed that the main contributor to M_2 for the constraint function in (5.36) was the control input u. While x and ξ are not known exactly between time steps t_k and t_{k+1}, the value of $u(t) = u(t_k)$ for $t \in (t_k, t_{k+1})$ is a known quantity, and thus can be removed from the uncertainty bound M_2. Motivated by this, suppose there exists functions $\phi_1 : \mathcal{U} \to \mathbb{R}_{\geq 0}$ and $\phi_2 : \mathcal{T} \times \mathcal{S} \times \mathcal{U} \times \Xi \to \mathbb{R}$ such that

$$\dot{\phi}(t, x, u, \xi) \equiv \phi_1(u) + \phi_2(t, x, u, \xi) \tag{5.108}$$

for all $t \in \mathcal{T}, x \in \mathcal{S}, u \in \mathcal{U}, \xi \in \Xi$. The value of $\phi_1(u)$ is known, so define a constant analogous to (5.102) using ϕ_2 only as

$$M_2^{\text{alt}} \triangleq \sup_{t \in \mathcal{T}, x \in \mathcal{S}(t), u \in \mathcal{U}, \xi \in \Xi} \phi_2(t, x, u, \xi)\,. \tag{5.109}$$

See also [125] for a related decomposition. Then define the polynomial

$$p_\eta^{\text{alt}}(t, x, u, \tau) \triangleq \eta(t, x) + \phi(t, x, u)\tau + M_1\tau + \frac{1}{2}\left(\phi_1(u) + M_2^{\text{alt}}\right)\tau^2\,. \tag{5.110}$$

Corollary 5.18. *Suppose η is twice continuously differentiable and of relative-degree 1 with respect to* (5.30) *and u satisfies* (5.41) *for all $k \in \mathbb{Z}_{\geq 0}$. Suppose ϕ_1 in* (5.108) *is positive semi-definite and $M_2^{\text{alt}} \geq 0$ in* (5.109)*. If $x(t_0) \in \mathcal{V}(t_0)$ and*

$$p_\eta^{\text{alt}}(t_k, x(t_k), u(t_k, x(t_k)), T) \leq 0 \tag{5.111}$$

for every $k \in \mathbb{Z}_{\geq 0}$, then $x(t) \in \mathcal{V}(t)$ for all $t \in \mathcal{T}$.

Proof. For brevity, I only write the time arguments of $\phi_1, p_\eta^{\text{alt}}, \eta$, and their derivatives in this proof.

Similar to (5.106), $p_\eta^{\text{alt}}(t_k, \tau)$ in (5.110) is an upper bound on $\eta(t_k + \tau)$. By (5.41), u is constant between time steps and therefore the quadratic coefficient of p_η^{alt} given by $\phi_1(u(t_k)) + M_2^{\text{alt}}$ is constant. Because ϕ_1 maps to $\mathbb{R}_{\geq 0}$ and $M_2^{\text{alt}} \geq 0$, the coefficient $\phi_1(u(t_k)) + M_2^{\text{alt}}$ is also nonnegative. Thus, $p_\eta^{\text{alt}}(t_k, \tau)$ is a concave upwards quadratic polynomial in τ, so if $\eta(t_k) \leq 0$ and $p_\eta^{\text{alt}}(t_k, T) \leq 0$, then it follows by the same logic as Theorem 5.17, that $x(t) \in \mathcal{V}(t)$ for all $t \in \mathcal{T}$. ■

I now give the complete requirements for a relative-degree 1 constraint function η to be a CBF in ZOH applications.

Definition 5.4 (Degree 1 Zero-Order-Hold Control Barrier Function)**.** *A twice continuously differentiable function η_i is a* Degree 1 Zero-order-hold Control Barrier Function (D_1ZohCBF) *on the set $\mathcal{S}$ for time-step T if there exists a positive semidefinite function $\phi_1 : \mathcal{U} \to \mathbb{R}_{\geq 0}$ and a function $\phi_2 : \mathcal{T} \times \mathcal{S} \times \mathcal{U} \times \Xi$ (where one can use $\phi_1(u) \equiv 0$) satisfying* (5.108) *such that*

$$\min_{u \in \mathcal{U}} p_{\eta_i}^{\text{alt}}(t, x, u, T) \leq 0, \ \forall x \in \mathcal{S}(t), \forall t \in \mathcal{T} \tag{5.112}$$

where $p_{\eta_i}^{\text{alt}}$ is as given in (5.110)*.*

That is, η is a D_1ZohCBF if the condition (5.105) is always feasible inside the safe set. Unlike Definition 5.3, Definition 5.4 does not contain any additional tuning parameters. I assume that the function η has already been constructed or tuned so as to be possible to render the corresponding set $\mathcal{V}$ forward invariant in the presence of input constraints. This is reasonable in the context of spacecraft attitude control, because the function η_ω in (5.36) represents spacecraft kinetic energy. A fundamental requirement of control design should be that the spacecraft is able to reduce its kinetic energy from any safe state. In math, this requirement is equivalent to (5.112) for η_ω. One case in which this requirement is not satisfied is if the spacecraft is allowed to achieve large angular velocities while operating at a control frequency too slow to stabilize the system. In this case, no amount of tuning will yield a safe controller, so (5.112) will be violated, and one will need to operate at lower angular velocities or smaller time-steps to achieve a stable system and satisfy (5.112).

For the D_1ZohCBF, denote

$$\mathcal{Z}(t) \equiv \mathcal{V}(t), \quad \mathcal{U}_{\text{zoh}}(t, x) = \left\{ u \in \mathbb{R}^m \mid p_\eta^{\text{alt}}(t, x, u, T) \leq 0 \right\}, \tag{5.113}$$

which solves the relative-degree 1 case of Problem 5.2. Note that if $\phi_1 \equiv 0$ in (5.108), then $\mathcal{U}_{\text{zoh}}$ in (5.113) is a half-space and safe control inputs can again be computed using a QP-based control law. Alternatively, if ϕ_1 is a convex function, then $\mathcal{U}_{\text{zoh}}$ in (5.113) is not necessarily a polytope, but will still be a convex set, allowing the use of other convex optimization tools to choose control

inputs. For instance, in Section 5.2.5.1, ϕ_1 will be a strictly convex quadratic function, yielding a quadratically constrained quadratic program (QCQP) as a control law.

Case Study Part xii (Application of Corollary 5.18)**.** *Suppose η_2 is as described in Case Study Part xi, and let $\phi_1(u) = 2u^{\mathrm{T}} Z_{12}^{\mathrm{T}} Z_{12} u$. This leads to $M_2^{\text{alt}} = 4.88(10)^{-4}$, resulting in an effective margin of $\left(\frac{1}{2} M_2^{\text{alt}} T^2\right) / w_{\max}^2 \approx 3.2\%$, much less than in the prior case with $\phi_1(u) \equiv 0$. Thus, when $\phi_1(u)$ is large, the controller still ends up applying the same amount of margin as in Case Study Part xi, but when $\phi_1(u)$ is small (i.e. u is small), the margin inherent in p_η^{alt} in* (5.110) *is reduced compared to the margin in p_η in* (5.104)*.*

Next, for η_ω with P as described in Case Study Part xi, let $\phi_1(u) = u^{\mathrm{T}} Z_{12}^{\mathrm{T}} P Z_{12} u$, resulting in $M_2^{\text{alt}} = 1.95(10)^{-5}$. This yields an effective margin of $\left(\frac{1}{2} M_2^{\text{alt}} T^2\right) / e_{\max} \approx 0.77\%$, and is therefore the setup used for simulation in Section 5.2.5.1.

5.2.5 Applications to Spacecraft Attitude Control

5.2.5.1 Simulations in MATLAB

In this section, I demonstrate the above methods in simulation. I assume a spacecraft with two instruments with boresight vectors b_1, b_2 and keep-out zones θ_1, θ_2 in Table 5.5, which induce two pointing constraint functions κ_1, κ_2 of the form in (5.34). Let $s_1 = s_2$ be the local sun vector, which is slowly time-varying. I then construct two D_2ZohCBFs h_1, h_2 as in Section 5.2.3 with the constants in Table 5.4. Suppose there is also an angular velocity constraint function η_3 of the form in (5.36) with the previously presented parameters in Table 5.3, and with $\phi_1(u) = u^{\mathrm{T}} Z_{12}^{\mathrm{T}} P Z_{12} u$ as discussed in Case Study Part xii. Then η_3 is a D_1ZohCBF. The set of safe control inputs is $\mathcal{U} \cap \mathcal{U}_{\text{zoh},1}(t,x) \cap \mathcal{U}_{\text{zoh},2}(t,x) \cap \mathcal{U}_{\text{zoh},3}(t,x)$.

Suppose the spacecraft (visualized in Fig. 5.9) is required to point instrument b_1 at inertially fixed target b_t, given in Table 5.5. Define the following shortest-path proportional-derivative control law

$$\varphi = \mathrm{sat}_{\varphi^*}\left(\arccos\left(b_t^{\mathrm{T}} R(q) b_1\right)\right), \tag{5.114a}$$

$$y = b_1 \times \left(R(q)^{\mathrm{T}} b_t\right), \tag{5.114b}$$

$$u_{\text{pd}}(t,x) = Z_{12}^{\dagger}\left(k_p \sin\left(\frac{\varphi}{2}\right)\frac{y}{\|y\|} - k_d \omega\right), \tag{5.114c}$$

where u_{pd} may be unsafe and does not necessarily satisfy the input constraints. Here, let sat be the saturation function and $Z_{12}^{\dagger}$ be the Moore-Penrose pseudoinverse. I then construct the final control

Parameter	Value
b_1	$[0.5774,\ 0.5774,\ 0.5774]^{\mathrm{T}}$
b_2	$[-0.8660,\ 0.5,\ 0]^{\mathrm{T}}$
b_t	$[0,\ -0.7072,\ -0.7072]^{\mathrm{T}}$
θ_1	$\pi/4$ rad
θ_2	$\pi/4$ rad
φ^*	0.2 rad
k_p	0.1
k_d	0.5
$q(t_0)$	$[0.5,\ 0.5,\ 0.5,\ 0.5]^{\mathrm{T}}$
$\omega(t_0)$	$[0,\ 0,\ 0]^{\mathrm{T}}$
$w(t_0)$	$[0,\ 0,\ 0,\ 0]$

Table 5.5: Satellite Reorientation Simulation Parameters. Simulation parameters for Section 5.2.5.1

law as a QCQP

$$u_{\text{zohcbf}} = \underset{u\in\mathcal{U}\cap\mathcal{U}_{\text{zoh},1}(t,x)\cap\mathcal{U}_{\text{zoh},2}(t,x)\cap\mathcal{U}_{\text{zoh},3}(t,x)}{\arg\min} \|u - u_{\text{pd}}(t,x)\|^2 . \tag{5.115}$$

Using this "ZohCBF" controller, I simulated a single reorientation maneuver with initial and final parameters given in Table 5.5, and in the presence of a random disturbance bounded by ξ_{max} in Table 5.3. For more details, I refer the interested reader to the simulation code at https://github.com/jbreeden-um/phd-code/tree/main/2022/AIAA%20Autonomous%20Attitude%20Reorientation/MATLAB. The simulation is short enough that this section does not presently consider momentum-management (i.e. ensuring w_i remains bounded for $i = 1, 2, 3, 4$). A diagram of the excluded pointing zone and the trajectories of the two instrument vectors is shown in Fig. 5.15, and a video of the reorientation in three dimensions is found at https://youtu.be/EVuyZ-06-1Y. The constraint values over the maneuver duration are shown in Fig. 5.16, and the control inputs are shown in Fig. 5.17. As expected, safety is maintained, and the control input constraints are always satisfied. The absolute value of the maximum value of η_3 in Fig. 5.16 is the *physical margin* explained in Definition 5.2. Both ZohCBF plots in Fig. 5.16 exhibit a physical margin, but the margin is only noticeable for the constraint η_3 without zooming in.

For comparison, I also simulated the controllers in 1) [27, Eq. 22], denoted "Log-B", with $\alpha = 0.75$, $\beta = 8$, $k_1 = 0.0165$; 2) [236, Eq. 17] denoted "SMC", with $k = 0.01$, $k_1 = 5015$,

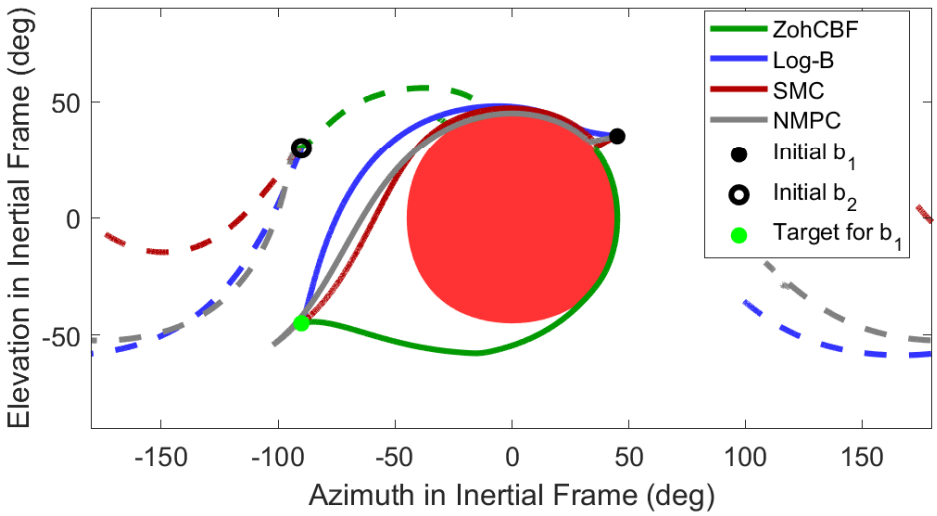

Figure 5.15: Satellite Reorientation Azimuth and Elevation. Plot of the azimuth and elevation in an inertial coordinate system of the two instrument pointing vectors b_1 (solid) and b_2 (dashed) and the keep-out zone (red) centered about the sun vector

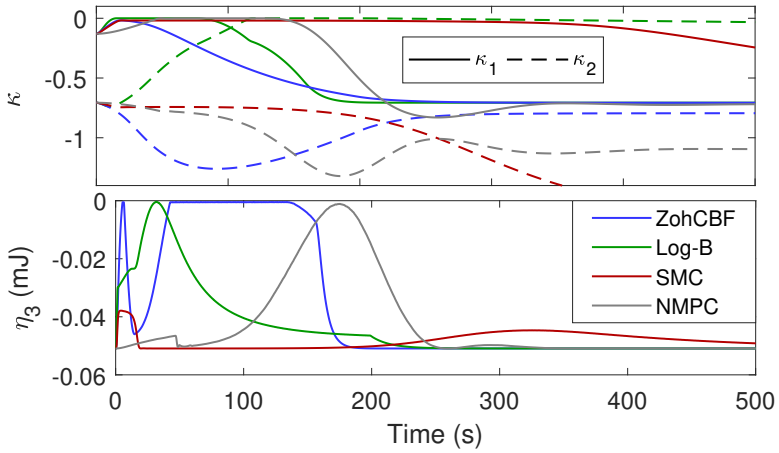

Figure 5.16: Satellite Reorientation Constraint Values. Plots of the two instrument constraint function values κ_1, κ_2 for the lines in Fig. 5.15, and the system energy constraint function values η_3 using all control laws

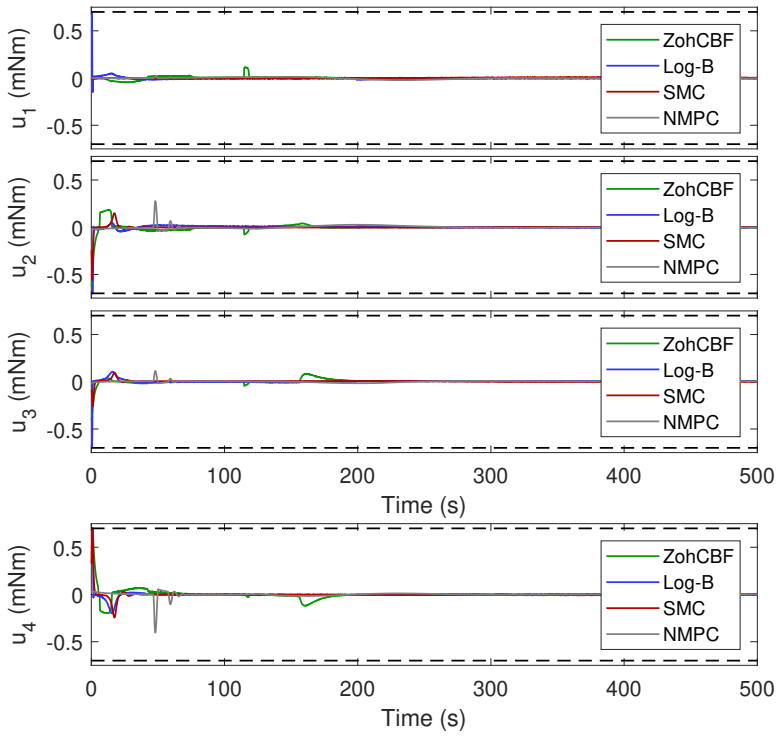

Figure 5.17: Satellite Reorientation Control Values. Plots of the control inputs and input constraints (black dashed lines) for Fig. 5.15 using all control laws

$k_2 = 0.0167$, $\hat{\bar{d}} \equiv 0$; and 3) [231], denoted "NMPC"[1], with $n = 5$, $h = 0.2$, $Q_1 = P_1 = 0.01I$, $Q_2 = P_2 = 38I$, $Q_3 = 100I$, where I is the identity matrix. The resultant trajectories are shown in Figs. 5.15-5.17 and described in Table 5.6, where all simulations were run on a 3.5 GHz Intel Xeon processor. While the Log-B and SMC controllers do not guarantee safety in the presence of input constraints or ZOH control inputs, Figs 5.15-5.17 show that when properly tuned, all of the above controllers can behave similarly. That said, the ZohCBF controller took a different route around the exclusion zone than all of the comparison controllers. The ZohCBF and NMPC controllers approached closer to the edge of the safe set than the Log-B and SMC controllers, and the NMPC controller briefly violated the κ_1 constraint. Also, the Lyapunov function introduced in [27] is infinitely continuously differentiable, so the trajectories under the Log-B controller are smooth. This is shown particularly in Fig. 5.16, where the green lines have unique maximizers, whereas the other controllers spend much of the trajectory very close to zero. This allowed the ZohCBF controller to achieve the fastest settling time, defined here as time to 0.1 degrees error, in Table 5.6. Note that for larger values of k_1, the Log-B controller could be faster but would exceed the angular velocity constraint, and for much larger values of k_1, the Log-B controller would violate

[1]I wish to thank Dr. Rohit Gupta of the Indian Institute of Technology Bombay for providing the optimization code used to simulate the "NMPC" controller. Without Dr. Gupta's assistance, I would not have been able to include this comparison.

Method	Settling Time (s)	Mean Compute Time (s)	Max Compute Time (s)
CBF	207.0	0.0088	0.021
Log-B	338.8	0.00022	0.0082
SMC	1719.8	0.00016	0.0087
NMPC	803.6	0.15	2.3

Table 5.6: Satellite Reorientation Simulation Times. Simulation times for Figs. 5.15-5.17

the pointing constraints due the ZOH implementation. The NMPC controller approached the target at a rate similar to the ZohCBF controller, but exhibited oscillations around the target due to small prediction horizon, thus resulting in a large settling time. The SMC controller was the slowest due to the upper bound on k implied by [236, Eq. 16].

Another notable difference between the ZohCBF and Log-B controllers is that the Lyapunov function in [27] is *strictly* convex, so the controller is globally convergent. This is not true of the ZohCBF or NMPC controllers. To examine this, I increased the value of θ_1, θ_2 to 0.95 rad and resimulated the ZohCBF and Log-B controllers. The results are shown in Figs. 5.18-5.19 and Table 5.7 and in the video at https://youtu.be/sZ_F4N75kcw. Note that the blue lines (ZohCBF controller) in Fig. 5.18 both approach the edge of the red region, and then stop when the controller cannot safely move closer to the target direction (green dot) due to the set $\mathcal{S} \cap \mathcal{Z}_1 \cap \mathcal{Z}_2 \cap \mathcal{Z}_3$ being nonconvex. The spacecraft remains safe, but does not complete its objective. On the other hand, the Log-B controller is eventually able to navigate around the exclusion zone and converge to the target vector. That said, the Log-B controller is very slow in Table 5.7. Lastly, I note that the ZohCBF technique can be applied to any nominal controller, so consider the control law

$$u_{\text{combined}} = \underset{u \in \mathcal{U} \cap \mathcal{U}_{\text{zoh},1}(t,x) \cap \mathcal{U}_{\text{zoh},2}(t,x) \cap \mathcal{U}_{\text{zoh},3}(t,x)}{\arg\min} \|u - u_{\text{logb-fast}}(t,x)\|^2 , \tag{5.116}$$

which I call the "Combined" controller. The controller $u_{\text{logb-fast}}$ in (5.116) is the same as the Log-B controller, but with a much more aggressive choice of gain $k_1 = 0.04$. Without the ZohCBF application, the controller $u_{\text{logb-fast}}$ would violate the system energy constraint η_3, but with the additional ZohCBF acting as a safety-filter, the controller u_{combined} yields the orange trajectory in Figs. 5.18-5.19. Unlike under u_{zohcbf}, the trajectory under u_{combined} converged to the target, and exhibited a reduced settling time in Table 5.7 compared to the Log-B controller.

Remark 5.5. *Note that while the controllers* (5.115),(5.116) *were successful in the simulations above, it still may be possible for the optimizations* (5.115),(5.116) *to become infeasible because these controllers apply multiple CBFs at once. Progress towards provably guaranteed feasibility of multiple CBFs simultaneously with input constraints is studied in continuous-time in Chapter 6, and such studies for sampled-data CBFs are left to future work.*

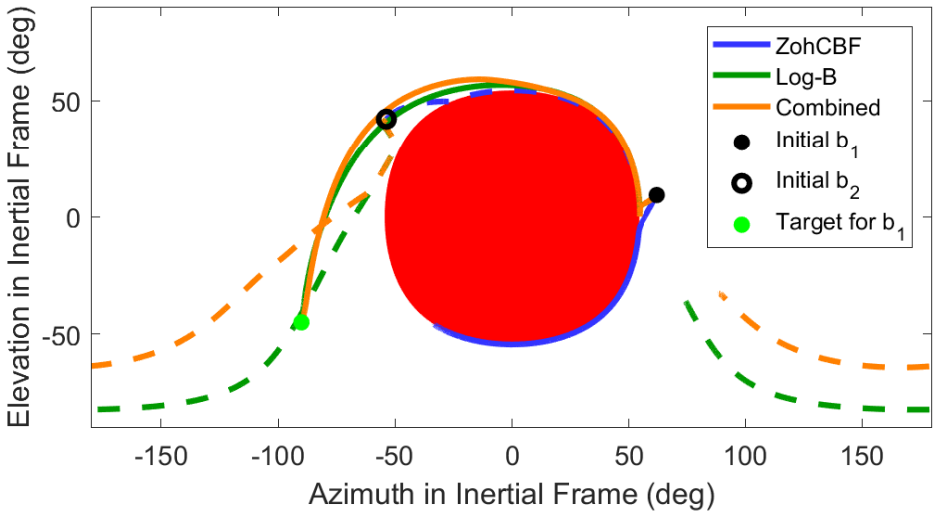

Figure 5.18: Nonconvex Reorientation Azimuth and Elevation. Plot of the azimuth and elevation in an inertial coordinate system of the two instrument pointing vectors b_1 (solid) and b_2 (dashed) in the presence of a larger exclusion zone than in Fig. 5.15

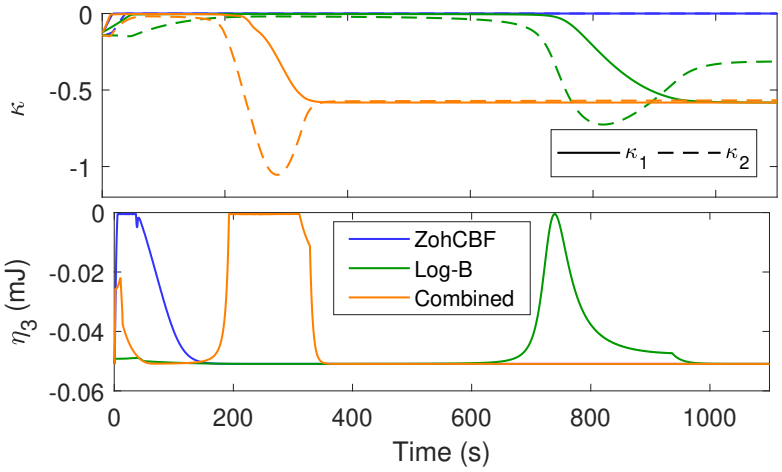

Figure 5.19: Nonconvex Reorientation Constraint Values. Plots of the two instrument constraint function values κ_1, κ_2 for the lines in Fig. 5.18, and the system energy constraint function values η_3 using all three control laws

Method	Settling Time (s)	Mean Compute Time (s)	Max Compute Time (s)
CBF	∞	0.0072	0.0237
Log-B	1079.6	0.00024	0.0084
Combined	374.2	0.0097	0.0243

Table 5.7: Nonconvex Reorientation Simulation Times. Simulation times for Figs. 5.18-5.19

5.2.5.2 NASA Simulator Results

The prior subsection validates the methods in Sections 5.2.3-5.2.4 in a simple simulation, so I now present results from a more detailed spacecraft simulator, specifically the NASA "42" open-source spacecraft attitude control simulator [243]. Here, rather than random disturbances, the disturbances are representative of disturbances in the orbital environment for an input spacecraft geometry and specified solar and geomagnetic activity indices.

Specifically, I simulated a 6U CubeSat with the parameters presented in Table 5.3 in a 500 km altitude circular Earth orbit. Suppose the spacecraft has a single instrument which must point at a sequence of targets but which must avoid the Sun by at least 25° (encoded in κ_1), and a star tracker which must not point at the Sun within 45° (encoded in κ_2) or the Moon within 30° (encoded in κ_3). The angular velocity is constrained by $\eta_4 = \eta_\omega$ as in Case Study Part xi, where I now use $\phi_1(u) = 0$, so that $M_2^{\text{alt}} = M_2 = 8.3(10)^{-5}$. This change makes $\mathcal{U}_{\text{zoh},4}$ more conservative than in the prior subsection, but makes $\mathcal{U} \cap \mathcal{U}_{\text{zoh},1} \cap \mathcal{U}_{\text{zoh},2} \cap \mathcal{U}_{\text{zoh},3} \cap \mathcal{U}_{\text{zoh},4}$ a polytope and thus changes (5.115) from a QCQP to a regular QP, which was implemented using the fast Operator Splitting QP solver [196]. Finally, the code limited the QP solver to only 20 solver iterations to mimic realistic spacecraft computing constraints. For more details and input parameters, the interested reader is referred to the simulation code https://github.com/jbreeden-um/phd-code/tree/main/2022/AIAA%20Autonomous%20Attitude%20Reorientation/42.

The instrument and star tracker pointing vectors are shown in Fig. 5.20, the constraint values are shown in Fig. 5.21, and the control inputs are shown in Fig. 5.22 using both u_{zohcbf} in (5.115) and u_{pd} in (5.114c). A video of the reorientation sequence is at https://youtu.be/qeB-F5J4ZFI. All constraints and actuator limits were satisfied for the entire pointing sequence using the ZohCBF controller (solid lines in Fig. 5.20), while there were several constraint violations using the nominal controller (dashed lines in Fig. 5.20). Note that three of the targets (green dots in Fig. 5.20) were located very close to the sun vector (i.e. outside the safe set), so the ZohCBF controller prioritized safety over convergence for these targets.

5.2.6 Conclusions

5.2.6.1 Remarks on Presented Results

The above work presented a methodology for ensuring that trajectories of a dynamical system always remain within a specified constraint set in the presence of ZOH sampled-data control inputs, bounded disturbances, and input constraints by developing extensions to CBF theory. This methodology is generally applicable to constraint functions of relative-degree 1 or 2, and was specialized to spacecraft attitude control. Formal definitions of "ZOH-CBF" incorporating all

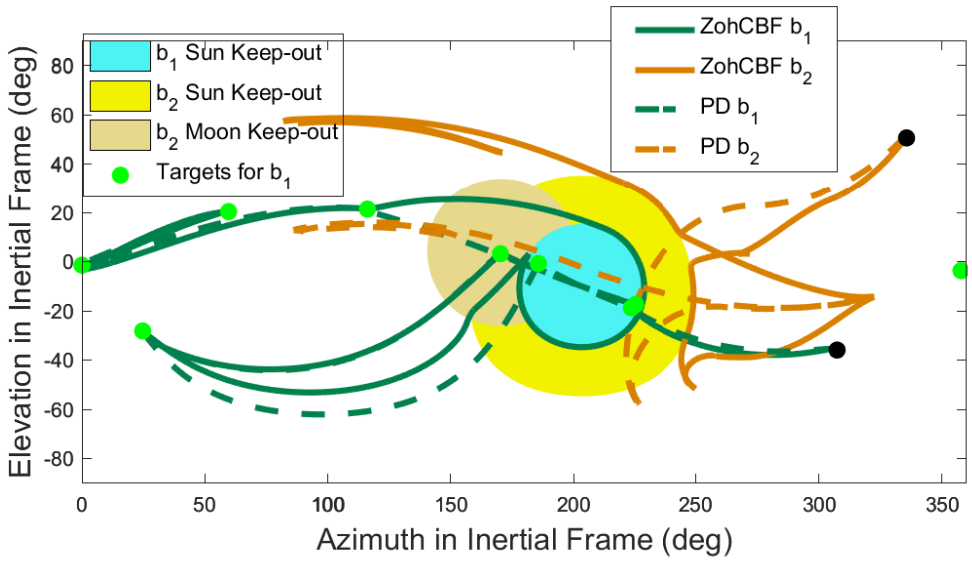

Figure 5.20: NASA Simulator Azimuth and Elevation. Plot of the azimuth and elevation in an inertial coordinate system of the instrument pointing vector b_1 (green) and star tracker pointing vector b_2 (orange) along with all three keep-out zones

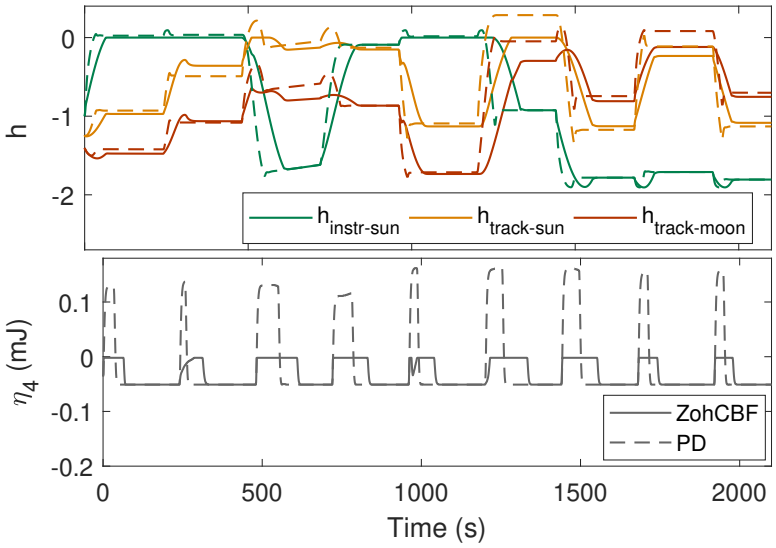

Figure 5.21: NASA Simulator CBF Values. Plots of the instrument to Sun CBF values (green), the star tracker to Sun CBF values (orange), and the star tracker to Moon CBF values (red) for the lines in Fig. 5.20, and the system energy constraint (gray) values η_4 using both control laws.

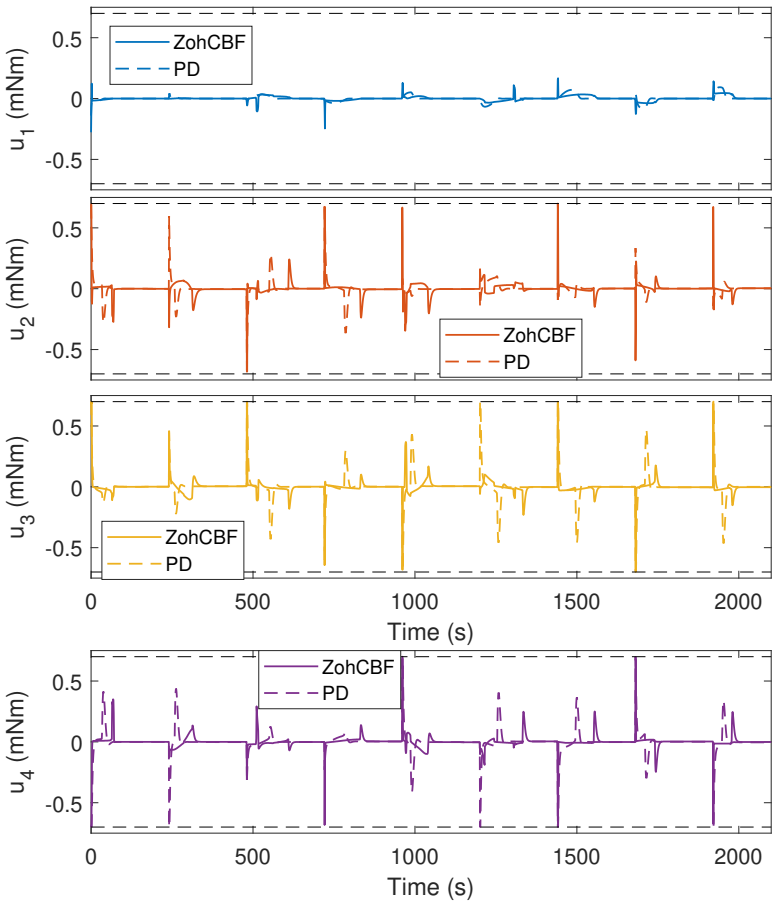

Figure 5.22: NASA Simulator Control Values. Plots of the control inputs and input constraints (black dashed lines) for Fig. 5.20 using both control laws

required margins and control input bounds were presented. Special attention was devoted to decreasing the margins for overshoot in the case of relative-degree 2 constraints, and for the case of a relative-degree 1 kinetic energy constraint specifically. The methodology was then demonstrated in simulation, where it exhibited faster settling times than all compared online controllers (note that path-planning methods were not tested). While these CBF-based methods provably achieve all desired safety criteria, the comparison plots show that similar safe reorientations can be achieved with the comparison methods, albeit only with careful tuning and without proof of safety under these circumstances. The improvement in convergence by the "combined" controller over the original ZohCBF controller show that this approach may be limited in part by the capabilities of the nominal control law, so choosing "optimal" nominal control laws is one area of future work. Additional future work includes the incorporation of momentum-management techniques and measurement-delay considerations, and study of more general conditions on the existence of a guaranteed safe control input in the presence of several CBFs simultaneously (see also Chapter 6).

5.2.6.2 Remarks on Future Sampled-Data Applications

While the above work provably achieved all desired results, this section showcased a lot of effort devoted to a very specific problem (i.e. the relative-degree 2 results are only for the CBF (5.38), not for other forms of CBFs). I now discuss some alternatives to the methodology above that I think may make these results easier to generalize in the future.

The first alternative result is the notion of "practical safety". This is partially discussed in [130] and also arose during a conversation with Andrew Taylor at the 2022 IEEE Conference on Decision and Control. Note that all the complicated math in Sections 5.2.3-5.2.4 in the end was used to produce a single number: the controller margin for a specified time-step T. This number took a substantial amount of calculus (or in Section 5.1, analysis of Lipschitz constants) to derive. Moreover, even after all these steps to reduce unnecessary margins, the constraint function trajectories in Figs. 5.16, 5.19, and 5.21 still never reach zero. This suggests that the controller margins produced by Sections 5.2.3-5.2.4 are still overly conservative. Thus, instead of going through all the tedious math above, one could instead guess what a suitable controller margin should be, run a simulation, and increase/decrease the controller margin according to the simulation results (i.e., increase the margin if the safe set is violated; decrease the controller margin if there is a large physical margin). For many systems, I predict that this method of guessing what a good controller margin will be and then tuning the margin as needed will be a more practical way to design ZOH control laws. That said, this technique will not allow for a proof of safety, so applications with extremely strict requirements may still prefer deriving bounds similar to what was done in this section. Going through this derivation may also yield other insights into system behaviors.

The second alternative result is ZOH techniques applied to High Order CBFs (HOCBFs). In the HOCBF technique [87–89], one starts with a constraint function $\kappa : \mathcal{T} \times \mathcal{X} \to \mathbb{R}$ and derives a sequence of functions

$$\psi_0(t,x) = \kappa(t,x) \tag{5.117a}$$
$$\psi_1(t,x) = \dot{\kappa}(t,x) - \alpha_1(-\psi_0(t,x)) \tag{5.117b}$$
$$\psi_2(t,x) = \ddot{\kappa}(t,x) - \alpha_2(-\psi_1(t,x)) \tag{5.117c}$$
$$\vdots$$
$$\psi_{r-1}(t,x) = \kappa^{(r-2)}(t,x) - \alpha_{r-1}(-\psi_{r-2}(t,x)) \tag{5.117d}$$

and associated sets

$$\Psi_i(t) = \{x \in \mathcal{X} \mid \psi_i(t,x) \leq 0\}, \tag{5.118}$$

where r is the relative-degree of κ (note that my indexing may differ from that in [87–89]). One can then treat ψ_{r-1} as a conventional CBF on the set $\mathbf{\Psi} = \Psi_0 \cap \Psi_1 \cap \cdots \cap \Psi_{r-1}$ (analogous to $\mathcal{H}_{\text{res}}$), and the condition $\dot{\psi}_{r-1} \leq \alpha(-\psi_{r-1}(t,x))$ is sufficient to render $\mathbf{\Psi}$ forward invariant [88, Thm. 4]. Thus, one could use a similar ZOH analysis of the derivatives and/or Lipschitz constants of ψ_{r-1} as in Sections 5.1.3 and 5.2.4 to render a subset $\mathbf{\Psi}$ (analogous to $\mathcal{H}$) of $\mathcal{Q}$ forward invariant for the high-relative degree function κ, possibly without all the complicated steps followed in Section 5.2.3. To my knowledge, no one has yet published work on generic ZOH applications of ψ_{r-1} in this manner, but this is a natural extension of the HOCBF method.

I mention this natural extension both 1) because I predict it will be used more frequently than the results in Section 5.2.3, as those results are specific to the CBF (5.38), and 2) because I want to highlight one of the deficiencies of this hypothetical approach that is also illustrative of other CBF phenomena. As mentioned in Section 4.3.2, the CBF (5.38) is equivalent to the HOCBF

$$\psi_{r-1}(t,x) = \psi_1(t,x) = \dot{\kappa}(t,x) - \sqrt{-2\mu\kappa(t,x)}\,. \tag{5.119}$$

However, the square root function is not differentiable or Lipschitz continuous, which violates the assumptions of [87–89]. Moreover, because the square root function is not differentiable, if one treats ψ_{r-1} in (5.119) as a relative-degree 1 constraint function, then the quantity M_2 in (5.102) is undefined. Thus, it appears that (5.119) is not compatible with the math in Sections 5.1.3 and 5.2.4, yet I have already shown in Section 5.2.3 that a subset of $\mathbf{\Psi}$ with ψ_{r-1} as in (5.119) can be rendered forward invariant with a ZOH control law. In light of this, I hypothesize that there is a relationship between the fact that (5.119) is non-differentiable, and the fact that the derivative of (5.38) vanishes at $\dot{\kappa} = 0$. This relationship would be interesting to explore in the future, as a more complete

understanding of it may allow for the design of better HOCBFs. Section 5.2.3 circumvented the problem of $\dot{\kappa}$ vanishing by using a quadratic approximation of the system evolution, which then motivated introducing additional (physical) margins $\delta_1, \Delta_2, \Delta_3$. These margins represent a maximum amount of deviation that can occur over a single time-step, whereas the prior methods relied on upper bounding the maximum rate of deviation that can occur at any instant, which is infinite. Similarly, I hypothesize that the non-differentiability of ψ_{r-1} can be circumvented by using a more precise approximation method (i.e. one that avoids the non-differentiable point in the square root function), likely involving a related margin applied to one or more of the sets (5.118). What this margin might look like is not considered further in this thesis, but I think that this is an excellent area for future work. Note that this non-differentiability issue does not occur with the most common form of HOCBF, the Exponential CBF (ECBF), also discussed in Section 4.3.2. However, the ECBF results in a more conservative CBF set than the CBF in (5.38) because the ECBF is always linear in $\dot{\kappa}$, whereas (5.38) is quadratic in $\dot{\kappa}$.

The above remarks conclude the discussion of ZOH control laws in this dissertation. The following section discusses a related variety of sampled control law, the impulsive control law.

5.3 Sampling and Impulsive Control Laws

Now that I have thoroughly discussed implementing CBF-based QP controllers with zero-order-hold dynamics, I now consider the related problem of implementing CBF-based control laws with impulsive dynamics and with a minimum time between impulses. I call this the "impulsive timed CBF problem". As I will show, the philosophy behind the following method is nearly identical to the philosophy used for the zero-order-hold controllers (i.e. use Lipschitz constants to guarantee that the trajectory does not leave the CBF set between control updates). The primary differences are 1) the actuators are impulsive rather than continuously applied, and 2) the time between control impulses is much larger than the time between zero-order-hold control updates. This second difference motivates the use of additional predictions to reduce the amount of uncertainty arising from the Lipschitz constants. The following work then concludes by revisiting a spacecraft docking problem similar to that in Section 4.7.4, now with impulsive actuators.

5.3.1 Introduction to Control Barrier Functions for Hybrid Systems

Control Barrier Functions (CBFs) [66] are a tool for designing control laws that render state trajectories always inside a specified set. Each CBF converts a set of allowable states, herein called the *CBF set*, to a set of allowable control inputs at every state in that set [140]. Any control input within this set will render the future state trajectory inside the CBF set. The controller thus has

freedom to work towards other goals, such as convergence, as long as the control remains within the input set generated by the CBF. CBFs thus provide a computationally tractable solution to many nonlinear constrained control problems. While the original formulations of CBFs [28, 29, 72] considered continuous-time systems, subsequent authors have published numerous extensions to sampled systems [119, 244–246], discrete-time systems [30, 128], and hybrid systems [33, 131–133], among others. In this section, I develop set invariance rules for a specific class of hybrid systems: systems with impulsive actuators that are only permitted to be used after a minimum *dwell time* has elapsed since their previous use. This models, for instance, a spacecraft with chemical thrusters.

Impulsive systems are a special class of hybrid systems, and there has been much work on stability of hybrid systems over the past two decades [247–252], and more recently work on set invariance [32, 72, 133] and CBFs [33, 131, 132, 134, 135] for hybrid systems. A hybrid system is a combination of a set of time intervals where a system *flows* according to a state differential equation called the *flow map*, and a set of times where the state *jumps* (changes instantaneously) according to an algebraic function called the *jump map*. Control may be applied along the flows, at the jumps (also called impulses), or both. In this section, I study systems where control occurs only via jumps, and jumps occur only when control is applied, as is formalized in Section 5.3.2. The work in [32, 133] show that a hybrid system renders a set forward invariant if 1) the flow map always lies within the tangent cone of the set, and 2) the image of the set through the jump map is a subset of the set. The authors in [33, 72, 132] then rewrite these conditions for CBFs and CBF sets. However, these two conditions have no way to incorporate a minimum dwell time constraint (equivalently, a minimum time between events [251]).

Recall that the problem of finding a CBF is equivalent to the problem of finding a controlled-invariant set. For hybrid systems, this equivalency follows from, e.g., [33, Def. 3.6] and [135, Def. 5]. In the following work, due to the minimum dwell time constraint, rather than applying control to render the state always inside such a controlled-invariant set, one must apply control so as to render the state into a set whose forward reachable set remains a subset of the CBF set at least until the dwell time has elapsed. This is an inherently different problem than that addressed by typical CBFs [29, 66] or by the hybrid CBFs in [32, 33, 72, 131–135], and has more in common with margins for ensuring set invariance between samples under sampled controllers such as [119, 125, 130, 244, 245] and Sections 5.1-5.2. This section applies the same concept of sampling margins as in the prior two sections, now modified for impulsive rather than zero-order-hold control, to guarantee set invariance under a minimum dwell time. Additionally, in Section 5.3.3.6, I propose a variation of this method for reducing conservatism, particularly for long dwell times.

Finally, the addition of the minimum dwell time constraint also complicates existing stability arguments. The work in [247] provides a formula for a maximum dwell time at which stability is still

guaranteed, and similar stability certificates for specified dwell times are presented in [250–252]. However, all of these results are overly restrictive, because they all place weak assumptions (e.g., exponentially bounded divergence) on the flows in exchange for strong requirements (e.g., rapid exponential contractivity) on the jumps. This is sensible in general, since the jumps are controlled and the flows are uncontrolled, but the spacecraft community has long developed controllers with weaker assumptions on the jumps [21, 25, 253], though not with the desired minimum dwell time. Thus, building up from the minimum dwell time constraint, the following work presents conditions to

1. render a CBF set forward invariant subject to impulsive control with a minimum dwell time constraint, and

2. render the origin asymptotically stable subject to the same impulsive control and dwell time rules.

The following subsections are organized as follows. Section 5.3.2 presents the hybrid system model. Section 5.3.3 presents the set invariance strategy for this system, new results on asymptotic stability, and some mathematical tools. Section 5.3.4 presents simulations of these methods on a satellite docking problem with impulsive thrusters. Section 5.3.5 presents concluding remarks.

5.3.2 Preliminaries

5.3.2.1 Notations

As in Chapter 4, for a function $\varphi : \mathcal{T} \times \mathcal{X} \to \mathbb{R}$, let $\partial_t \varphi$ denote the partial derivative with respect to t, let $\nabla \varphi$ denote the gradient row vector with respect to x, and let $\dot{\varphi} = \partial_t \varphi + \nabla \varphi \dot{x}$ denote the total derivative of φ in time along the model (5.120). The following work considers hybrid systems, so let x^+ denote the value of x immediately following a jump in a hybrid system. That is, $x^+(t) \triangleq \lim_{\lambda \to t^+} x(\lambda)$. Also consider the following generalization of class-$\mathcal{K}$ functions.

Definition 5.5 (Class-$\mathcal{K}_r$)**.** *A continuous function $\beta : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is said to belong to class-$\mathcal{K}_r$, denoted $\beta \in \mathcal{K}_r$, if $\beta(0) = 0$ and $\beta(\lambda) > 0$ for all $\lambda > 0$.*

5.3.2.2 Impulsive Model and Problem Statement

Spacecraft with chemical thrusters are frequently modeled as evolving according to an ordinary differential equation (ODE) with impulsive jumps. When activated, the thruster subsystem causes an instantaneous change, called an impulse, to the spacecraft velocity, and then the system flows according to the ODE until the next impulse is applied. Control can only be applied at the impulses,

as here I assume that there are no other actuators capable of applying control during the flows. I also assume the following two restrictions on impulses:

R-1 the controller is sampled with fixed period Δt and an impulse can only be applied at the sample times; and

R-2 the controller can only apply an impulse at least ΔT after the last impulse was applied, where $\Delta T > \Delta t$.

Let $\mathcal{T} \subseteq \mathbb{R}$ be a set of considered times, $\mathcal{X} \subseteq \mathbb{R}^n$ be the state space, and $\mathcal{U} \subseteq \mathbb{R}^m$ be the set of allowable controls. To encode R-1, let

$$\mathcal{D}_0 \triangleq \{t \in \mathcal{T} \mid t = t_0 + k\Delta t, k \in \mathbb{Z}_{\geq 0}\} \tag{5.120a}$$

be the set of controller sample times originating from an initial time $t_0 \in \mathcal{T}$. To encode R-2, let the additional state $\sigma \in \mathbb{R}_{\geq 0}$ encode the time since the last impulse was applied. A tuple (t, σ) is an *impulse opportunity* if $t \in \mathcal{D}_0$ and $\sigma \geq \Delta T$, or equivalently, if (t, σ) lies in the set of impulse opportunities

$$\mathcal{D} \triangleq \mathcal{D}_0 \times \{\sigma \in \mathbb{R}_{\geq 0} \mid \sigma \geq \Delta T\}. \tag{5.120b}$$

The control is thus a map $u : \mathcal{D} \times \mathcal{X} \to \mathcal{U}$ defined only at the set of impulse opportunities $\mathcal{D}$. The time ΔT is called the minimum *dwell time* between impulses [247]. For simplicity, assume that $\Delta T = q\Delta t$ for some $q \in \mathbb{Z}_{\geq 1}$.

Such a spacecraft can thus be modelled generally as

$$\begin{cases} \begin{cases} \dot{x} = f(t, x) \\ \dot{\sigma} = 1 \end{cases} & (t, \sigma) \notin \mathcal{D} \\ \begin{cases} x^+ = g(t, x, u) \\ \sigma^+ = \sigma \text{ if } u = 0 \\ \sigma^+ = 0 \text{ if } u \neq 0 \end{cases} & (t, \sigma) \in \mathcal{D} \end{cases} \tag{5.120c}$$

The system (5.120c) defines a hybrid system with flow set $\mathcal{C} \triangleq (\mathcal{T} \times \mathbb{R}_{\geq 0}) \setminus \mathcal{D}$, flow map $f : \mathcal{T} \times \mathcal{X} \to \mathbb{R}^n$, jump set $\mathcal{D}$, and jump map $g : \mathcal{T} \times \mathcal{X} \times \mathcal{U} \to \mathcal{X}$. I note that (5.120c) has time-dependent jumps, and therefore is also a timed automaton [254]. In the following work, I assume that the maps f and g are known and single-valued (rather than being differential inclusions), that $g(t, x, 0) = x$ for all $t \in \mathcal{T}, x \in \mathcal{X}$, and that solutions to (5.120) exist and are unique for all $t \in \mathcal{T}$. Also assume that $\sigma(t_0) = \Delta T$ at the initial time t_0, so that the initial state tuple $(t_0, \sigma(t_0), x(t_0))$ is an impulse opportunity. Note that, strictly speaking, both the flow and jump maps of a hybrid

system as in (5.120c) should be closed sets. I ignore this detail here because this section avoids discussion of some of the more complex phenomena that can occur in hybrid systems, so the meaning of (5.120c) is clearer as is, even if not entirely mathematically correct.

Note that at every impulse opportunity $(t, \sigma) \in \mathcal{D}$, the controller u can choose whether or not to apply an impulse, so impulses will generally be aperiodic, and may lack an average dwell time as in [247]. For brevity in Section 5.3.3, given a control law $u : \mathcal{D} \times \mathcal{X} \to \mathcal{U}$, denote the set of impulse opportunities where the control law chooses to not apply an impulse as

$$\mathcal{Z}_{\text{coast}} \triangleq \{(t, \sigma, x) \in \mathcal{D} \times \mathcal{X} \mid u(t, \sigma, x) = 0\}. \tag{5.121}$$

The central problem addressed in Section 5.3.3 is as follows.

Problem 5.3. *Given dynamics* (5.120) *and a set* $\mathcal{S} : \mathcal{T} \to 2^{\mathcal{X}}$*, derive conditions on the control* $u : \mathcal{D} \times \mathcal{X} \to \mathcal{U}$ *that are sufficient to 1) guarantee* $x(t)$ *remains in* $\mathcal{S}(t), \forall t \in \mathcal{T}$*, and 2) render the origin asymptotically stable, where I assume* $0 \in \mathcal{S}(t), \forall t \in \mathcal{T}$*.*

The conditions arising from Problem 5.3 can then be enforced online using optimization-based control laws as is typical in the CBF literature [66, Sec. II-C]. Unlike the rest of this dissertation, in this section I allow these optimizations to be nonlinear programs. Such programs are more computationally expensive than the quadratic programs in Section 2.4, but I assume that this cost is acceptable because of the long dwell time ΔT between impulses.

5.3.3 Impulsive Timed Control Barrier Functions and Control Lyapunov Functions

In this section, I first present some definitions and tools in Section 5.3.3.1, before using these tools to address invariance of a subset of $\mathcal{S}(t)$ in Section 5.3.3.2. I then address stability of the origin in two parts in Sections 5.3.3.3-5.3.3.4, and provide examples and additional tools in Section 5.3.3.5.

5.3.3.1 Flows and Bounding Functions

In the following work, I will utilize predictions about the future state. Suppose that no jumps occur in some interval $[t, \tau] \subset \mathcal{T}$. Then define the flow operator $p : \mathcal{T} \times \mathcal{T} \times \mathcal{X} \to \mathcal{X}$ (similar to χ in (4.31)) as

$$p(\tau, t, x) = y(\tau) \text{ where } \dot{y}(s) = f(s, y(s)),\ y(t) = x. \tag{5.122}$$

Next, given a scalar function $h : \mathcal{T} \times \mathcal{X} \to \mathbb{R}$, and an initial state (t, x), denote by ψ_h :

$\mathcal{T} \times \mathcal{T} \times \mathcal{X} \to \mathbb{R}$ any function satisfying

$$\psi_h(\tau, t, x) \geq h(s, p(s, t, x)), \ \forall s \in [t, \tau]\,. \tag{5.123}$$

That is, ψ_h is an upper bound on the future evolution of the function h for any interval $[t, \tau]$ during which there are no control impulses. Methods to find such a bounding function are described in Sections 5.1-5.2 and in [119, 125, 130, 244], among others, and thus these methods are only briefly elaborated upon here in Section 5.3.3.5. Note that the above references all include a term that accounts for the effects of the control input $u \in \mathcal{U}$, whereas this term can be ignored here since f in (5.120) is independent of u.

5.3.3.2 Set Invariance

I first address the safety part of Problem 5.3. As is typical with CBFs, this will require a function $h : \mathcal{T} \times \mathcal{X} \to \mathbb{R}$ such that the set

$$\mathcal{H}(t) \triangleq \{x \in \mathcal{X} \mid h(t, x) \leq 0\} \tag{5.124}$$

satisfies $\mathcal{H}(t) \subseteq \mathcal{S}(t), \forall t \in \mathcal{T}$. I assume that it is possible to find a candidate function h using any of the methods previously presented or within the broader CBF literature. The notion of CBF in Definition 2.14 can then be generalized to the system (5.120) as follows.

Definition 5.6 (Impulsive Timed Control Barrier Function)**.** *Let ψ_h be as in* (5.123)*. A continuous function $h : \mathcal{T} \times \mathcal{X} \to \mathbb{R}$ is an* Impulsive Timed Control Barrier Function (ITCBF) *for the system* (5.120) *if*

$$\inf_{u \in \mathcal{U}} \psi_h(t + \Delta T, t, g(t, x, u)) \leq 0, \forall x \in \mathcal{H}(t), \forall t \in \mathcal{T}\,. \tag{5.125}$$

Note that 1) I relax Definition 2.14 to no longer require differentiability of h, though differentiability is helpful when applying the tools in the prior sections, and 2) condition (5.125) does not include a class-$\mathcal{K}$ function, as this is unnecessary in sampled controllers such as (5.120). The following theorem then provides sufficient conditions for forward invariance of $\mathcal{H}(t)$.

Theorem 5.19. *Given an ITCBF $h : \mathcal{T} \times \mathcal{X} \to \mathbb{R}$ for the system* (5.120)*, let $\mathcal{H}(t)$ be as in* (5.124)*, and ψ_h as in* (5.123)*. Let $u : \mathcal{D} \times \mathcal{X} \to \mathcal{U}$ be a control law, and let $\mathcal{Z}_{coast}$ be as in* (5.121)*. If u satisfies*

$$\psi_h(t + \Delta t, t, x) \leq 0, \quad \forall (t, \sigma, x) \in (\mathcal{D} \times \mathcal{H}) \cap \mathcal{Z}_{coast}, \tag{5.126a}$$

$$\psi_h(t + \Delta T, t, y) \leq 0, \quad \forall (t, \sigma, x) \in (\mathcal{D} \times \mathcal{H}) \setminus \mathcal{Z}_{coast}, \tag{5.126b}$$

where $y = g(t, x, u(t, \sigma, x))$, then u renders time-varying set $\mathcal{H}(t)$ forward invariant for all $t \in \mathcal{T}$.

Proof. Given $(t_0, \sigma(t_0)) \in \mathcal{D}$ and $x(t_0) \in \mathcal{H}(t_0)$, divide $\{t \in \mathcal{T} \mid t \geq t_0\}$ into a sequence of intervals $\mathcal{I}_k = [t_k, t_{k+1}]$, $k \in \mathbb{Z}_{\geq 0}$, where $t_{k+1} > t_k$. Then a sufficient condition for u to render $\mathcal{H}(t)$ forward invariant for all future $t \in \mathcal{T}$ is for the following two properties to hold for every $k \in \mathbb{Z}_{\geq 0}$: 1) u renders $x(t) \in \mathcal{H}(t)$ for all t in the interval $\mathcal{I}_k$, and 2) the endpoint t_{k+1} of $\mathcal{I}_k$ is an impulse opportunity. If $u = 0$, then condition (5.126a) implies that both properties hold for $t_{k+1} = t_k + \Delta t$. If $u \neq 0$, then condition (5.126b) implies that both properties hold for $t_{k+1} = t_k + \Delta T$. Thus, u renders $\mathcal{H}(t)$ forward invariant. ■

Thus, I have presented conditions analogous to those in Theorems 2.2, 2.6, and 2.8 that render sets of the form (5.124) forward invariant subject to the impulsive dynamics (5.120). The remaining challenge is to determine functions h and ψ_h satisfying (5.125) and (5.123), respectively. I first discuss conditions for asymptotic stability before providing examples of h and ψ_h in Section 5.3.3.5.

5.3.3.3 One-Step MPC Impulsive Stability

I now begin to address the stability part of Problem 5.3. There has been much work on stability of hybrid systems with continuous actuators [245, 248, 249, 255], impulsive actuators [250–252], or both [247]. In summary, given a Lyapunov function $V : \mathcal{T} \times \mathcal{X} \to \mathbb{R}_{\geq 0}$, the conditions [250, Eq. 5], [251, Eq. 8], and [247, Eq. 4b] state that if $V(t, g(t, x, u)) \leq cV(t, x)$ for $c \in (0, 1)$, then for sufficiently frequent jumps, the origin of the system (5.120c) is exponentially stable. These conditions can be readily applied to stabilize (5.120) using periodic impulses. However, when the dwell time ΔT is large, a more efficient strategy may be to examine the predicted value of the Lyapunov function after ΔT has elapsed rather than immediately after the impulse. To this end, consider the following lemma.

Assumption 5.1. *Let $V : \mathcal{T} \times \mathcal{X} \to \mathbb{R}_{\geq 0}$ be a continuously differentiable function satisfying*

$$\alpha_1(\|x\|) \leq V(t, x) \leq \alpha_2(\|x\|) \tag{5.127}$$

for all $x \in \mathcal{X}$ and all $t \in \mathcal{T}$ for two functions $\alpha_1, \alpha_2 \in \mathcal{K}_\infty$.

Lemma 5.20. *Let Assumption 5.1 hold. Assume that there exists $\alpha_3 \in \mathcal{K}_\infty$ such that f in* (5.120) *satisfies $\|f(t, x)\| \leq \alpha_3(\|x\|)$ for all $t \in \mathcal{T}$ and $x \in \mathcal{X}$. Let p be as in* (5.122)*. Let $u : \mathcal{D} \times \mathcal{X} \to \mathcal{U}$ be a control law, and denote $\mathcal{Z}_1 \equiv \mathcal{Z}_{coast}$ as in* (5.121) *and $\mathcal{Z}_2 = (\mathcal{D} \times \mathcal{X}) \setminus \mathcal{Z}_1$. For the system* (5.120)*, if u satisfies*

$$V(t + \Delta t, p(t + \Delta t, t, x)) \leq V(t, x), \quad \forall (t, \sigma, x) \in \mathcal{Z}_1, \tag{5.128a}$$

$$V(t + \Delta T, p(t + \Delta T, t, y)) \leq V(t, x), \ \forall (t, \sigma, x) \in \mathcal{Z}_2, \tag{5.128b}$$

where $y = g(t, x, u(t, \sigma, x))$, then u renders the origin uniformly stable as in [71, Def. 4.4].

Proof. First, note that (5.127) implies that every sublevel set of V is compact and contains the origin. Also recall that I assumed solutions to (5.120) always exist, so p in (5.122) and (5.128) is well defined.

Let $(t_k, \sigma(t_k)) \in \mathcal{D}$ be an impulse opportunity. For brevity, denote $z_k = (t_k, \sigma(t_k), x(t_k))$. First, suppose that $z_k \in \mathcal{Z}_2$, which implies that $u(z_k) \neq 0$. Then no other impulses can be applied for $t \in (t_k, t_k + \Delta T)$, so $x(t_k + \Delta T) = p(t_k + \Delta T, t_k, g(t_k, x(t_k), u(z_k)))$, and thus (5.128b) ensures that $V(t_k + \Delta T, x(t_k + \Delta T))$ at the next impulse opportunity $(t_k + \Delta T, \sigma(t_k + \Delta T)) \in \mathcal{D}$ is upper bounded by $V(t_k, x(t_k))$. Since V is bounded at both t_k and $t_k + \Delta T$ and V satisfies (5.127), it follows that both $\|x(t_k)\|$ and $\|x(t_k + \Delta T)\|$ are also bounded. Since $\|f(t, x)\|$ is bounded by $\alpha_3(\|x\|)$ and ΔT is fixed and finite, the amount that V can grow during the jump at $x(t_k^+)$ and during the flow $t \in (t_k, t_k + \Delta T)$ are bounded as $V(t, x(t)) \leq V(t_k, x(t_k)) + \alpha_4(\|x(t_k)\|)$ for some $\alpha_4 \in \mathcal{K}_\infty$.

Next, suppose that $z_k \in \mathcal{Z}_1$. Then $u(z_k) = 0$, so the next impulse opportunity will occur at $t_k + \Delta t$. By (5.128a), $V(t_k + \Delta t, x(t_k + \Delta t)) \leq V(t_k, x(t_k))$. By the same argument as the prior case, V is again bounded as $V(t, x(t)) \leq V(t_k, x(t_k)) + \alpha_4(\|x(t_k)\|)$ for some $\alpha_4 \in \mathcal{K}_\infty$ for all $t \in [t_k, t_k + \Delta t)$.

Thus, V is nonincreasing at the impulse opportunities $(t_k, \sigma(t_k)) \in \mathcal{D}$, and the maximum value of $V(t, x(t)) - V(t_k, x(t_k))$ is uniformly bounded (i.e. uniform in t) by $\alpha_4(\|x(t_k)\|)$ for t between the impulse opportunities. Thus, for initial state $x(t_0) \in \mathcal{X}$, $(t_0, \sigma(t_0)) \in \mathcal{D}$, and any future time $t \in \mathcal{T}$, $t \geq t_0$, it holds that $\|x(t)\| \leq \alpha_1^{-1}(V(t, x(t))) \leq \alpha_1^{-1}(V(t_k, x(t_k)) + \alpha_4(\|x(t_k)\|)) \leq \alpha_1^{-1}(V(t_0, x(t_0)) + \alpha_4(\|x(t_0)\|)) \leq \alpha_1^{-1}(\alpha_2(\|x(t_0)\|) + \alpha_4(\|x(t_0)\|))$. This is equivalent to uniform stability of the origin. ■

Lemma 5.20 differs from [247, 250, 251] in three ways. First, (5.128) provides conditions on the future state, which is explicitly computed using (5.122), rather than the present state. Second, these predictions allow one to avoid explicitly checking for upper bounds on the growth of V during flows, as is required in [250, 251]. Third, Lemma 5.20 allows for aperiodic impulses, as long as (5.128) are checked at their respective frequencies.

I refer to (5.128) as a "one-step Model Predictive Control (MPC)" strategy. That is, to evaluate (5.128), I input the control u at a single (i.e. "one-step") time instance, make a prediction using (5.122), and then check a condition on V, analogous to checking constraints in an MPC optimization. Note that encoding (5.128) into an optimization problem could be computationally expensive, since checking (5.128) entails computing the solution to a differential equation during every iteration of the optimization. In Section 5.3.4, I assume that this cost is acceptable, or that there exists an analytic form for the solution, as is the case for many spacecraft orbits.

5.3.3.4 Impulsive Stability via Restriction to Stable Flows

Motivated by fuel efficiency, a strategy in aerospace systems (e.g. [21]) is to allow a system to coast uncontrolled until a control impulse is necessary to continue stabilization. In this subsection, I implement such a strategy subject to constraints R-1 and R-2 via a specialization of Lemma 5.20. In technical terms, given a Lyapunov function V as in (5.127), I seek to render the set

$$\mathcal{S}_v(t) \triangleq \{x \in \mathcal{X} \mid v(t,x) \leq 0\} \tag{5.129}$$

forward invariant, where, for readability, I denote

$$v(t,x) \equiv \dot{V}(t,x) = \partial_t V(t,x) + \nabla V(t,x) f(t,x)\,. \tag{5.130}$$

This is possible under dynamics (5.120) if $v : \mathcal{T} \times \mathcal{X} \to \mathbb{R}$ is also an ITCBF as in Definition 5.6. Let ψ_v be an upper bound for v analogous to ψ_h in (5.123). In the following theorem, I provide new conditions to establish stability using such a coasting strategy. However, if $x(t_0) \notin \mathcal{S}_v(t_0)$, then these conditions will not initially apply, so I instead fall back on the "one-step MPC" strategy in (5.128). Divide the state space into two sets: 1) $\mathcal{Z}_1 \cup \mathcal{Z}_2$, where the controller enforces (5.128), and 2) $\mathcal{Z}_3 \cup \mathcal{Z}_4$, where the controller enforces the new conditions (5.131).

Theorem 5.21. *Let Assumption 5.1 hold. Assume that there exists $\alpha_3 \in \mathcal{K}_\infty$ such that f in* (5.120) *satisfies $\|f(t,x)\| \leq \alpha_3(\|x\|)$ for all $t \in \mathcal{T}$ and $x \in \mathcal{X}$. Let v be as in* (5.130)*, ψ_v be as in* (5.123)*, and p be as in* (5.122)*. Let $\mathcal{Z}_1$, $\mathcal{Z}_2$, $\mathcal{Z}_3$, and $\mathcal{Z}_4$ be four disjoint sets such that $\mathcal{Z}_1 \cup \mathcal{Z}_3 = \mathcal{Z}_{coast}$ in* (5.121)*, and $\mathcal{Z}_2 \cup \mathcal{Z}_4 = (\mathcal{D} \times \mathcal{X}) \setminus \mathcal{Z}_{coast}$. Then for the system* (5.120)*, any control law $u : \mathcal{D} \times \mathcal{X} \to \mathcal{U}$ satisfying* (5.128) *and all of the following*

$$\psi_v(t + \Delta t, t, x) \leq 0, \qquad \forall (t,\sigma,x) \in \mathcal{Z}_3, \tag{5.131a}$$

$$\psi_v(t + \Delta T, t, g(t,x,u(t,\sigma,x))) \leq 0, \quad \forall (t,\sigma,x) \in \mathcal{Z}_4, \tag{5.131b}$$

$$V(t, g(t,x,u(t,\sigma,x))) \leq V(t,x), \qquad \forall (t,\sigma,x) \in \mathcal{Z}_4, \tag{5.131c}$$

will render the origin uniformly stable as in [71, Def. 4.4].

Proof. Let $(t_k, \sigma(t_k)) \in \mathcal{D}$ be an impulse opportunity, and let $(t_{k+1}, \sigma(t_{k+1})) \in \mathcal{D}$ be the next impulse opportunity. For brevity, denote $z_k = (t_k, \sigma(t_k), x(t_k))$. First, Lemma 5.20 implies that if $z_k \in \mathcal{Z}_1 \cup \mathcal{Z}_2$, then $V(t_{k+1}, x(t_{k+1})) \leq V(t_k, x(t_k))$.

Next, if $z_k \in \mathcal{Z}_3 \cup \mathcal{Z}_4$, conditions (5.131a)-(5.131c) similarly imply that $V(t_{k+1}, x(t_{k+1})) \leq V(t_k, x(t_k))$. Specifically, if $z_k \in \mathcal{Z}_3 \subseteq \mathcal{Z}_{\text{coast}}$, then no impulse is applied, and (5.131a) implies that $V(t, x(t))$ is nonincreasing along the flow f for all $t \in [t_k, t_k + \Delta t)$ until the next impulse opportunity at $t_{k+1} = t_k + \Delta t$. Next, if $z_k \in \mathcal{Z}_4$, then a nonzero impulse is applied, (5.131c) implies

that V is nonincreasing during the impulse, and (5.131b) implies that $V(t, x(t))$ is nonincreasing along the flow f for all $t \in (t_k, t_k + \Delta T)$ until the next impulse opportunity at $t_{k+1} = t_k + \Delta T$. Thus, $V(t_{k+1}, x(t_{k+1})) \leq V(t_k, x(t_k))$ for all $(t_k, \sigma(t_k)) \in \mathcal{D}$, so the origin is uniformly stable by the same argument as Lemma 5.20. ■

Compared to [247, 250, 251], Theorem 5.21 imposes stricter conditions on the flows (5.131a)-(5.131b) in order to allow relaxed conditions on the jumps (5.131c) and the jump times. In [250, 251], it is assumed that the flows are destabilizing and jumps are exponentially stabilizing, whereas Theorem 5.21 says that if one can restrict the flow (5.131a)-(5.131b) to the set in (5.129) where $\dot{V} \leq 0$, as is often possible in practice, then the jump (5.131c) only needs to be stabilizing, not exponentially stabilizing. This coasting strategy can reduce control usage compared to the exponentially stabilizing impulses in [250, 251], and is distinct from the coasting strategy in [21] because of the explicit inclusion of a minimum time between impulses. Note that (5.131a)-(5.131b) are identical to (5.126a)-(5.126b), so a controller as in Theorem 5.21 will further render $\mathcal{S}_v$ in (5.129) forward invariant if $x(t_0) \in \mathcal{S}_v(t_0)$ and $\mathcal{Z}_3 \cup \mathcal{Z}_4 = \mathcal{D} \times \mathcal{S}_v$. Finally, I present a result on asymptotic stability that I will use in Section 5.3.4.

Corollary 5.22. *Let the conditions of Theorem 5.21 hold. If there exists $\beta_1, \beta_2 \in \mathcal{K}_r$ and $\Delta T_{max} \in \mathbb{R}_{>0}$ such that 1)* (5.132a)-(5.132b) *hold and 2) either 2a)* (5.132c)-(5.132d) *hold or 2b)* (5.132e)-(5.132f) *hold*

$$V(t + \Delta t, p(t + \Delta t, t, x)) - w \leq -\beta_2(w), \quad \forall (t, \sigma, x) \in \mathcal{Z}_1, \tag{5.132a}$$

$$V(t + \Delta T, p(t + \Delta T, t, y)) - w \leq -\beta_2(w), \quad \forall (t, \sigma, x) \in \mathcal{Z}_2, \tag{5.132b}$$

$$\psi_v(t + \Delta t, t, x) \leq -\beta_1(w), \quad \forall (t, \sigma, x) \in \mathcal{Z}_3, \tag{5.132c}$$

$$\psi_v(t + \Delta T, t, y) \leq -\beta_1(w), \quad \forall (t, \sigma, x) \in \mathcal{Z}_4, \tag{5.132d}$$

$$V(t, y) - w \leq -\beta_2(w), \quad \forall (t, \sigma, x) \in \mathcal{Z}_4, \tag{5.132e}$$

$$\sigma \geq \Delta T_{max} \implies u(t, \sigma, x) \neq 0, \quad \forall (t, \sigma, x) \in \mathcal{D} \times \mathcal{X}, \tag{5.132f}$$

where $y = g(t, x, u(t, \sigma, x))$ and $w = V(t, x)$, then the origin is uniformly asymptotically stable [71, Def. 4.4].

Proof. The main idea of this proof is to show that there exists a convergent sequence $\{V_k\}_{k=1}^N$, where I denote $V_k = V(t_k, x(t_k))$, each $(t_k, \sigma(t_k)) \in \mathcal{D}$ is an impulse opportunity, and $\Delta t \leq t_{k+1} - t_k \leq \Delta T_{\text{max}}$. I do this in three parts. For brevity, denote $z_k = (t_k, \sigma(t_k), x(t_k))$.

First, conditions (5.132a)-(5.132b) strengthen (5.128a)-(5.128b) so that the "one-step MPC" strategy is now asymptotically stabilizing for all $z_k \in \mathcal{Z}_1 \cup \mathcal{Z}_2$. Specifically, if $z_k \in \mathcal{Z}_1$, then let $t_{k+1} = t_k + \Delta t$, so that (5.132a) is equivalent to $V_{k+1} - V_k \leq -\beta_2(V_k) \leq -\beta_2(V_{k+1})$. Similarly, if $z_k \in \mathcal{Z}_2$, then (5.132b) implies the same result for $t_{k+1} = t_k + \Delta T$.

Second, if 2a holds, then impulses are stabilizing as in Theorem 5.21, and flows are now asymptotically stabilizing for all $z_k \in \mathcal{Z}_3 \cup \mathcal{Z}_4$. If $z_k \in \mathcal{Z}_3$, then let $t_{k+1} = t_k + \Delta t$. Then (5.132c) implies that $v(t, x(t)) \equiv \dot{V}(t, x(t)) \leq -\beta_1(V(t, x(t))) \leq -\beta_1(V_{k+1})$ for all $t \in (t_k, t_{k+1})$. It follows that $V_{k+1} - V_k \leq -\beta_1(V_{k+1})\Delta t$. If 2a holds and instead $z_k \in \mathcal{Z}_4$, then let $t_{k+1} = t_k + \Delta T$. Then (5.131c) implies that V is nonincreasing during the impulse, so (5.132d) similarly implies that $V_{k+1} - V_k \leq -\beta_1(V_{k+1})\Delta T$.

Third, if 2b holds, then flows are stabilizing as in Theorem 5.21, and impulses are now asymptotically stabilizing for all $z_k \in \mathcal{Z}_3 \cup \mathcal{Z}_4$. Condition (5.132f) implies that impulses occur at least as frequently as ΔT_{max}, so let t_{k+1} be the time of the last impulse opportunity in $[t_k, t_k + \Delta T_{\text{max}}]$. Next, let $\{\tau_j\}_{j=1}^{M}$ be the sequence of impulse opportunity times starting at t_k and ending at t_{k+1}. Then Theorem 5.21 implies that $V(\tau_{j+1}, x(\tau_{j+1})) \leq V(\tau_j, x(\tau_j))$ for all $j \in \{1, \cdots, M\}$. Moreover, (5.132e)-(5.132f) imply that there exists at least one $\tau_j \in \{\tau_j\}_{j=1}^{M-1}$ such that $V(\tau_{j+1}, x(\tau_{j+1})) - V(\tau_j, x(\tau_j)) \leq -\beta_2(V(\tau_j, x(\tau_j)))$. It follows that $V_{k+1} - V_k \leq -\beta_2(V(\tau_j, x(\tau_j))) \leq -\beta_2(V_{k+1})$.

Recall that $\mathcal{Z}_1 \cup \mathcal{Z}_2 \cup \mathcal{Z}_3 \cup \mathcal{Z}_4 = \mathcal{D} \times \mathcal{X}$, so the combination of (5.132a)-(5.132b) and either (5.132c)-(5.132d) or (5.132e)-(5.132f) covers all possible states. In every case, I showed that $V_{k+1} - V_k \leq -\beta(V_{k+1})$ for some $\beta \in \mathcal{K}_r$. Equivalently, $V_{k+1} + \beta(V_{k+1}) \leq V_k$. Since each $V_k \geq 0$, this condition describes a convergent sequence $\{V_k\}_{k=1}^{N}$. If $\mathcal{T}$ is unbounded, then $N = \infty$, and $\lim_{k\to\infty} V_k = 0$. Note that this convergence is uniform in time, because β is only a function of V (i.e. β is not a function of t and V). Since (5.120) is uniformly stable, $\|x\|$ satisfies $\|x\| \leq \alpha_1^{-1}(V(t, x(t)))$, the sequence t_k satisfies $t_{k+1} - t_k \leq \Delta T_{\text{max}}$, and V_k is uniformly convergent, it follows that the origin of (5.120) is uniformly asymptotically stable. ■

That is, if the Lyapunov function V is nonincreasing as in Theorem 5.21, and either the flows (5.132c)-(5.132d) or the jumps (5.132e)-(5.132f) cause V to strictly decrease, then the origin is asymptotically stable. Again, I provide alternative "one-step MPC" conditions (5.132a)-(5.132b) in case (5.132c)-(5.132f) cannot be satisfied because $x(t) \notin \mathcal{S}_v(t)$. If one further assumes that β_1 and β_2 are linear functions, then the conditions in Corollary 5.22 become special cases of [247, Thm. 1].

5.3.3.5 Examples of Bounding Functions

In this subsection, I discuss in more detail how to develop ψ_h and ψ_v to use in the preceding theorems. Suppose second order dynamics such that $x = [r^{\text{T}}, \dot{r}^{\text{T}}]^{\text{T}} \in \mathbb{R}^n$ for flow dynamics $\ddot{r} = f_r(x)$. First, an obstacle avoidance constraint can be written using the Exponential CBF h

[85] as follows:

$$\kappa(t,x) = \rho - \|r - r_0(t)\| \tag{5.133a}$$

$$h(t,x) = \kappa(t,x) + \gamma\dot{\kappa}(t,x) \tag{5.133b}$$

$$\psi_h(t+\delta,t,x) = \max\left\{h(t,x), \kappa(t,x) + (\gamma+\delta)\dot{\kappa}(t,x) + \left(\tfrac{1}{2}\delta^2 + \gamma\delta\right)\ddot{\kappa}_{\max}\right\} \tag{5.133c}$$

where $\rho \in \mathbb{R}_{>0}$ is the obstacle radius, $\gamma \in \mathbb{R}_{>0}$ is a constant, $r_0 : \mathcal{T} \to \mathbb{R}^{n/2}$ is the center of the obstacle, and $\ddot{\kappa}_{\max} \in \mathbb{R}_{\geq 0}$ is an upper bound on the possible values of $\ddot{\kappa}$ between t and $t+\delta$. I use formula (5.133c) for the bound ψ_h because κ in (5.133a) is not thrice differentiable, so one cannot make use of any higher order derivatives. Next, the rate of change of a quadratic Lyapunov function $V(t,x) = x^{\mathrm{T}}Px$ can be upper bounded as

$$\psi_v(t+\delta,t,x) = \dot{V}(t,x) + \max\{0, \ddot{V}(t,x)\}\delta + \tfrac{1}{2}\dddot{V}_{\max}\delta^2 \tag{5.134}$$

where $\dddot{V}_{\max} \in \mathbb{R}_{\geq 0}$ is an upper bound on the possible values of $\dddot{V}$. I stop the approximation ψ_v at the third derivative of V, because $\dddot{V}$ is a function of only derivatives and higher powers of f_r, so higher order approximations do not substantially decrease conservatism.

5.3.3.6 Decreasing Conservatism

Note that the upper bounds derived in Sections 5.1-5.2 and [119, 125, 130, 244] and implemented above in (5.133c) and (5.134) were intended for relatively short horizon times $\tau - t$. For very large horizon times, these upper bounds can become overly conservative. One can optionally decrease this conservatism by breaking the interval $\tau - t$ into $n_\psi \in \mathbb{Z}_{\geq 1}$ smaller intervals. To this end, let $\delta = (\tau - t)/n_\psi$ and $\tau_j = t + j\delta$, and for a scalar function $h : \mathcal{T} \times \mathcal{X} \to \mathbb{R}$, replace ψ_h as above with $\psi_h^* : \mathcal{T} \times \mathcal{X} \to \mathbb{R}^{n_\psi}$ with elements defined as

$$[\psi_h^*(\tau,t,x)]_j = \psi_h(\tau_j, \tau_{j-1}, p(\tau_{j-1},t,x)) \tag{5.135}$$

for $j = 1, \cdots, n_\psi$. That is, ψ_h^* makes n_ψ exact state predictions using p in (5.122), which could be more expensive to compute, and bounds the evolution between these predictions using the original ψ_h function. This division is analogous to MPC with a control horizon of 1, a prediction horizon of n_ψ, and a discretization margin encoded in ψ_h. An illustration of the advantages of this approach is shown in Fig. 5.23. In the above work, all statements of the form $\psi_a(\cdot) \leq 0$, where a is h or v, can be equivalently replaced by $\psi_a^*(\cdot) \leq 0$ elementwise. I will demonstrate the utility of this strategy in simulation in Section 5.3.4.

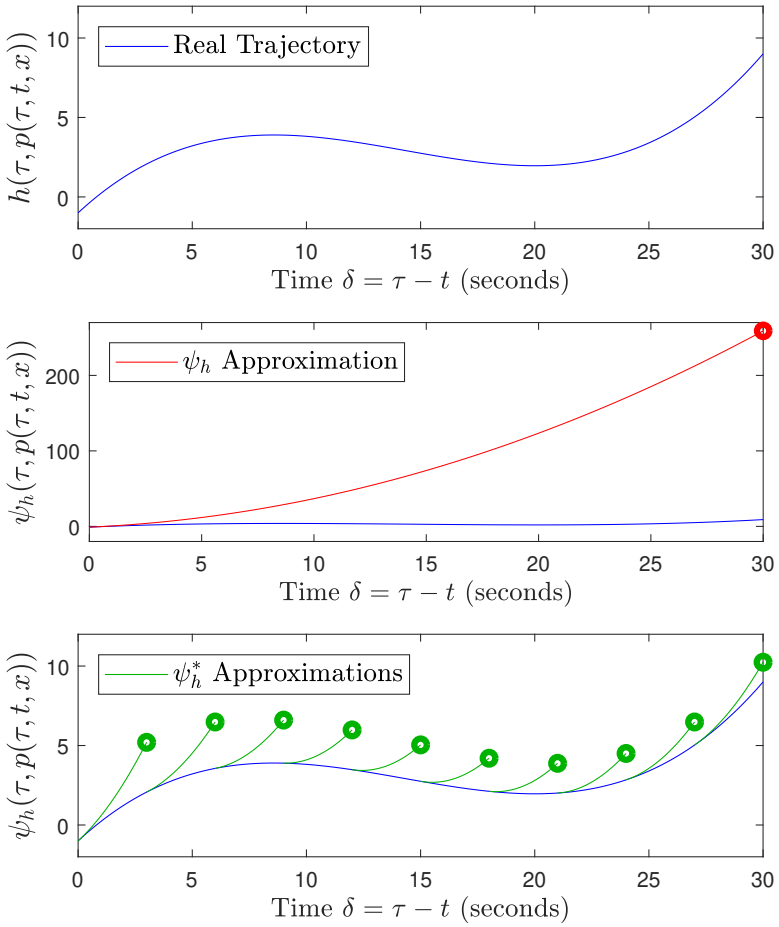

Figure 5.23: Illustration of Using a Single Trajectory Sample versus Multiple Samples. An example of a continuous future trajectory curve (top), an approximation ψ_h (middle), and an approximation ψ_h^* with $n_\psi = 10$ (bottom). The margin of the approximation (5.133c) in the middle plot is quadratic in the propagated time. Thus, making n_ψ samples of the continuous curve allows for a total smaller margin (the 10 green samples, where each green line is also a quadratic curve), at the expense of creating a nonlinear optimization problem.

5.3.4 Application to Satellite Docking with Impulsive Thrusters

I now validate the above methods by simulating an impulsive system representative of spacecraft docking in low Earth orbit. Let $\mathcal{X} = \mathbb{R}^4, \mathcal{U} = \mathbb{R}^2$, let $\mu \in \mathbb{R}_{>0}$ be constant, and let

$$f(\cdot) = \begin{bmatrix} x_3 \\ x_4 \\ -\mu x_1/(x_1^2 + x_2^2)^{3/2} \\ -\mu x_2/(x_1^2 + x_2^2)^{3/2} \end{bmatrix}, \; g(\cdot) = \begin{bmatrix} x_1 \\ x_2 \\ x_3 + u_1 \\ x_4 + u_2 \end{bmatrix}. \tag{5.136}$$

Let there be four CBFs h_i of the form (5.133b) for various obstacles $r_i(t) \in \mathbb{R}^2$, representing other objects in orbit, with ψ_{h_i} as in (5.133c). Let there be an additional constraint $\kappa_5(t, x) = (r - r_5)^{\mathrm{T}}(\dot{r}_5/\|\dot{r}_5\|) \leq 0$ with associated CBF h_5 also as in (5.133b). That is, κ_5 encodes that the controlled satellite r must always lie behind an uncontrolled target satellite $r_5(t) \in \mathbb{R}^2$. Let $x_t(t) = [r_5(t)^{\mathrm{T}}, \, \dot{r}_5(t)^{\mathrm{T}}]^{\mathrm{T}}$. I choose a Lyapunov function $V(t, x) = (x - x_t(t))^{\mathrm{T}} P (x - x_t(t))$ and approximation ψ_v^* as in (5.134) and (5.135). Let $\gamma_1, \gamma_2 \in \mathbb{R}_{\geq 0}$ and $J \in \mathbb{R}_{>0}$ be constants. The chosen control law is

$$u = \begin{cases} 0 & \psi_v(\cdot) \leq \gamma_1 V(t, x) \text{ and } \psi_{h_i}(\cdot) \leq 0, \, \forall i \in \mathcal{I} \\ u^* & \text{else} \end{cases} \tag{5.137a}$$

where $(\cdot) = (t + \Delta t, t, x)$, $\mathcal{I} = \{1, 2, 3, 4, 5\}$, and u^* is

$$u^* = \underset{u \in \mathbb{R}^2, d \in \mathbb{R}_{\geq 0}}{\arg\min} \; u^{\mathrm{T}} u + J d^2 \tag{5.137b}$$

$$\text{such that } \psi_v^*(t + \Delta T, t, g(t, x, u)) \leq \gamma_1 V(t, x) + d \tag{5.137c}$$

$$V(t, g(t, x, u)) \leq \gamma_2 V(t, x) + d \tag{5.137d}$$

$$\psi_{h_i}(t + \Delta T, t, g(t, x, u)) \leq 0, \, \forall i \in \mathcal{I}. \tag{5.137e}$$

I assume that the optimization (5.137) is always feasible, though I note that this is difficult to guarantee when there are multiple CBFs [138, 140], as will be elaborated upon in Chapter 6. I simulated (5.137) using various choices of ΔT, and then repeated these simulations with ψ_h in (5.137e) replaced with ψ_h^* as in (5.135) with $n_{\psi_h} = 10$. The resultant trajectories, converted to Hill's frame for visualization, are shown in Fig. 5.24, and full results are shown in the video at https://youtu.be/_o-FAGbvfgg. A comparison to a trajectory pre-planner is also shown in Fig. 5.24, and details on select trajectories are shown in Figs. 5.25-5.26. All simulation code and parameters can be found at https://github.com/jbreeden-um/phd-code/tree/main/2023/LCSS%20Impulsive%20Control.

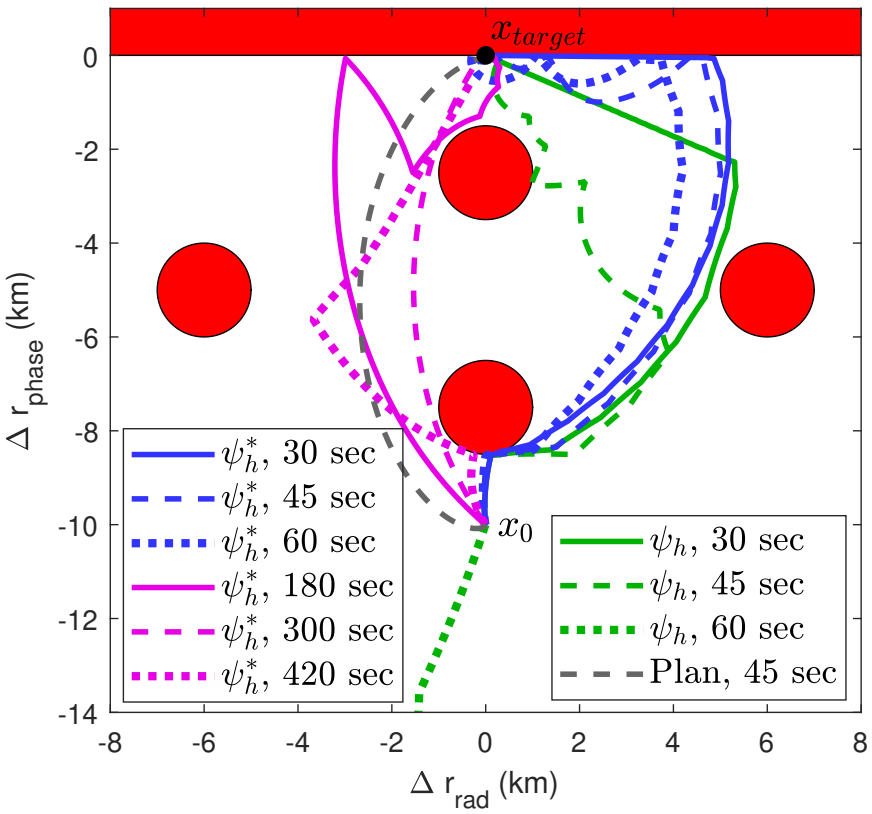

Figure 5.24: Impulsive Satellite Docking Trajectories. Trajectories of (5.120) and (5.136) subject to the control (5.137)

All of the simulations in Fig. 5.24 remained safe, and eight of the nine trajectories converged to the target. The trajectory using ψ_h with $\Delta T = 60$ was so conservative that it immediately turned away from the target, whereas trajectories using ψ_h^* still converge with much larger ΔT, though the rate of convergence is slow for $\Delta T \geq 420$. This is because ψ_h^* implements (5.133c) with a smaller, less conservative, δ than ψ_h alone. That said, this decreased conservatism came at an average computational cost per control cycle, for $\Delta T = 45$, of 0.22 s using ψ_h^* and 0.022 s using ψ_h. These computation times are for a 3.5 GHz CPU, and would likely be much larger onboard a spacecraft processor. The total fuel consumption varied from 188 m/s ($\Delta T = 30$ with ψ_h) to 18.2 m/s ($\Delta T = 300$ with ψ_h^*). For comparison, the pre-planned trajectory consumed between 12.2 m/s and 13.9 m/s depending on the choice of ΔT. This improvement is expected since (5.137) only considers ΔT seconds of the trajectory at a time, whereas a pre-planner can optimize over longer sequences.

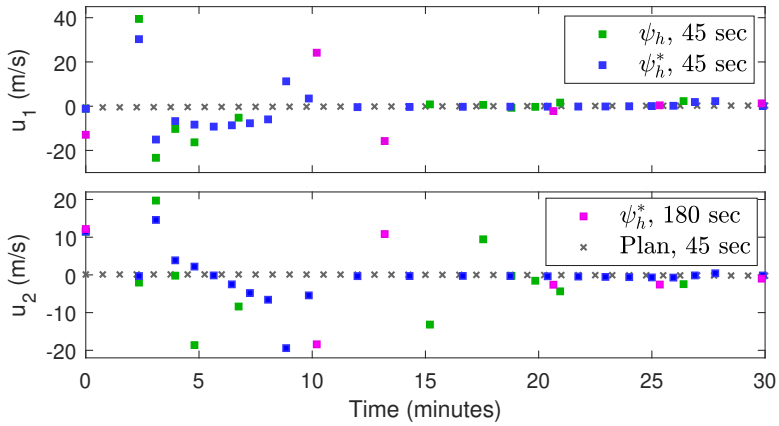

Figure 5.25: Impulsive Satellite Docking Control Inputs. Control inputs along selected trajectories in Fig. 5.24

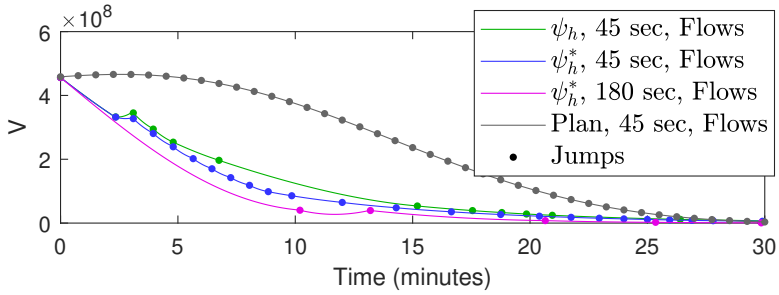

Figure 5.26: Impulsive Satellite Docking Lyapunov Function Values. Lyapunov function along selected trajectories in Fig. 5.24

5.3.5 Conclusions

This section has developed a methodology for extending the provable set invariance guarantees provided by CBFs to systems with impulsive actuators subject to a minimum dwell time constraint, and for ensuring asymptotic stability in the same systems. I then encoded the resulting conditions in an optimization-based control law similar to (2.9), which was successful in a simulated spacecraft docking. The conditions presented are similar to those in Sections 5.1-5.2, but are generally nonlinear in the control input, thus leading to controllers that are solutions to nonlinear optimizations. I showed how one can reduce the conservatism of these controllers, in exchange for greater computational cost, by dividing the safety prediction horizon into multiple intervals using an MPC-like strategy.

A natural future direction for this research is to consider extensions of the above results to systems with disturbances, analogous to how Section 5.2 added disturbances to the method in Section 5.1.3.2. Such extensions could be accomplished by a "robust tube" approach [256], where the predicted path p in (5.122) instead belongs to an expanding tube of possible paths. Such a tube could be modelled either using Lipschitz constants as in Section 5.2, or by generalizing (5.120) to a differential inclusion for all considered disturbances, or possibly other methods. In either case, such a tube would likely have an impact on the maximum available precision, similar to in Section 4.7. This research direction is not considered further in this dissertation, and is left as an open area for future researchers. Other directions for future research include studying additional methods to decrease conservatism, and the use of ITCBFs with optimal trajectory planning to yield more fuel-efficient trajectories. Future researchers might also consider how to design control laws for impulsive actuators when the minimum impulse magnitude is lower bounded, and how this lower bound might impact the range of achievable results, similar to how the disturbance bounds impacted the range of achievable results in Section 4.7.

5.4 Conclusions on Sampled-Data Systems

To review, this chapter considered the problem of how to implement the CBF conditions as in Theorems 2.2, 2.6, 2.8 and Theorems 4.1, 4.3, 4.4 using a sampled-data control law, for a variety of models (5.1), (5.30), (5.120). I started with the simplest form of this problem, a ZOH control law applied to the deterministic and time-invariant model (5.1). I then expanded to a time-varying model with disturbances in (5.30). I concluded by showing how these ZOH control tools could also be applied to the impulsive model (5.120). One natural extension of the work in this chapter would be a model containing all of the above, i.e. a model subject to disturbances, continuously-applied ZOH control channels, and impulsively-applied control channels. Though such an all-

inclusive model is not studied in this dissertation, the above tools have nonetheless substantially expanded the applicability of CBFs. At the start of my studies, CBFs were generally applied in control laws with very high update frequencies to closely approximate the continuous-time results. Now, using the tools above, one can provably achieve safe set forward invariance even when the control law is sampled so infrequently that the system behavior no longer closely approximates the hypothetical behavior following from a continuously-varying control law. This tolerance for sampling is extremely important for spacecraft applications, as spacecraft typically have limited computational resources, and as there is not yet a metric for CBFs analogous to the phase margin of linear-time-invariant systems that can concisely capture the effects of the controller sampling. The remaining chapters of this dissertation consider other obstacles to the adoption of CBFs on space systems.

CHAPTER 6

Simultaneous Application of Multiple Control Barrier Functions

In this chapter, I finally address the "multiple control barrier functions problem" to which I have alluded repeatedly in the previous chapters. At this point, I emphasize that none of the prior work in this dissertation explicitly considered whether the application of multiple control barrier functions at once in a QP was a valid control strategy. Given two or more CBFs, it is common in the CBF literature to simply express the two CBF conditions (e.g. (3.9), (4.8), (5.5), (5.126)) as two affine constraints on the QP (2.9). This strategy was also used in the Eros simulations in Chapters 3-4, the unicycle with two constraints in Section 5.1, the multiple pointing constraints in Section 5.2, and the multiple obstacles in Section 5.3. Even though I have not yet studied whether such a strategy is valid—and in fact I will show in this chapter that this strategy is usually not valid everywhere in the CBF sets—all of the aforementioned simulations were successful (note that the docking controllers in Section 4.7 with multiple CBFs are still provably valid even after taking into account the concerns in this chapter). Similarly, other papers in the CBF literature often successfully apply multiple CBF conditions simultaneously without considering interactions between the CBFs. Thus, the reader may conclude that the work in this chapter, while necessary for provable safety, is not necessary to achieve useful control results. Reasons for this are expanded upon in the conclusions in Section 6.5.

To motivate this problem, consider a single CBF in the time-invariant case. If a function $h : \mathcal{X} \to \mathbb{R}$ satisfies the definition of a CBF in (2.4) in Definition 6.1 for an open set $\mathcal{D}$ containing $\mathcal{H}$ in (2.2), then the optimization problem in the control law (2.9) is always feasible over $\mathcal{D}$, and thus by Theorem 2.2, such a control law will render $\mathcal{H}$ forward invariant. However, if there are two CBFs $h_1 : \mathcal{X} \to \mathbb{R}$ and $h_2 : \mathcal{X} \to \mathbb{R}$, then Definition 6.1 does not necessarily imply that both CBF conditions (2.5) are feasible simultaneously. This will be demonstrated in Example 6.1.

That said, if the system is control affine and the control set $\mathcal{U}$ is $\mathbb{R}^m$, then as long as $\frac{\partial h_1(x)}{\partial x} g(x) \neq -\frac{\partial h_2(x)}{\partial x} g(x)$, a QP as in (2.9) with two constraints will be feasible. Moreover, even if $\frac{\partial h_1(x)}{\partial x} g(x) = -\frac{\partial h_2(x)}{\partial x} g(x)$, as long as $x \in \mathcal{H}_1 \cap \mathcal{H}_2$, the modified QP in (6.12) will be feasible and will render

$\mathcal{H}_1 \cap \mathcal{H}_2$ forward invariant. However, if $\mathcal{U} \neq \mathbb{R}^m$, then the QP may or may not be feasible. The work in this chapter is focused on determining how to construct CBFs so that their associated CBF conditions are provably concurrently feasible.

The work in this chapter is divided into two perspectives. The first (Section 6.3.1) is a collection of geometric observations. Recall that an affine condition on the control input results in a half-space constraint. Thus, using geometry, I identify conditions that ensure that the intersection of these half spaces with $\mathcal{U}$ is non-empty. Second, if these geometric results fail or are overly conservative, I suggest an iterative algorithm (Section 6.3.3) for deriving a controlled invariant set parameterized by CBFs. Note that this algorithm can be followed by hand, or coded to search for a controlled invariant set using a computer.

6.1 Introduction and Literature Review

Control Barrier Functions (CBFs) are a control synthesis method for ensuring that system state trajectories always remain within some specified *safe set* [66]. In general, there may exist points within the safe set for which all feasible trajectories originating at these points will eventually exit the safe set. In such cases, one often seeks to construct a CBF so that a subset of the safe set, herein called the *CBF set*, is controlled invariant for the given system dynamics and input bounds. The problem of finding a CBF is equivalent to the problem of finding a description for such a controlled invariant set. This in itself is a challenging problem, but for simple safe sets, i.e. safe sets that can be described as a sublevel set of a single *constraint function*, several authors have proposed strategies to find CBFs as functions of the constraint function, including [66, 86, 87, 92] and Chapters 3-4, among others.

One open challenge in the CBF literature is that most works assume that there exists a controlled invariant subset of the safe set that is sufficiently simple to be described by the zero sublevel (or superlevel) set of a single CBF. More complex sets could be expressed as the intersection of the zero sublevel sets of multiple CBFs, e.g. when each CBF represents a single obstacle in a cluttered environment. This is sometimes achieved by taking a smooth [169], nonsmooth [151], or adaptive [139] maximum over several CBFs, or by simply applying multiple CBFs at once in a Quadratic Program (QP) [136, 138]. However, these approaches all assume that one is able to find a collection of CBFs whose CBF sets form a controlled invariant set when intersected, or else the QP could become infeasible, as is illustrated by the two planar constraints in Fig. 6.1. Finding such a collection of CBFs is a much more challenging problem than finding a single CBF. The authors are not aware of any general algorithms analogous to [66, 86, 87, 92] for finding several CBFs at once, excepting learning approaches [156, 157], which can only yield probabilistic safety guarantees.

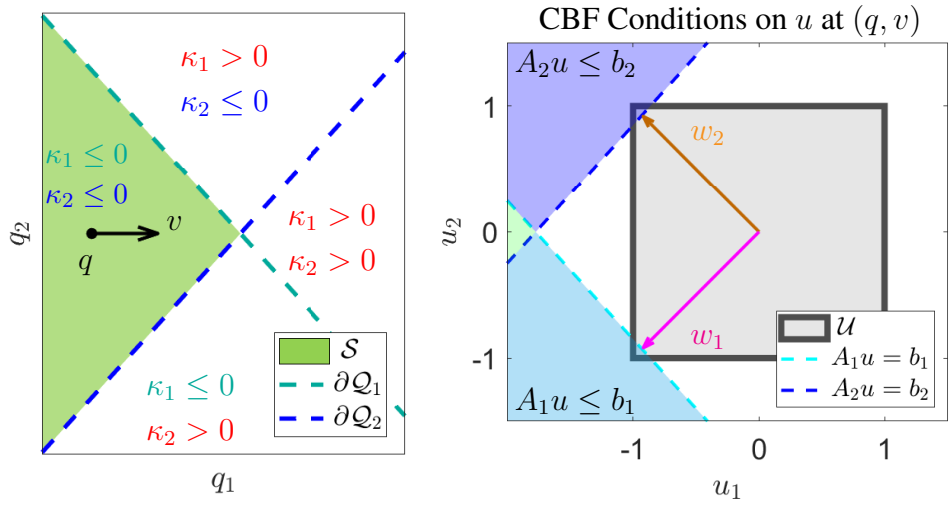

Figure 6.1: Two CBF Sets and the Associated CBF Conditions in the Control Space.
Left: Visualization of the safe positions $q = (q_1, q_2) \in \mathbb{R}^2$ (green) for a double-integrator agent with two constraints κ_1, κ_2 as in Example 6.1. The state $x_0 = (q, v) \in \mathbb{R}^4$ in Example 6.1 has position q labeled above and velocity v pointing to the right.
Right: Visualization of the control space at x_0 in Example 6.1. Both CBF conditions are individually feasible via control inputs w_1 and w_2, but there is no $u \in \mathcal{U}$ satisfying both conditions simultaneously, so the intersection of the CBF sets is not a controlled invariant set.

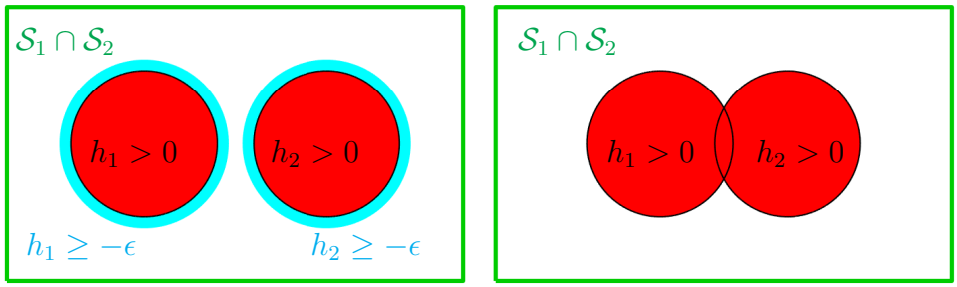

Figure 6.2: Intersecting and Nonintersection CBF Set Boundaries.
Left: In [67], the problem of multi-CBF safety is simplified by assuming that $\{x \mid h_1(x) = 0\} \cap \{x \mid h_2(x) = 0\} = \emptyset$, and then designing separate safe controllers for the cases $h_1(x) \geq -\epsilon$ and $h_2(x) \geq -\epsilon$.
Right: By contrast, this work is interested in the case where $\{x \mid h_1(x) = 0\} \cap \{x \mid h_2(x) = 0\} \neq \emptyset$, and thus both CBF conditions must be simultaneously satisfied, potentially resulting in conflicts such as that in Fig. 6.1.

The most common strategy to employ multiple CBFs is to assume that all the CBFs act independently of each other [67, 93]. For example, the CBFs may apply to different states with decoupled input channels, as occurs for the docking simulation in Section 4.7 and in [93], or it may be the case that only one CBF acts at a time [67]. The latter is equivalent to assuming that the boundaries of the individual CBF zero sublevel sets, i.e. the CBF zero level sets, do not intersect, as shown in Fig. 6.2. In this case, the CBFs may be designed in a one-at-a-time fashion using existing algorithms. However, if this does not hold, then the CBFs must be designed all-at-once, or else the intersection of the CBF sets may not be controlled invariant. Note that for most problems with high relative-degree constraint functions, the corresponding CBF sets usually have intersecting boundaries because of the velocity states, even if this is not obvious from the zero level sets of the constraint functions (i.e. from the position states).

There exist many tools for the computation of viable sets and controlled invariant sets in the control verification literature [186–188, 257–260]. Such tools have also been used to construct CBFs [147], and have been augmented via application of CBF concepts [185]. However, all of these algorithms are computationally expensive, even for linear systems [187, 188].

Compared to prior works, this chapter presents two contributions. First, I present a methodology for decoupling the design of multiple CBFs in the presence of prescribed input bounds. This decoupling allows one to leverage existing single-CBF algorithms in Chapters 3-4 and [66, 86, 87, 92], while avoiding conflicts such as those in Fig. 6.1. Second, I present an iterative algorithm to find a controlled invariant set parameterized by these decoupled CBFs. That is, instead of using zonotopes [186, 258], polytopes [188, 259], or other parameterized functions [185, 260, 261], this work expresses the viable sets in terms of an intersection of CBF sets. As each CBF forbids state trajectories from crossing the boundary of its own CBF set, the proposed algorithm focuses on verifying Nagumo's condition only at the states where the boundaries of multiple CBF sets intersect. One can then compute safe control actions using a QP [66, 262] subject to the CBF conditions arising from every CBF, and I show that this QP is always feasible. Note that, compared to [186–188, 257–260], the viable sets resulting from this algorithm will generally be more conservative, as the intention of this work is to enable the use of existing (usually conservative) CBF literature rather than to approximate the maximal viability kernel.

6.2 Preliminaries

6.2.1 Notations

The results of this chapter can be applied to any constraint function, but this dissertation is generally concerned with constraint functions of relative-degree 2, so I will specialize the notations

to that case. Given a function $\varphi : \mathbb{R}^{n_1} \times \mathbb{R}^{n_2} \to \mathbb{R}$, denoted $\varphi(q, v)$, let $\nabla_q \varphi(q, v)$ denote the row vector of derivatives (i.e. the gradient) of φ with respect to inputs $q \in \mathbb{R}^{n_1}$, and let $\nabla_v \varphi(q, v)$ denote the row vector of derivatives of φ with respect to inputs $v \in \mathbb{R}^{n_2}$. Let $\mathcal{C}^r$ denote the set of functions r-times continuously differentiable in all arguments. Given one or more vectors $\{v_1, \cdots, v_N\}$ and a vector u of the same dimension, define the projection operator as $\text{proj}_{\{v_1,\cdots,v_N\}} u = \sum_{i \in [M]} (u \cdot \hat{b}_i) \hat{b}_i$, where $\{\hat{b}_1, \cdots, \hat{b}_M\}$ is an orthonormal basis spanning span$\{v_1, \cdots, v_N\}$.

6.2.2 Model

Let $q \in \mathbb{R}^{n_1}$ be the coordinates, and $v \in \mathbb{R}^{n_2}$ the velocities, of a second-order system of the form

$$\dot{q} = g_1(q)v, \tag{6.1a}$$

$$\dot{v} = f(q, v) + g_2(q)u\,, \tag{6.1b}$$

with state $x \triangleq (q, v) \in \mathbb{R}^{n_1+n_2}$ and control input $u \in \mathcal{U} \subset \mathbb{R}^m$. Let $f : \mathbb{R}^{n_1} \times \mathbb{R}^{n_2} \to \mathbb{R}^{n_2}$, $g_1 : \mathbb{R}^{n_1} \to \mathbb{R}^{n_1 \times n_2}$, and $g_2 : \mathbb{R}^{n_1} \to \mathbb{R}^{n_2 \times m}$ belong to $\mathcal{C}^2$. Let $\mathcal{U}$ encode the set of allowable control inputs, herein called the *control set*, and assume that $\mathcal{U}$ is compact and convex and contains the zero vector. Assume that f, g_1, g_2, and u are sufficiently regular that trajectories of (6.1) exist and are unique for all times $t \in \mathcal{T} = [t_0, t_f)$, where t_f is possibly ∞. Note that the model (6.1) includes Euler-Lagrange systems, where $n_1 = n_2$, as in [92, 93, 263], but is also more general. I use this form of model because it also includes the attitude control case study from Section 5.2, which I will expand upon in this chapter.

6.2.3 Safety Definitions

A principal requirement of any autonomous system should be to at all times satisfy certain operation constraints. Specifically, suppose that one is given several *constraint functions* $\kappa_i : \mathbb{R}^{n_1} \to \mathbb{R}, \kappa_i \in \mathcal{C}^2, i \in [N_1]$ and $\eta_j : \mathbb{R}^{n_2} \to \mathbb{R}, \eta_j \in \mathcal{C}^1, j \in [N_2]$, which each generate a *constraint set*:

$$\mathcal{Q}_i \triangleq \{q \in \mathbb{R}^n \mid \kappa_i(q) \leq 0\}\,, \tag{6.2}$$

$$\mathcal{V}_j \triangleq \{v \in \mathbb{R}^n \mid \eta_j(v) \leq 0\}\,. \tag{6.3}$$

I call the composition of all the constraint sets the *safe set* $\mathcal{S}$, and say that the system is *safe* at time t if $x(t) \in \mathcal{S}$:

$$\mathcal{S} \triangleq \left(\bigcap_{i \in [N_1]} \mathcal{Q}_i \right) \times \left(\bigcap_{j \in [N_2]} \mathcal{V}_j \right) . \tag{6.4}$$

That is, I allow for both position constraints κ_i and velocity constraints η_i, but assume that these constraints are encoded separately. This implies that each κ_i is of relative-degree 2 along (6.1) and each η_i is of relative-degree 1 along (6.1). This separation of $\mathcal{Q}_i$ and $\mathcal{V}_i$ is not necessary, but is greatly simplifying, as will be explained in Remark 6.3.

I now introduce two more specific notions of CBF.

Definition 6.1 (CBF on a State and Control Set)**.** *Let $\mathcal{X} \subseteq \mathcal{S}$ and $\mathcal{Y} \subseteq \mathcal{U}$. A function $h : \mathbb{R}^{n_1+n_2} \to \mathbb{R}, h \in \mathcal{C}^1$ is a* control barrier function (CBF) for $(\mathcal{X}, \mathcal{Y})$ *if there exists $\alpha \in \mathcal{K}_e$ such that the set*

$$\mathcal{U}_{\text{cbf}}(q, v, \mathcal{Y}) \triangleq \{u \in \mathcal{Y} \mid \dot{h}(q, v, u) \leq \alpha(-h(q, v))\} \tag{6.5}$$

is nonempty for all $(q, v) \in \mathcal{H} \cap \mathcal{X}$, where

$$\mathcal{H} \triangleq \{(q, v) \in \mathbb{R}^{n_1+n_2} \mid h(q, v) \leq 0\} . \tag{6.6}$$

I call $\mathcal{H}$ the CBF induced set*, or simply* CBF set.

Note that under the dynamics (6.1), $\dot{h}$ is affine in u,

$$\dot{h}(q, v, u) = \nabla_q h(q, v) g_1(q) v + \nabla_v h(q, v) f(q, v) + \nabla_v h(q, v) g_2(q) u \, , \tag{6.7}$$

so if $\mathcal{Y}$ is convex, then the set $\mathcal{U}_{\text{cbf}}$ in (6.5) is also convex.

Definition 6.2 (Simple CBF on a State and Control Set)**.** *Let $\mathcal{X} \subseteq \mathcal{S}$ and $\mathcal{Y} \subseteq \mathcal{U}$. A function $h : \mathbb{R}^{n_1+n_2} \to \mathbb{R}, h \in \mathcal{C}^1$ is a* simple control barrier function (SCBF) for $(\mathcal{X}, \mathcal{Y})$ *if 1) h is a CBF for $(\mathcal{X}, \mathcal{Y})$, and 2) the set*

$$\mathcal{U}_{\text{scbf}}(q, v, \mathcal{Y}) \triangleq \{u \in \mathcal{U}_{\text{cbf}}(q, v, \mathcal{Y}) \mid \exists c \geq 0 : u = -c[\nabla_v h(q, v) g_2(q)]^{\mathrm{T}}\} \tag{6.8}$$

is nonempty for all $(q, v) \in \mathcal{H} \cap \mathcal{X}$.

That is, h is an SCBF if the condition $\dot{h}(q, v, u) \leq \alpha(-h(q, v))$ can always be satisfied for a control input u anti-parallel to the vector $\nabla_v h(q, v) g_2(q)$. I introduce the notion of SCBF because it allows for slightly less conservative viable set computations, as I will show in Section 6.3. SCBFs arise naturally in many practical applications. Note that if $\mathcal{Y} = \{u \in \mathbb{R}^m \mid \|u\|_2 \leq u_{\max}\}$, then any CBF for $(\mathcal{X}, \mathcal{Y})$ is automatically an SCBF for $(\mathcal{X}, \mathcal{Y})$ too.

Next, recall the viability definitions in Definition 2.7 and Definition 2.8. I combine these two definitions as follows.

Definition 6.3 (Viable Controlled Invariant Set)**.** *A set $\mathcal{A} \subseteq \mathcal{S}$ is called a* viable controlled-invariant set (VCIS) *if for every point $x(t_0) \in \mathcal{A}$, there exists a control signal $u(t) \in \mathcal{U}, t \in \mathcal{T}$ such that the trajectory $x(\cdot)$ of* (6.1) *satisifes $x(t) \in \mathcal{A}$ for all $t \in \mathcal{T}$.*

Note that if 1) h is a CBF for $(\mathcal{S},\mathcal{U})$ and 2) $\mathcal{H} \subseteq \mathcal{S}$, then it follows from Definition 6.1 and Theorem 2.2 that $\mathcal{H}$ is a VCIS. This is useful because generally $\mathcal{S}$ in (6.4) is not a VCIS, but one can use CBFs to describe subsets of $\mathcal{S}$ that are VCISs. However, in this work, I assume that $\mathcal{S}$ is sufficiently complex that it is difficult to find a single CBF h for which $\mathcal{H} \subseteq \mathcal{S}$, thus motivating the following extensions of Theorem 2.2.

6.2.4 Safe Quadratic Program Control

The heart of safety-critical control is Nagumo's theorem, previously stated in Lemma 2.7. For a set $\mathcal{A} \subset \mathbb{R}^{n_1+n_2}$, recall that if $x \in \text{int}(\mathcal{A})$, then $\mathbf{T}_{\mathcal{A}}(x) = \mathbb{R}^{n_1+n_2}$, so Lemma 2.7 in effect only depends on the flow $\dot{x}$ when $x \in \partial\mathcal{A}$. I use this fact in Lemma 6.2 below.

Given multiple constraint functions $\kappa_i, i \in [N_1]$ and $\eta_j, j \in [N_2]$, it is common [136, 138, 140] to construct multiple CBFs $h_k, k \in [M]$. These can then be used for safe control according to the following lemma, which follows from Theorem 2.2.

Lemma 6.1. *Given functions $\{h_k\}_{k\in[M]}$, with $\mathcal{H}_k$ as in* (6.6)*, denote $\mathcal{H}_{\text{all}} = \cap_{k\in[M]}\mathcal{H}_k$. Assume that each h_k is a CBF for $(\mathcal{H}_{\text{all}},\mathcal{U})$ and let $\mathcal{D}$ be an open set satisfying $\mathcal{D} \supset \mathcal{H}_{\text{all}}$. Then any control law $u : \mathbb{R}^{n_1+n_2} \to \mathcal{U}$ satisfying*

$$u(q,v) \in \mathcal{U}_{\text{cbf,all}}(q,v) \triangleq \bigcap_{k\in[M]} \mathcal{U}_{\text{cbf},k}(q,v,\mathcal{U}) \tag{6.9}$$

for all $(q,v) \in \mathcal{D}$ will render $\mathcal{H}_{\text{all}}$ forward invariant.

Remark 6.1. *Note that while the definitions of CBFs and SCBFs in Definitions 6.1-6.2 are concerned only with states $(q,v) \in \mathcal{H}$, Lemma 6.1 instead requires that* (6.9) *is satisfied for all $(q,v) \in \mathcal{D}$ for a larger open set $\mathcal{D}$, similar to Theorem 2.2. As explained in Chapter 2, this need for a larger open set $\mathcal{D}$ is a mathematical technicality, and this condition can be replaced by instead assuming that ∇h does not vanish at the boundary of each set $\mathcal{H}$. This technicality is not addressed further in this chapter (i.e. this chapter continues to focus only on designing CBFs for $\mathcal{H}$ rather than for a larger open set $\mathcal{D}$), but should still be accounted for upon implementation. See Section 2.3.2 for more information.*

Note that Lemma 6.1 does not include the safe set $\mathcal{S}$ from (6.4), so safety is only guaranteed if $\mathcal{H}_{\text{all}} \subseteq \mathcal{S}$. I next introduce a modified version of Lemma 6.1 to handle the possibility that $\mathcal{H}_{\text{all}} \not\subseteq \mathcal{S}$, as frequently occurs when using High Order CBFs (e.g. [87] and Section 4.2.4).

Lemma 6.2. *Given functions $\{h_k\}_{k\in[M]}$, with $\mathcal{H}_k$ as in (6.6), denote $\mathcal{H}_{\text{all}} = \cap_{k\in[M]}\mathcal{H}_k$ and $\mathcal{A} = \mathcal{H}_{\text{all}} \cap \mathcal{S}$. Assume that each h_k is a CBF for $(\mathcal{A},\mathcal{U})$. Let $u : \mathbb{R}^{n_1+n_2} \to \mathcal{U}$ be a control law. Assume that $\nabla\kappa_i(q,v) \neq 0$ and $\nabla\eta_j(v) \neq 0$ for all $(q,v) \in \mathcal{S}$ and all $i \in [N_1], j \in [N_2]$. For every $i \in [N_1]$, $j \in [N_2]$, and all $(q,v) \in \text{int}(\mathcal{H}_{\text{all}})$, assume that $\kappa_i(q) = 0 \implies \dot{\kappa}_i(q,v) \leq 0$ and that $\eta_j(v) = 0 \implies \dot{\eta}_j(q,v,u(q,v)) \leq 0$. Let $\mathcal{D}$ be an open set satisfying $\mathcal{D} \supset \mathcal{A}$. If u satisfies (6.9) for all $(q,v) \in \mathcal{D}$, then u will render $\mathcal{A}$ forward invariant.*

Note that $\kappa_i(q) = 0 \implies \dot{\kappa}_i(q,v) \leq 0$ is equivalent to Nagumo's necessary condition in Lemma 2.7, and the conditions $\nabla\kappa_i \neq 0$ and $\nabla\eta_i \neq 0$ are similar to the condition on $|\frac{\partial h(t,x)}{\partial t}| + \|\frac{\partial h(t,x)}{\partial x}\|$ in Theorem 2.8. Thus, Lemma 6.2 highlights how one purpose of the CBFs h_k is to construct a set $\mathcal{A} \subseteq \mathcal{S}$ that excludes all the states in $\mathcal{S}$ where Nagumo's necessary condition does not hold for any $u \in \mathcal{U}$. This idea is the central motivation for the algorithm in Section 6.3.3.

Finally, given a collection of constraints κ_i, η_j and CBFs h_k satisfying the conditions of Lemma 6.2, it is common to construct control laws $u : \mathbb{R}^{n_1+n_2} \to \mathcal{U}$ of the form

$$u(q,v) = \underset{u\in\mathcal{U}}{\arg\min} \|u - u_{\text{nom}}(q,v)\|_2^2 \tag{6.10a}$$

$$\text{s.t. } \dot{h}_k(q,v,u) \leq \alpha_k(-h_k(q,v)), \forall k \in [M] \tag{6.10b}$$

where α_k comes from (6.5), u_{nom} is any control law, and (6.10b) is affine due to (6.7). If u in (6.10) always exists, i.e. if $\mathcal{U}_{\text{cbf,all}}$ in (6.9) is nonempty for all $(q,v) \in \mathcal{A}$, then it follows from Lemma 6.2 that the set $\mathcal{A}$ in Lemma 6.2 is a VCIS. However, if there exists $(q,v) \in \mathcal{A}$ for which $\mathcal{U}_{\text{cbf,all}}(q,v)$ is empty, then (6.10) could become infeasible and thus trajectories originating at (q,v) might (remember that Lemma 6.2 is sufficient, not necessary) exit the safe set $\mathcal{S}$. Thus, the goal of this chapter is to present tools to ensure that $\mathcal{A}$ is a VCIS, so that the related controller (6.12) (to be introduced) is always feasible.

Problem 6.1. *Determine a set of CBFs $\{h_k\}_{k\in[M]}$ such that the set $\mathcal{A}$ in Lemma 3 is a VCIS.*

6.2.5 Assumptions and Motivating Example

The principal challenge addressed in this chapter is the problem of multi-CBF compositions. For this reason, I assume that the single-CBF problem is sufficiently solved.

Assumption 6.1. *Given sets $\mathcal{X} \subseteq \mathcal{S}$ and $\mathcal{Y} \subseteq \mathcal{U}$, focus on any single constraint function κ_i (or η_j). Denote $\mathcal{O} = \mathcal{Q}_i \times \mathbb{R}^{n_2}$ (or $\mathcal{O} = \mathbb{R}^{n_1} \times \mathcal{V}_j$). Consider the set $\mathcal{Z} = \mathcal{X} \cap (\text{int}(\mathcal{X}) \cup \partial\mathcal{O})$.*

Assume that there exists an algorithm (e.g. [66, 86, 87, 92] or Chapters 3-4) to derive one or more functions $\{h_k\}_{k=k_1}^{k_2}$, each a CBF for $(\mathcal{X}, \mathcal{Y})$, such that $\kappa_i(q) = 0 \implies \dot{\kappa}_i(q, v) \leq 0$ (or $\eta_j(v) = 0 \implies \dot{\eta}_j(q, v, u) \leq 0$) for all $(q, v) \in (\mathcal{Z} \cap (\cap_{k=k_1}^{k_2} \operatorname{int}(\mathcal{H}_k)))$ and all $u \in \mathcal{U}$, where $\mathcal{H}_k$ is as in (6.6)*. That is, for each constraint function, assume that one already possesses sufficiently capable tools to find one or more CBFs that prevents state trajectories from violating that particular constraint function.*

Remark 6.2. *In practice, for the relative-degree 1 constraint functions η_j, one can often choose a CBF h_k so that $h_k = \eta_j$. In this case, the set $\mathcal{Z} \cap \operatorname{int}(\mathcal{H}_k)$ in Assumption 6.1 has empty intersection with the set $\{(q, v) \in \mathcal{X} \mid \eta_j(v) = 0\}$, so the condition "$\eta_j(v) = 0 \implies \dot{\eta}(q, v, u) \leq 0$ for all $(q, v) \in (\mathcal{Z} \cap \operatorname{int}(\mathcal{H}_k)), u \in \mathcal{U}$" is automatically satisfied. Therefore, the "all $u \in \mathcal{U}$" part of Assumption 6.1 is a rarely used technicality. One only needs to check this technicality if one chooses a CBF h_k such that there exists $(q, v) \in \mathcal{H}_k$ where $\eta_j(v) > 0$. I am unaware of a practical example where this occurs for a relative-degree 1 constraint function η_j, though such a choice of h_k is common for relative-degree 2 constraint functions κ_i, as was discussed in Sections 4.3.1-4.3.2.*

Successive application of Assumption 6.1 to every constraint function one-at-a-time then produces a collection of CBFs $\{h_k\}_{k\in[M]}$ satisfying the assumptions of Lemma 6.2. However, this still does not imply joint feasibility of all the CBFs h_k, as illustrated in the following example.

Example 6.1. *Consider the 2D double integrator $\dot{q} = v, \dot{v} = u$, $q = (q_1, q_2) \in \mathbb{R}^2, v = (v_1, v_2) \in \mathbb{R}^2$, $u = (u_1, u_2) \in \mathcal{U} = [-1, 1] \times [-1, 1]$ subject to two constraint functions $\kappa_1(q) = q_1 + \gamma q_2$ and $\kappa_2(q) = q_1 - \gamma q_2$ for some constant $\gamma > 0$, resulting in the safe set $\mathcal{S} = (\mathcal{Q}_1 \cap \mathcal{Q}_2) \times \mathbb{R}^2$, pictured in Fig. 6.1. From Section 4.3.1, one can derive CBFs h_1, h_2 for $(\mathcal{S}, \mathcal{U})$, where $h_i(q, v) = \dot{\kappa}_i(q, v) - \sqrt{-2(1+\gamma)\kappa_i(q)}$, that satisfy the conditions of Lemma 6.2. Denote $\mathcal{A} = \mathcal{H}_1 \cap \mathcal{H}_2 \cap \mathcal{S}$ and let $x_0 = (q, v) = (-\frac{1}{2(1+\gamma)}, 0, 1, 0) \in \partial\mathcal{A}$. Then there is no $u \in \mathcal{U}$ that renders $\mathcal{A}$ forward invariant from x_0 (see the right side of Fig. 6.1). That is, Nagumo's necessary condition (Lemma 2.7) for forward invariance of $\mathcal{A}$ is violated at x_0.*

6.3 Proposed Methods

It is clear from Example 6.1 that possessing a collection of CBFs for $(\mathcal{S}, \mathcal{U})$ is not sufficient to solve Problem 6.1. Thus, my first proposed strategy is to identify other control sets $\mathcal{Y}$, for which possessing CBFs for $(\mathcal{S}, \mathcal{Y})$ is sufficient to solve Problem 6.1. That is, if one restricts w_1, w_2 in Fig. 6.1 to a smaller set $\mathcal{Y} \subset \mathcal{U}$ when designing CBFs for these constraint functions, then under certain conditions, presented in Section 6.3.1, one can ensure that several CBFs will be concurrently feasible over the full control set $\mathcal{U}$. When this strategy fails to yield a full solution

to Problem 6.1, I then present a more typical iterative algorithm in Section 6.3.3 to remove the remaining infeasible states in $\mathcal{S}$. I also present a brief remark on QP controllers in Section 6.3.2.

6.3.1 When All CBFs are Non-Interfering

Some properties I will need are as follows:

Definition 6.4 (Non-Interference)**.** *Two CBFs h_i and h_j are called* non-interfering on $\mathcal{X}$ *if* $(\nabla_v h_i(q,v)g_2(q)) \cdot (\nabla_v h_j(q,v)g_2(q)) \geq 0$ *for all* $(q,v) \in \mathcal{X}$*. A collection of CBFs* $\{h_k\}_{k\in[M]}$ *is called non-interfering if every pair of CBFs is non-interfering.*

Definition 6.5 (Orthogonal Extension Property)**.** *Given a set* $\mathcal{U}' \subset \mathcal{U}$*, let* $\{w_i\}_{i\in[m]}$ *be a set of* m *vectors* $w_i \in \mathcal{U}'$ *satisfying* $w_i \cdot w_j \geq 0, \forall i \in [m], j \in [m]$*. Let* $\{y_i\}_{i\in[m]}$ *be the set of orthogonal projections* $y_i = w_i - \text{proj}_{(\{w_j\}_{j\in[i-1]})} w_i$ *(or* $y_i = w_i$ *if* $\{w_i\}_{i\in[m]}$ *are orthogonal). The set* $\mathcal{U}'$ *has the* orthogonal extension property (OEP) with respect to $\mathcal{U}$ *if for every such set* $\{w_i\}_{i\in[m]}$*, the point* $z = \sum_{i\in[m]} y_i$ *belongs to* $\mathcal{U}$*.*

Definition 6.6 (Quadrant Extension Property)**.** *Given a set* $\mathcal{U}' \subset \mathcal{U}$*, let* $\{\mathcal{P}_i\}_{i\in[m]}$ *be a set of* m *orthogonal hyperplanes in* $\mathbb{R}^m$ *satisfying* $\mathcal{P}_i \cap \mathcal{U}' \neq \emptyset, \forall i \in [m]$*. Let* p *be the point where all* m *hyperplanes intersect (where* p *is guaranteed to exist because* $\{\mathcal{P}_i\}_{i\in[m]}$ *are orthogonal). The set* $\mathcal{U}'$ *has the* quadrant extension property (QEP) with respect to $\mathcal{U}$ *if for every such set* $\{\mathcal{P}_i\}_{i\in[m]}$*, the point* p *belongs to* $\mathcal{U}$*.*

Definition 6.6 is perhaps better termed the "orthant extension property", but I use the term QEP anyways so that Definitions 6.5-6.6 have differing acronyms. Examples of sets satisfying Definitions 6.5-6.6 are as follows:

Example 6.2. *Given various prescribed input bounds* $\mathcal{U}$*, the following sets* $\mathcal{U}'$ *possess the OEP or QEP with respect to* $\mathcal{U}$*. Note that these choices of* $\mathcal{U}'$ *are not unique.*

1. *If* $\mathcal{U} = \{u \in \mathbb{R}^m \mid \|u\|_\infty \leq \gamma\}$*, then one possible set with the OEP is* $\mathcal{U}' = \{u \in \mathbb{R}^m \mid \|u\|_1 \leq \gamma\}$ *(Fig. 6.3a). One possible set with the QEP is* $\mathcal{U}' = \{u \in \mathbb{R}^m \mid \|u\|_1 \leq \gamma^*\}$ *(Fig. 6.3b), where* γ^* *must be computed. If* $m = 2$*, then* $\gamma^* = \frac{2\gamma}{1+\sqrt{2}}$*.*

2. *If* $\mathcal{U} = \{u \in \mathbb{R}^m \mid \|u\|_1 \leq \gamma\}$*, then one possible set with the OEP is* $\mathcal{U}' = \{u \in \mathbb{R}^m \mid \|u\|_\infty \leq \frac{\gamma}{m}\}$*. One possible set with the QEP is* $\mathcal{U}' = \{u \in \mathbb{R}^m \mid \|u\|_\infty \leq \frac{\gamma^*}{m}\}$*, where* γ^* *is as in the prior case.*

3. *If* $\mathcal{U} = \{u \in \mathbb{R} \mid \|u\|_2 \leq \gamma\}$*, then one possible set with the OEP (Fig. 6.3c) and QEP (Fig. 6.3d) is* $\mathcal{U}' = \{u \in \mathbb{R}^m \mid \|u\|_2 \leq \frac{\gamma}{\sqrt{m}}\}$*.*

4. *If $\mathcal{U} = \{u \in \mathbb{R}^m \mid \max_i |a_i u_i| \leq \gamma\}$ for constants $\{a_1, \cdots, a_m\}$, then one possible set with the OEP is $\mathcal{U}' = \{u \in \mathbb{R}^m \mid \sum_i |a_i u_i| \leq \gamma^*\}$ (Fig. 6.3e), where $\gamma^* \leq \gamma$ must be computed. A set with the QEP can be constructed similarly (Fig. 6.3f).*

The core idea of this subsection is that given a set $\mathcal{U}'$ with the OEP or QEP, if one designs the CBFs $\{h_k\}_{k\in[M]}$ one-at-a-time for $(\mathcal{S}, \mathcal{U}')$, then, subject to an additional condition on the CBF gradients, one can guarantee a priori that $\{h_k\}_{k\in[M]}$ will have jointly feasible CBF conditions. To show this, I begin with several lemmas about the geometry of the OEP and QEP, first for two constraints, and then for M constraints.

Lemma 6.3. *Let $\mathcal{U}'$ have the OEP with respect to $\mathcal{U}$. Given two row vectors $A_1, A_2 \in \mathbb{R}^{1\times m}$ and two scalars $b_1, b_2 \in \mathbb{R}$, if $A_1 \cdot A_2 \geq 0$ and there exists $w_1, w_2 \in \mathcal{U}'$ such that 1) $A_1 w_1 \leq b_1$, 2) $A_2 w_2 \leq b_2$, 3) $A_1 \cdot w_1 = -\|A_1\|_2 \|w_1\|_2$, and 4) $A_2 \cdot w_2 = -\|A_2\|_2 \|w_2\|_2$, then there exists $z \in \mathcal{U}$ such that $A_1 z \leq b_1$ and $A_2 z \leq b_2$.*

Proof. Note that conditions 3 and 4 imply that either 1) w_1 is parallel and opposite to A_1 when $b_1 < 0$, or 2) that $w_1 = 0$ when $b_1 \geq 0$, and similarly for A_2, b_2, w_2. Without loss of generality, assume that $\|w_1\|_2 \geq \|w_2\|_2$. Let $w^* = w_2 - \frac{w_2 \cdot w_1}{w_1 \cdot w_1} w_1$. Then $z = w_1 + w^*$ satisfies $A_1 z = A_1 w_1 \leq b_1$ since w^* is orthogonal to w_1 and A_1, and

$$A_2 z = A_2 w_2 + \underbrace{A_2 w_1}_{\leq 0}(1 - \underbrace{\tfrac{w_2 \cdot w_1}{w_1 \cdot w_1}}_{\leq 1}) \leq A_2 w_2 \leq b_2 \,.$$

By the OEP, $z \in \mathcal{U}$. ■

Lemma 6.4. *Let $\mathcal{U}'$ have the OEP with respect to $\mathcal{U}$. Given M row vectors $\{A_k\}_{k\in[M]}$ and scalars $\{b_k\}_{k\in[M]}$ with $A_k \in \mathbb{R}^{1\times m}$, if 1) $A_i \cdot A_j \geq 0$ for all $i \in [M], j \in [M]$, 2) there exists $w_k \in \mathcal{U}'$ such that $A_k w_k \leq b_k$ for all $k \in [M]$, and 3) $A_k w_k = -\|A_k\|_2 \|w_k\|_2$ for all $k \in [M]$, then there exists $z \in \mathcal{U}$ such that $A_k z \leq b_k$ for all $k \in [M]$.*

Proof. The proof follows from that of Lemma 6.3. Assume that the vectors are ordered by decreasing $\|w_k\|_2$. If $M \leq m$, then use orthogonal projections to construct a vector $z = \sum_{i\in[M]}(w_i - \text{proj}_{(\{w_j\}_{j\in[i-1]})} w_i)$ that satisfies all M constraints $A_k z \leq b_k$ simultaneously. By the OEP, $z \in \mathcal{U}$. If $M > m$, then because $A_i \cdot A_j \geq 0$ for all i, j, the set $\mathcal{W} = \{u \in \mathbb{R}^m \mid A_k u \leq b_k, \forall k \in [M]\}$ must contain a complete orthant $\mathbb{O} = \{u \in \mathbb{R}^m \mid O_k u \leq O_k y, \forall k \in [m]\} \subseteq \mathcal{W}$ for some orthogonal set $\{O_k\}_{k\in[m]}$, $O_k \in \mathbb{R}^{1\times m}$ and some point $y \in \mathbb{R}^m$. Therefore, at least $l_0 = M - m$ of the constraints $A_k z \leq b_k$ must be redundant, i.e. $l \geq l_0$ of the constraints must be automatically satisfied if the remaining constraints are satisfied. Use the remaining $M - l$ constraints as in the prior case to construct a vector $z \in \mathcal{U}$ satisfying all M inequalities simultaneously. ■

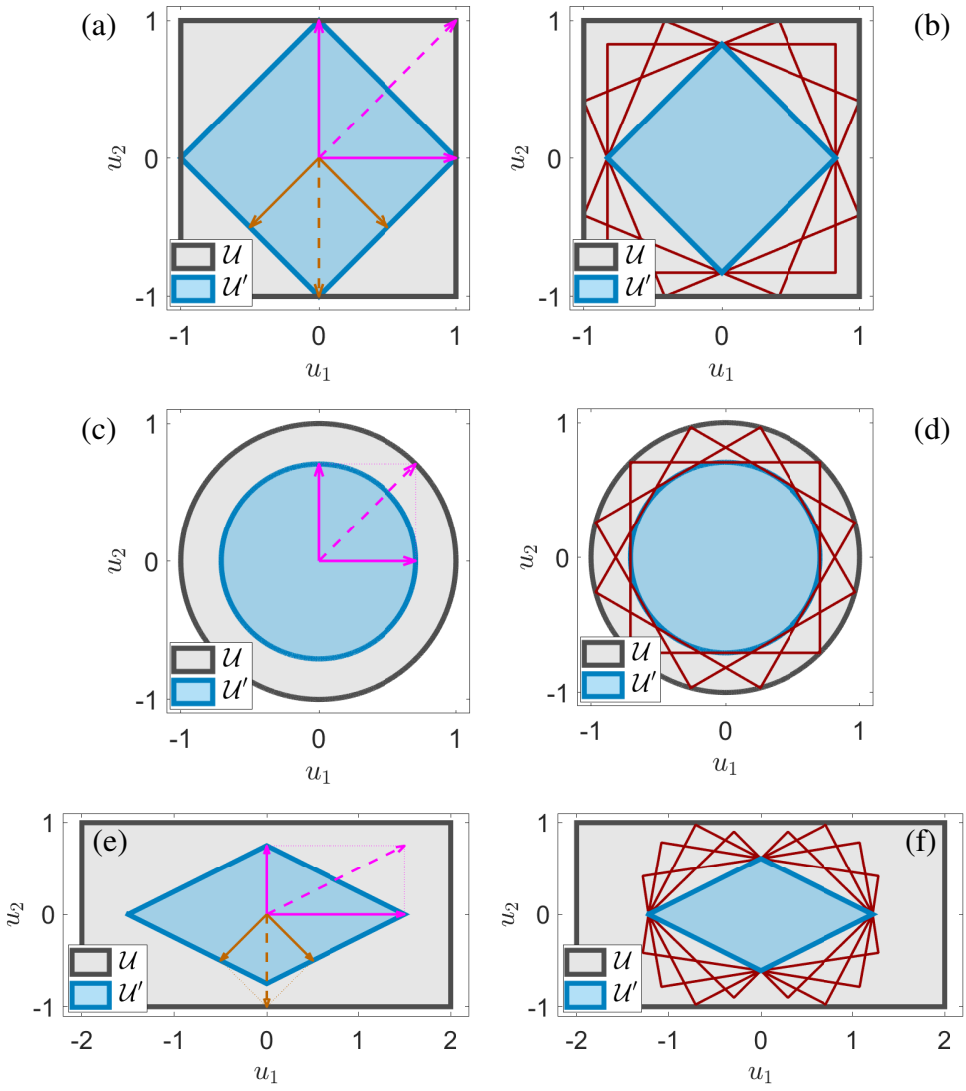

Figure 6.3: Illustration of OEP and QEP. Given three different control sets $\mathcal{U}$ (gray), the above illustrates possible choices of set $\mathcal{U}'$ (blue) that: left) have the OEP with respect to $\mathcal{U}$, and right) have the QEP with respect to $\mathcal{U}$, as in Definitions 6.5-6.6. The left plots also show how given two orthogonal vectors (solid arrows) in $\mathcal{U}'$, their sum (dashed arrow) must by construction belong to $\mathcal{U}$. The right plots also show various choices of orthogonal hyperplanes (i.e. lines in $\mathbb{R}^2$) whose intersections (the red right angles) by construction must belong to $\mathcal{U}$.

Lemma 6.5. *Let $\mathcal{U}'$ have the QEP with respect to $\mathcal{U}$. Given two row vectors $A_1, A_2 \in \mathbb{R}^{1\times m}$ and two scalars $b_1, b_2 \in \mathbb{R}$, if $A_1 \cdot A_2 \geq 0$ and there exists $w_1, w_2 \in \mathcal{U}'$ such that $A_1 w_1 \leq b_1$ and $A_2 w_2 \leq b_2$, then there exists $p \in \mathcal{U}$ such that $A_1 p \leq b_1$ and $A_2 p \leq b_2$.*

Proof. Consider the following figure:

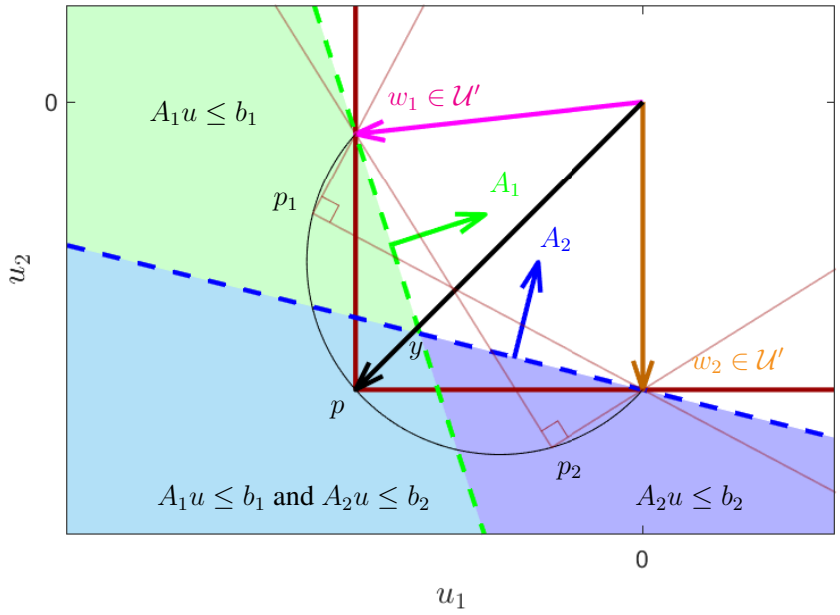

Figure 6.4: Visualization for Lemma 6.5 proof

Since $w_1, w_2 \in \mathcal{U}'$, by the QEP, any two orthogonal lines (i.e. hyperplanes in $\mathbb{R}^2$) that intersect w_1 and w_2 (i.e. intersect points in $\mathcal{U}'$) must meet at a point in $\mathcal{U}$, such as the points p, p_1, p_2 above. That is, every point on the black arc must belong to $\mathcal{U}$. Next, since $A_1 \cdot A_2$ is at least zero, the point y where the hyperplanes $\{u \mid A_1 u = b_1\}$ and $\{u \mid A_2 u = b_2\}$ (dashed lines) intersect must be enclosed by the black arc. It follows that at least one point, above labeled p, on this arc must satisfy both inequalities simultaneously. ■

Lemma 6.6. *Let $\mathcal{U}'$ have the QEP with respect to $\mathcal{U}$. Given M row vectors $\{A_k\}_{k\in[M]}$ and scalars $\{b_k\}_{k\in[M]}$ with $A_k \in \mathbb{R}^{1\times m}$, if $A_i \cdot A_j \geq 0$ for all $i \in [M], j \in [M]$ and there exists $w_k \in \mathcal{U}'$ such that $A_k w_k \leq b_k$ for all $k \in [M]$, then there exists $p \in \mathcal{U}$ such that $A_k p \leq b_k$ for all $k \in [M]$.*

Proof. The argument is similar to that in Lemma 6.5, now extended to higher dimension. Let $\{\mathcal{P}_k\}_{k\in[m]}$ be a set of orthogonal hyperplanes in $\mathbb{R}^m$ that meet at some point $p \in \mathbb{R}^m$ and satisfy $\mathcal{P}_k \cap \mathcal{U}' \neq \emptyset, \forall k \in [m]$. Let $\mathbb{P} = \{u \in \mathbb{R}^m \mid P_k u \leq P_k p, \forall k \in [m]\}$ be the orthant of $\mathbb{R}^m$ originating from p and enclosed by $\{\mathcal{P}_k\}_{k\in[m]}$, for appropriate vectors $\{P_k\}_{k\in[m]}, P_k \in \mathbb{R}^{1\times m}$. As in Lemma 6.4, there are at most m non-redundant constraints. Let $\mathcal{N}$ be the set of indices of these non-redundant constraints, and construct $\mathbb{P}$ so that $w_k \in \partial\mathbb{P}, \forall k \in \mathcal{N}$, analogous to how the solid red lines ($\partial\mathbb{P}$) intersect the vectors w_1, w_2 in Fig. 6.4. Then the point p will lie on the boundary of

an m-hypersphere $\mathbb{S}$ analogous to the black arc in Fig. 6.4, and all possible choices of p must lie in $\mathcal{U}$ by the QEP. Let $y \in \mathbb{R}^m$ satisfy $A_k y = b_k, \forall k \in \mathcal{N}$. By the same argument as in Lemma 6.5, $\mathbb{S}$ must enclose at least one such y. Thus, there exists at least one $p \in \mathbb{S} \subseteq \mathcal{U}$ that satisfies all M inequalities simultaneously. ■

I now apply the above geometric observations to the concurrent feasibility of several SCBFs or CBFs as follows.

Theorem 6.7. *Let $\mathcal{U}'$ be any set with the OEP with respect to $\mathcal{U}$. Let $\{h_k\}_{k\in[M]}$ each be an SCBF for $(\mathcal{S},\mathcal{U}')$. If $\{h_k\}_{k\in[M]}$ are non-interfering on $\mathcal{S}$ as in Definition 6.4, then the set $\mathcal{U}_{\text{cbf,all}}$ in* (6.9) *is nonempty for all $(q,v) \in \mathcal{S} \cap (\cap_{k\in[M]}\mathcal{H}_k)$.*

Proof. Let $A_k = \nabla_v h_k(q,v)g_2(q)$ and $b_k = \alpha_k(-h_k(q,v)) - \nabla_q h_k(q,v)g_1(q)v$ for all $k \in [M]$, where α_k each come from (6.5). It follows from Definition 6.4 that $A_i \cdot A_j \geq 0$ for all $i \in [M], j \in [M]$ everywhere in $\mathcal{S}$. It follows from Definition 6.2 that for each $k \in [M]$, there exists $w_k \in \mathcal{U}'$ such that $A_k w_k \leq b_k$ and w_k is anti-parallel to A_k. Lemma 6.4 then implies that these M inequalities are simultaneously feasible for some u in the full set $\mathcal{U}$, which is equivalent to the sets $\{\mathcal{U}_{\text{cbf},k}\}_{k\in[M]}$ having a nonempty intersection $\mathcal{U}_{\text{cbf,all}}$. ■

Theorem 6.8. *Let $\mathcal{U}'$ be any set with the QEP with respect to $\mathcal{U}$. Let $\{h_k\}_{k\in[M]}$ each be a CBF for $(\mathcal{S},\mathcal{U}')$. If $\{h_k\}_{k\in[M]}$ are non-interfering on $\mathcal{S}$ as in Definition 6.4, then the set $\mathcal{U}_{\text{cbf,all}}$ in* (6.9) *is nonempty for all $(q,v) \in \mathcal{S} \cap (\cap_{k\in[M]}\mathcal{H}_k)$.*

Proof. The proof is identical to that of Theorem 6.7, except that h_k are CBFs instead of SCBFs, so each w_k is not guaranteed to be anti-parallel to each A_k, respectively. Thus, I apply Lemma 6.6 instead of Lemma 6.4. ■

Theorems 6.7-6.8 constitute my first method for ensuring concurrent feasibility of multiple CBFs. Note that I provide theorems under both the OEP and QEP separately, because as shown in Fig. 6.3, the OEP is less conservative (i.e. allows for larger $\mathcal{U}'$), but is only applicable when the stricter SCBF condition (6.8) holds (which is often the case in practice). I also note the following remarks about the computation of the gradients of the CBFs h_k for Definition 6.4 and Theorems 6.7-6.8.

Remark 6.3. *If κ is a High Order CBF as in [87], then there exists a function $\phi \in \mathcal{K}$ such that $h(q,v) = \dot{\kappa}(q,v) - \phi(-\kappa(q))$ is a CBF as in Definition 6.1. Moreover, $\dot{\kappa}(q,v) = \nabla_q \kappa(q) g_1(q) v$, so $\nabla_v h(q,v) \equiv \nabla_q \kappa(q) g_1(q) g_2(q)$. That is, $\nabla_v h(q,v)$ 1) is independent of v and 2) does not depend on the function ϕ. Thus, Definition 6.4 can be checked using only the gradients of the constraint functions κ_i, κ_j and the dynamics* (6.1) *without knowing the exact CBFs (i.e. the choices of ϕ). This is the main motivation for considering the specific second-order model* (6.1)*.*

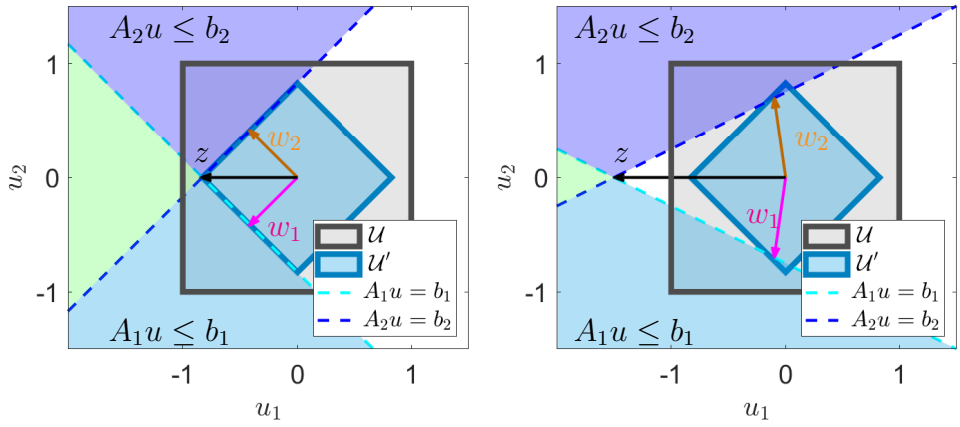

Figure 6.5: Modification of Fig. 6.1 After Shrinking the Assumed Control Set.
Left: Visualization of the control space at a point $x_0 \in \partial\mathcal{H}_1 \cap \partial\mathcal{H}_2$ using $\mathcal{H}_1, \mathcal{H}_2$ as in Example 6.3. Compared to Fig. 6.1, the CBFs are now designed over $\mathcal{U}'$ (the blue diamond) instead of $\mathcal{U}$, so there exists $z \in \mathcal{U}$ satisfying both $A_1 z \le b_1$ and $A_2 z \le b_2$.
Right: Visualization of the control space at the point x_0 in Example 6.4. Because the CBFs h_1, h_2 do not satisfy Definition 6.4, it may be the case that every point z satisfying both inequalities $A_k z \le b_k$ lies outside the set $\mathcal{U}$, so x_0 should not lie in the VCIS.

Remark 6.4. *To use the above strategies with a more general model than* (6.1)*, let* $h : \mathbb{R}^n \to \mathbb{R}$ *take a single argument* x*, and suppose* $\dot{x} = f(x) + g(x)u$*. Then* $\dot{h}(x,u) = \nabla h(x) f(x) + \nabla h(x) g(x) u$*. Thus, the term* $\nabla_v h(q,v) g_2(q)$ *in* (6.7) *can be replaced with* $\nabla h(x) g(x)$ *in the more general model. However, in this case, the argument in Remark 6.3 about* $\nabla_v h(q,v) g_2(q) \equiv \nabla_q \kappa(q) g_1(q) g_2(q)$ *being independent of the velocities* v *does not hold.*

Referring to Example 6.1, Theorems 6.7-6.8 address the problem of determining a VCIS $\mathcal{A}$ when $\gamma \in (0, 1]$. However, how to address Example 6.1 when $\gamma > 1$ still needs to be determined, as is done in Section 6.3.3.

Example 6.3. *Consider the problem in Example 6.1 with* $\gamma = 0.75$*. The set* $\mathcal{U}' = \{u \in \mathbb{R}^2 \mid \|u\|_1 \le \frac{2}{1+\sqrt{2}}\}$ *has the QEP with respect to* $\mathcal{U}$ *as in Example 6.1. Using the new set* $\mathcal{U}'$ *as the assumed allowable input bounds, the method in Section 4.3.1 yields the CBFs* $h_i(q,v) = \dot{\kappa}_i(q) - \sqrt{-\frac{4}{1+\sqrt{2}} \kappa_i(q)}$*. Unlike in Example 6.1, the new set* $\mathcal{A} = \mathcal{S} \cap \mathcal{H}_1 \cap \mathcal{H}_2$ *is indeed a VCIS. The impact of constructing the CBFs for* $(\mathcal{S}, \mathcal{U}')$ *instead of* $(\mathcal{S}, \mathcal{U})$ *is visualized in the left half of Fig. 6.5.*

6.3.2 Modifying the Quadratic Program

The work in the prior section required Definition 6.4 to apply to all states in $\mathcal{S}$. Now, recall that Lemma 2.7 is effectively only a condition on the boundary of the invariant set $\mathcal{A}$. Noting this, in Section 6.3.3, I will focus only on the boundary of $\mathcal{A}$, so unlike in Section 6.3.1, I will no longer

be able to guarantee that $\mathcal{U}_{\text{cbf,all}}(q,v)$ is nonempty for $(q,v) \in \text{int}(\mathcal{A})$. Rather, I will only be able to guarantee that there exists $u \in \mathcal{U}$ such that $\dot{h}_k(q,v,u) \leq 0$ for all k in the *active set*

$$\mathcal{I}(q,v) = \{k \in [M] \mid h_k(q,v) = 0\}. \tag{6.11}$$

That is, after employing the algorithm in Section 6.3.3, the condition $\dot{h}_k(q,v,u) \leq \alpha_k(-h_k(q,v))$ in (6.10b) will be feasible for all $k \in \mathcal{I}(q,v)$, but possibly not for $k \in [M] \setminus \mathcal{I}(q,v)$. By [76], if $\mathcal{A}$ in Lemma 6.2 is a VCIS, then there exists a set of class-$\mathcal{K}_e$ functions $\{\alpha_k^*\}_{k\in[M]}$, such that (6.10) is always feasible. Instead of computing such a set directly, I let $\alpha_k^*(\lambda) = \delta_k \alpha_k(\lambda)$, where δ_k is a free variable. I then let $J_k > 0$, and modify the QP (6.10) as in [262] to

$$u(q,v) = \underset{u\in\mathcal{U},\delta_k\geq 1}{\arg\min} \|u - u_{\text{nom}}(q,v)\|_2^2 + \sum_{k\in[M]} J_k\delta_k \tag{6.12a}$$

$$\text{s.t. } \dot{h}_k(q,v,u) \leq \delta_k\alpha_k(-h_k(q,v)), \forall k \in [M] \tag{6.12b}$$

Theorem 6.9. *Suppose the conditions of Lemma 6.2 hold. Then 1) the control law* (6.12) *will render $\mathcal{A}$ forward invariant for as long as* (6.12) *is feasible, 2)* (6.12) *is feasible for all $(q,v) \in \mathcal{A}$ for which $\mathcal{I}(q,v)$ has cardinality of 0 or 1, and 3)* (6.12) *is feasible for all $(q,v) \in \mathcal{A}$ if $\mathcal{A}$ is a VCIS.*

The proof follows from [262].

Remark 6.5. *In light of Theorem 6.9, the concept of non-interference in Definition 6.4 only needs to apply to states (q,v) where the boundaries of two CBF sets intersect. That is, if one uses* (6.12) *instead of* (6.10)*, then the requirement in Theorems 6.7-6.8 that all the CBFs be non-interfering on all of $\mathcal{S}$ can be relaxed to only require CBFs be non-interfering where their CBF set boundaries intersect (if they intersect at all). However, one generally does not know what the CBF sets are until all the CBFs have been designed, so control design is easier if the CBFs (and the constraint functions from which they were designed—see Remark 6.3) are non-interfering everywhere in $\mathcal{S}$.*

Note also that Theorem 6.9 does not make use of Definition 6.4, or any of the other work in Section 6.3.1. That is, this QP can be used regardless of whether the CBFs are interfering or non-interfering.

6.3.3 When the CBFs are Interfering

Next, consider what happens in Example 6.3 if $\gamma > 1$.

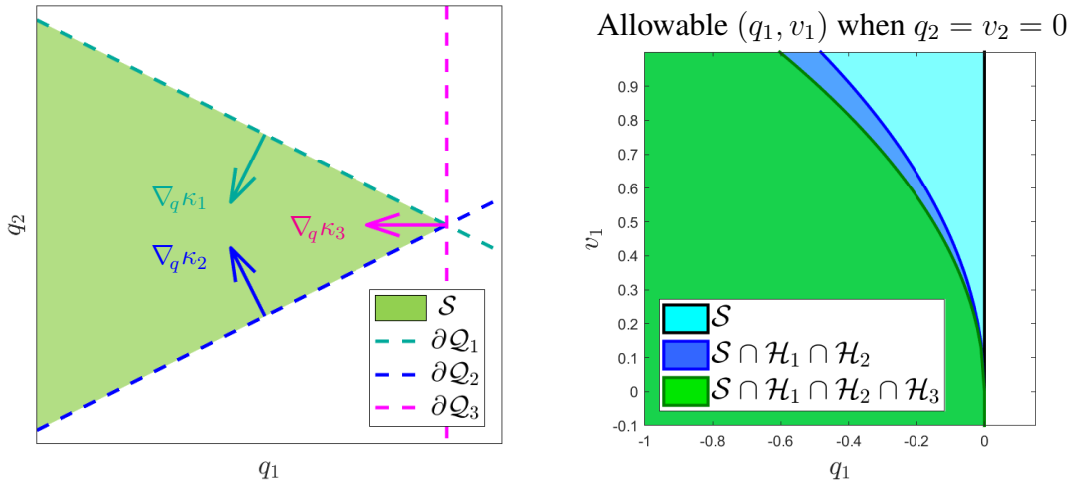

Figure 6.6: Modification of Figs. 6.1,6.5 After Adding a CBF.
Left: The CBFs h_1, h_2 do not satisfy Definition 6.4, so Theorems 6.7-6.8 do not apply. To remedy this, I add a third constraint κ_3 and associated CBF h_3 to remove the remaining states where (6.12) is infeasible. Note that $\nabla_q \kappa_k \equiv \nabla_v h_k$ for this system.
Right: Fixing $q_2 = v_2 = 0$, this plot show the allowable states (q_1, v_1) according to 1) the safe set (cyan), 2) the CBFs in Example 6.4 (blue), and 3) the CBFs in Example 6.5 (green).

Example 6.4. *Consider the problem in Example 6.3 for $\gamma = 1.25$. Using the same strategy as in Example 6.3 yields the new CBFs $h_i(q,v) = \dot{\kappa}_i(q,v) - \sqrt{-\frac{4\gamma}{1+\sqrt{2}}\kappa_i(q)}$. Denote $\mathcal{A} = \mathcal{S} \cap \mathcal{H}_1 \cap \mathcal{H}_2$ and let $x_0 = (q,v) = (-\frac{1+\sqrt{2}}{4\gamma}, 0, 1, 0) \in \partial\mathcal{A}$. From the right side of Fig. 6.5, one can see that there is no $u \in \mathcal{U}$ that satisfies both CBF conditions at x_0. That is, Nagumo's necessary condition for forward invariance of $\mathcal{A}$ is violated at x_0, so $\mathcal{A}$ is not a VCIS.*

The difference between Example 6.3 and Example 6.4 is that in Example 6.4, the CBFs h_1, h_2 do not satisfy Definition 6.4, so Theorems 6.7-6.8 do not apply. Thus, there is a need for additional tools to systematically remove states such as x_0 in Example 6.4 from $\mathcal{A}$ in such cases. I begin by presenting a solution to this simple example before discussing a general algorithm.

Example 6.5. *Consider the problem in Example 6.4. Introduce $\kappa_3(q) = q_1$, resulting in the constraint geometry in Fig. 6.6. Let $\mathcal{H}_1$ and $\mathcal{H}_2$ be as in Example 6.4, and using Section 4.3.1, one can derive the CBF $h_3(q,v) = \dot{\kappa}_3(q,v) - \sqrt{-\frac{4}{1+\sqrt{2}}\kappa_3(q)}$. Then $\mathcal{A} = \mathcal{S} \cap \mathcal{H}_1 \cap \mathcal{H}_2 \cap \mathcal{H}_3$ is a VCIS for the sets $\mathcal{S}$ and $\mathcal{U}$ (see the green set on the right side of Fig. 6.6).*

Example 6.5 improves upon Example 6.4 by further limiting the system's velocity v_1 so as to remove all the points where (6.12) is infeasible. This can be done either by adjusting h_1 and h_2, or by adding the additional CBF h_3. The first approach amounts to a joint tuning of all CBFs, which is challenging, so I instead focus on generalizing the latter strategy.

Algorithm 1 Get VCIS

Require: $\mathcal{S}$ in (6.4), $\mathcal{U}'$ possessing QEP (or OEP) w.r.t. $\mathcal{U}$, CBFs (or SCBFs) $\{h_k\}_{k=1}^{M_0}$ for $(\mathcal{S},\mathcal{U}')$ satisfying Lemma 6.2

$\mathcal{X} \leftarrow \mathcal{S} \cap (\cap_{k=1}^{M} \mathcal{H}_k)$
$\mathcal{D} = \cup_{i\in[M_0],j\in[M_0],i\neq j}(\partial\mathcal{H}_i \cap \partial\mathcal{H}_j)$
$M \leftarrow M_0$
$\mathcal{E} \leftarrow$ getInfeasibleSet$(\mathcal{D})$
while $\mathcal{E} \neq \emptyset$ **do**
    $\mathcal{E}_c \leftarrow$ getCluster$(\mathcal{E})$
    $h_{M+1} \leftarrow$ getCBF$(\mathcal{E}_c, \mathcal{X}, \mathcal{U}')$
    $\mathcal{X} \leftarrow \mathcal{X} \cap \mathcal{H}_{M+1}$
    $M \leftarrow M + 1$
    $\mathcal{D} = \cup_{i\in[M],j\in[M],i\neq j}(\partial\mathcal{H}_i \cap \partial\mathcal{H}_j)$
    $\mathcal{E} \leftarrow$ getInfeasibleSet$(\mathcal{D})$
end while
$\mathcal{A} \leftarrow \mathcal{X}$
return $\mathcal{A}$, $\{h_k\}_{k=1}^{M}$

To this end, I propose Algorithm 1. Here, $\{h_k\}_{k\in[M]}$ is a working set of CBFs and $\mathcal{X}$ is a working domain. Let $\mathcal{D}$ be a set of candidate points where (6.12) could be infeasible, namely all points where at least two CBFs are active as in (6.11), as Theorem 6.9 guarantees feasibility of (6.12) at all other points. Next, let $\mathcal{E}$ be the subset of $\mathcal{D}$ where (6.12) is actually infeasible, where $\mathcal{E}$ is more expensive to compute than $\mathcal{D}$. Next, let getCluster() be a function that divides all the points in $\mathcal{E}$ into clusters (e.g. see Fig. 6.7), and then returns a single cluster $\mathcal{E}_c \subseteq \mathcal{E}$ for focus. Given this cluster, the function getCBF() then determines a new CBF h_{M+1} for $(\mathcal{X},\mathcal{U}')$ such that the cluster $\mathcal{E}_c$ is entirely outside the CBF set $\mathcal{H}_{M+1}$. By Assumption 6.1, such a CBF h_{M+1} will always exist. Note also the use of the set $\mathcal{U}'$ satisfying Definition 6.5 or 6.6 in getCBF(); this is done to reduce the number of points where h_{M+1} might conflict with the existing CBFs $\{h_k\}_{k\in[M]}$. Algorithm 1 then adds h_{M+1} to the working set of CBFs and shrinks the working domain to $\mathcal{X} \cap \mathcal{H}_{M+1}$ (e.g. see Fig. 6.6). The **while** block then repeats until either all the CBFs are jointly feasible or the domain $\mathcal{X}$ becomes empty.

Note that the exact mechanics of grouping points into clusters during getCluster() and of generating CBFs during getCBF() will be specific to the system under study. An example of these steps is described in Section 6.4.

Proposition 6.10. *If Algorithm 1 converges, then $\mathcal{A}$ is a VCIS as in Definition 6.3 and the controller* (6.12) *is always feasible and renders $\mathcal{A}$ forward invariant.*

Algorithm 1 is similar to typical viability domain computation algorithms [186–188, 257–260],

except that the set $\mathcal{A}$ is parameterized by a collection of CBFs. Since these CBFs are defined for a control set $\mathcal{U}'$ possessing the QEP (or OEP), one can reduce the number of points that one needs to check for infeasibility to only the points where 1) multiple CBFs are active as in (6.11) and 2) the active CBFs violate the condition in Definition 6.4. I next illustrate the implementation of Algorithm 1 by example on a nonlinear system.

Remark 6.6. *The choice of the functions getCluster() and getCBF() in Algorithm 1 will also affect the conservativeness of the resulting set $\mathcal{A}$ compared to the viability kernel, and the time of convergence of the algorithm. A very small cluster size could result in many added CBFs and an explosion in computation. Note also that Algorithm 1 can be either coded and run autonomously, or followed step-by-step "by hand" by a practiced engineer.*

6.4 Application to Orientation Control

Let $q \in \{q \in \mathbb{R}^4 \mid \|q\|_2 = 1\}$ be the quaternion describing the orientation of a spherically symmetric rigid body with unit moments of inertia, and let $\omega = (\omega_1, \omega_2, \omega_3) \in \mathbb{R}^3$ be the angular velocity of the body. The dynamics are

$$\dot{q} = \frac{1}{2}\begin{bmatrix} 0 & \omega_3 & -\omega_2 & \omega_1 \\ \omega_3 & 0 & \omega_1 & \omega_2 \\ \omega_2 & -\omega_1 & 0 & \omega_3 \\ -\omega_1 & -\omega_2 & -\omega_3 & 0 \end{bmatrix} q,\ \dot{\omega} = u\,. \tag{6.13}$$

Suppose a sensitive instrument faces towards an axis $\hat{a}$ fixed to the body, and must avoid pointing towards the fixed directions $\hat{b}_1, \hat{b}_2$ in Fig. 6.7 by angles θ_1, θ_2 (red cones in Fig. 6.7). These constraints can be expressed as $\kappa_1(q) = q^{\mathrm{T}} P(\hat{a}, \hat{b}_1, \theta_1) q$ and $\kappa_2(q) = q^{\mathrm{T}} P(\hat{a}, \hat{b}_2, \theta_2) q$, where P is constant with respect to the state (q, ω) and is constructed as in [221] and Section 5.2.2.2. Let $\mathcal{U} = \{u \in \mathbb{R}^3 \mid \|u\|_\infty \leq u_{\max}\}$ for some $u_{\max}$. Assume the angular velocity is bounded by $\eta(\omega) = \|\omega\|_\infty - \omega_{\max}$, where the ∞-norm can be expressed as 6 continuously differentiable constraint functions $\{\eta_j\}_{j\in[6]}$.

First, I find a set $\mathcal{U}'$ with the OEP with respect to $\mathcal{U}$. Example 6.2 implies that $\mathcal{U}' = \{u \in \mathbb{R}^3 \mid \|u\|_2 \leq \frac{u_{\max}}{\sqrt{3}}\}$ is one such set, from which I derive SCBFs $h_{\kappa,1}, h_{\kappa,2}$ for $(\mathcal{S}, \mathcal{U}')$ given by $h_{\kappa,i}(q,\omega) = \dot{\kappa}_i(q,\omega) - \sqrt{-\beta_i \kappa_i(q)}$ for some $\beta_i \in \mathbb{R}_{>0}$. Note that each η_j already satisfies the definition of an SCBF on $(\mathcal{S}, \mathcal{U}')$, so denote these SCBFs as $h_{\eta,j} \equiv \eta_j$. The eight SCBFs together satisfy the conditions of Lemma 6.2. I now focus only on the $h_{\kappa,1}, h_{\kappa,2}$ SCBFs. If the corresponding sets $\mathcal{H}_{\kappa,1}, \mathcal{H}_{\kappa,2}$ satisfy $\partial\mathcal{H}_{\kappa,1} \cap \partial\mathcal{H}_{\kappa,2} = \emptyset$, then $\mathcal{A}$ in Lemma 3 is a VCIS and no further analysis is required. Here, I chose θ_1, θ_2 sufficiently large that this does not hold, as indicated by the

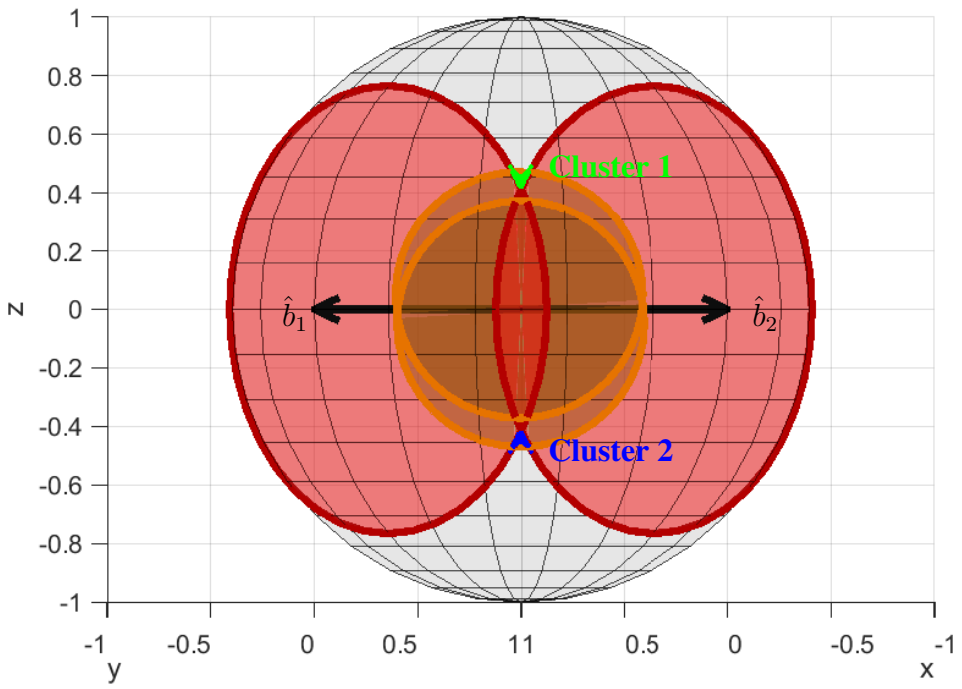

Figure 6.7: Visualization of Two Simultaneous Pointing Constraints. Visualization of the constrained reorientation problem studied in Section 6.4. The large red cones are the initial unsafe states (where the body cannot point body-fixed vector $\hat{a}$), the two clusters are the states where (6.12b) is infeasible, and the brown cones are the states excluded by the new CBFs introduced by Algorithm 1.

intersection of the red cones in Fig. 6.7. All code and parameters discussed in this subsection can be found at https://github.com/jbreeden-um/phd-code/tree/main/2023/ACC%20Multiple%20Control%20Barrier%20Functions.

I then apply Algorithm 1. For this system, I coded getCluster() to return connected subsets of $\mathcal{E}$. In this case, getInfeasibleSet() returned states $(q, v) \in \mathbb{R}^7$ with quaternions q corresponding to the green and blue chevrons in Fig. 6.7, and getCluster() identified these points as two clusters. I then coded getCBF() to remove the cluster states by introducing a new constraint of the form $\kappa_{M+1}(q) = q^{\mathrm{T}} P(\hat{a}, \hat{b}_{M+1}, \theta_{M+1}) q$, and computing an SCBF $h_{\kappa,M+1}$ of the same form as $h_{\kappa,1}$ and $h_{\kappa,2}$. The getCBF() function searched for the orientation $\hat{b}_{M+1}$ and minimum angle θ_{M+1} that removed the entire cluster and that satisfied Definition 6.4 when paired with each of the existing SCBFs. Visually, for states with $\omega = 0$, the new SCBF $h_{\kappa,M+1}$ resulted in removing one of the brown cones in Fig. 6.7. After removing the first cluster, Algorithm 1 repeated for a second loop, and then completed and returned the two initial SCBFs (red cones) and two new SCBFs (brown cones) shown in Fig. 6.7.

One can also implement Algorithm 1 by hand as in Remark 6.6. For this system, a practiced engineer might instead decide that both clusters in Fig. 6.7 should be grouped into a single cluster,

which can then be removed by adding a single new SCBF representing a larger cone. Note also that Fig. 6.7 is visually different from Fig. 5.15 in Chapter 5, because Fig. 6.7 depicts one instrument and two unsafe directions, whereas Fig. 5.15 depicts two instruments and one unsafe direction, though both are still multi-CBF scenarios.

Note that the computation time of Algorithm 1 depends most on the computation time of the getInfeasibleSet() step, which has to search a potentially large set for infeasibilities. However, this set is smaller than in typical verification algorithms, because one only has to consider states where the boundaries of two or more CBFs intersect. All other states already satisfy Lemma 2.7 because $\mathcal{X}$ is parameterized by CBFs. Possible implementations of getInfeasibleSet() include [140, 261], and the wider viability theory literature. Since these are primarily sampling-based algorithms, the computation time is dependent on the sampling interval, which depends on the Lipschitz constants of h_k and the margin by which each h_k satisfies (6.5). The latter is equivalent to the chosen amount of conservatism with which the CBFs are implemented; more conservative margins (e.g. smaller β_i) will allow for sparser sampling and quicker computations. In this case, the results in Fig. 6.7 took 1052 seconds to compute, and results with a 10x coarser sampling took 0.30 seconds to compute, using a 3.5 GHz processor.

6.5 Conclusions on Multiple CBFs

This chapter identified and partially addressed the challenge of designing controllers that are everywhere feasible in the presence of multiple CBFs. After presenting this challenge and two set invariance theorems utilizing multiple CBFs, this work proposed the definition of "non-interfering CBFs". I then showed how geometric observations following from this definition allow one to guarantee that a set of CBFs will be jointly feasible, while allowing each CBF to be designed in a one-at-a-time fashion under more conservative assumptions on the available control authority. However, if any of the CBFs are not "non-interfering", then it is still possible for there to exist points where an optimization-based controller with multiple constraints may become infeasible. Thus, I also introduced an algorithm to iteratively remove such points from the allowable state set using additional CBFs until this set becomes a controlled invariant set, so that a CBF-based QP controller is always feasible. Future work in this area may include robust and sampled-data extensions of this framework, generalizations of the observations about relative-degree 2 systems to other classes of systems, and extensions to relate this algorithm to the maximal viability kernel. One might also wish to further refine the proposed algorithm so as to have provable convergence guarantees.

It is also worth considering how all the work in this dissertation prior to this chapter was successful in generating safe trajectories, even though this earlier work did not consider the possibility

of infeasibility of the QP when multiple CBFs were employed simultaneously. Similar to the conclusions in Chapter 5, the above work represents a lot of effort devoted to a problem that may or may not occur. While it is good to understand this problem, it may not be essential to consider these multi-CBF factors at all phases of controller design. For instance, when doing a "rough draft" controller design, a controls engineer may choose to ignore these factors in order to speed up the controller design process. It is also possible that the problems identified in Example 6.1 and Example 6.4 can be removed through tuning. To see why this is, note how the only states at which (6.12) can become infeasible are states that are in the boundary of at least two CBF sets. If the α_k functions are chosen with small slopes, then it will take a long time for trajectories to reach this boundary. Intuitively, if the unsafe regions are obstacles, then by the time a system trajectory reaches the edge of an obstacle, the trajectory may have already passed the obstacle. Thus, unless the system's nominal objectives try to drive the trajectory towards states where multiple CBF set boundaries intersect, it is often the case that trajectories never encounter such states and thus the controller (6.12) is feasible along these specific trajectories. In this way, tuning of the class-$\mathcal{K}_e$ functions α_k can be used to "hide" such infeasibilities. One can see this for the example in Fig. 6.7, where the two clusters are a very small part of the total safe set. Given how small these clusters are, it is unsurprising that these states where the control law (5.115) would have been infeasible were never encountered in Section 5.2.5.1. That said, if the nominal control law does tend to drive trajectories towards these infeasible states, then the control law (6.12) may allow trajectories to reach the boundaries of the CBF sets in finite time (where increasing J_k will tend to increase the amount of time); i.e. even if α_k are locally Lipschitz continuous, the product $\delta_k \alpha_k$ in (6.12) is not necessarily locally Lipschitz continuous, so trajectories can still reach states at the boundary of one or more CBF sets. In conclusion, I suggest that it is equally important to 1) understand how to solve the multi-CBFs problem, and 2) understand for what scenarios this problem is likely to occur.

CHAPTER 7

A Predictive Control Barrier Function for Time-Varying Applications

In this chapter, I return to the problem of designing control barrier functions for constraints of high relative-degree with relevance to spacecraft problems. In particular, one problem not previously addressed is the problem of safety in constellations of satellites in differing orbits. When satellites are in similar enough orbits to be approximated by the HCW dynamics [8] (or possibly the ellipsoidal extensions of the HCW dynamics, e.g. [11–13]), then tools such as those in Chapters 3-4 apply well. However, satellites can still occupy the same positions even when their orbits (i.e. their Keplerian elements) are very different. Conversely, satellites that are at one time instant close to each other may live in orbits with a near zero risk of collision, thus leading the approaches in Chapters 3-4 to be extremely over-conservative, as is illustrated in simulation in Section 7.1.4. Thus, the motivation for this chapter is how to design a good CBF for satellites in dissimilar orbits.

The proposed strategy is to make a prediction of where satellites will be over a finite horizon, and to determine the minimum distance between the satellites during that horizon. I then encode this minimum predicted distance in a CBF. In addition, since this distance may be less than the required separation distance between satellites, I add to the CBF a relaxation term that depends on how much time is predicted until the satellites first become unsafe (i.e. the time instant when the required separation distance is first violated). I can then program a QP controller similar to (2.9) that ensures that the state always stays in the zero sublevel set of this new CBF, which Theorem 7.2 shows is sufficient for safety. The simulations in Section 7.1.4 show that this is a very promising strategy. In fact, I show how it is promising not just for spacecraft problems, but for other collision avoidance problems, like car intersections. That said, the tradeoff to this strategy is that it does not easily allow for provable input constraint satisfaction.

This chapter is organized into two parts. The first part presents the analysis required to produce the "path predictive CBF". Like the backup CBF in Section 3.3.1 and Section 4.3.3, this is general form of CBF that can be applied to any system meeting the theorem requirements. That said, I emphasize that the path predictive CBF is an entirely different sort of CBF from the backup

CBF, even though the backup strategy has been called a "predictive CBF" elsewhere in the CBF literature, e.g. [99]. The key difference is that the backup CBF tests whether a provided safe control law (the "backup control law") can render a trajectory safe and thus constraints trajectories only to states where switching to the backup law would lead to safety. By contrast, the path predictive CBF tests whether any provided control law is safe on a horizon, and if not, the QP controller uses the gradients of the path predictive CBF to *find* a safe control trajectory. This act of *finding* a safe trajectory is why it is difficult to apply the path predictive CBF with input constraints. In the tutorial [181], I also termed this approach a "bird's eye CBF", by analogy to these predictions giving the controller a "bird's eye view" of the environment and potential future obstacles. Alternatively, I invite future researchers to call this method "Joseph's CBF" (to my knowledge, the acronym JCBF is not yet taken) or the "Breeden-Black-CBF" in recognition of work done independently at the same time in [264].

Note that this chapter constitutes some of the later work done within this dissertation and is not entirely finished. I had hoped to integrate the path predictive CBF with constructive input constraint satisfaction and with multi-CBF considerations, and to develop relaxations of some of the path predictive CBF assumptions. However, I now believe that accomplishing all of these objectives could constitute over half of a second dissertation. Thus, at the end of this chapter, I present some criticisms of this approach that may be non-obvious. I am actively working on extensions that address many of these questions, but was not able to include these in this dissertation. Solutions to these questions may appear soon in other academic literature.

7.1 A General Form of CBF for Collision Avoidance Problems

7.1.1 Introduction

Control Barrier Functions (CBFs) are a tool for achieving safe control of state-constrained systems in an online fashion [66]. Specifically, CBFs are functions of the system state that motivate an affine condition on the instantaneous control input that, if satisfied pointwise, guarantees forward invariance of a given set of *safe* states. I call this condition the *CBF condition* and this set the *safe set*. For example, the state of a car on a shared road should belong to the set of states that are a minimum separation distance from every other car.

However, one drawback of CBFs is that the CBF condition is reactive (frequently called *myopic* [142]); i.e., CBFs only consider the safety of the current state and state derivative, rather than considering future objectives of the system. For this reason, the CBF condition may allow trajectories to reach states where large control inputs are required to maintain safety [142, 148], or when control inputs are constrained, CBF safe sets are often overly conservative, requiring large evasive

actions (e.g., the wide arcs around Ceres in Section 4.5.3). For instance, CBFs may cause two cars meeting at an intersection to both decelerate to a complete stop to maintain safety, whereas it would be more efficient for only one car to decelerate slightly, allowing both cars to pass without stopping. To address this and similar situations, I propose the notion of Path Predictive CBFs (P^2CBFs).

Intuitively, P^2CBFs are functions that encode both the present and future safety of the system in a model-predictive manner, while still providing the convenient CBF condition on the current control input. Specifically, the method in this section supposes a known nominal trajectory, herein called a *path*, not necessarily safe, that the agent is to follow, and computes a measure of the safety of the path on a receding horizon. This measure is then encoded into a special structure of CBF. If the safety measure is positive (i.e. unsafe), then the resultant CBF condition causes the controller to modify the nominal control input so as to render the trajectory safe.

P^2CBFs are closely related to Model Predictive Control (MPC) [98, 99, 128, 141, 148, 265, 266] and to the notion of Backup CBFs [86, 95–99], which both also consider safety on a receding horizon, but P^2CBFs provide distinct advantages. Unlike MPC, the CBF condition is always control-affine, even when the dynamics and constraints are nonlinear. The CBF condition also only applies to the current control input rather than to a predicted horizon of control inputs. Thus, the control input under a P^2CBF can be computed using a Quadratic Program (QP) with small dimension and affine constraints, which can be solved more easily than nonlinear MPC (e.g. [128, 148]). Moreover, as the P^2CBF approach does not introduce a fixed discretization, safety is guaranteed at all times in the prediction horizon rather than just at the MPC sampled times (this is particularly imporant in the orbital simulations in Section 7.1.4). That said, the P^2CBF does not yet support sampled-data controllers, as does MPC.

Compared to Backup CBFs, the P^2CBF works by propagating the system trajectory according to its nominal objectives, which are allowed to be unsafe, rather than according to some backup safety plan. Thus, a backup controller or backup set does not need to be known a priori [96], and the system trajectories are not limited to evolving near the backup set, which may be overly conservative. Both Backup CBFs and P^2CBFs can be constructed from a safety measure of any relative-degree. That said, unlike Backup CBFs, P^2CBFs do not guarantee input constraint satisfaction, though this is an area for future study.

Another recent approach for making CBFs consider the future system evolution is to introduce an additional constraint that bounds the future control inputs generated by a CBF [90, 267]. I further argue that any Higher Order CBF (e.g., [85, 88] or Chapters 3-4) also implies some degree of look-ahead (see the discussion of *generalized inertia* in Chapter 3), though this look-ahead is implicit in the choice of class-$\mathcal{K}_e$ functions and thus may be difficult to tune. The work in [171] tried to make robust CBFs less conservative by encoding a finite horizon prediction of future dis-

turbances (e.g. road curvature, friction coefficient) into the CBF, and proposed one such form of CBF for linear second-order systems. This work indeed made the CBF less intrusive (while maintaining provable safety) for problems that involved keeping the state within a specified operating tolerance (e.g. minimum/maximum velocity or maximum deviation from center), but this strategy would not make the resultant safety filters more proactive at avoiding future obstacles, as is sought in this work (in fact, making the CBF less conservative could make the safety filter react even later in this case). The authors in [268] also found that learned policies tended to proactively avoid unsafe regions better than CBFs alone. This section supplements all of these prior works by looking ahead only along a specified path rather than along all possible safe trajectories. This work is also closely related to the work in [264], but is generalized to arbitrary dynamics and paths rather than constant velocity vehicles.

This section is organized as follows. Section 7.1.2 introduces preliminaries on CBFs. Section 7.1.3 formalizes the notion of a nominal path (*path function*), introduces a predictive safety function, and proves that this function is a CBF. Section 7.1.4 presents simulations and comparisons that show how this function is proactive. Section 7.1.5 presents concluding remarks and highlights possible deficiencies of this approach.

7.1.2 Preliminaries

In this section, I return to considering a time-varying system of the form

$$\dot{x} = f(t, x) + g(t, x)u \tag{7.1}$$

with time $t \in \mathcal{T} \triangleq [t_0, t_f]$, state $x \in \mathcal{X} \subseteq \mathbb{R}^n$, and control input $u \in \mathbb{R}^m$. Note that the set of control inputs is assumed to be unconstrained in this section (for the first time in this dissertation). Assume that f, g, and u are sufficiently regular that $x(t)$ exists and is unique for all $t \in \mathcal{T}$.

Given a *constraint function* $\kappa : \mathcal{T} \times \mathcal{X} \to \mathbb{R}$, define the *safe set* $\mathcal{S} : \mathcal{T} \to 2^{\mathcal{X}}$ as

$$\mathcal{S}(t) \triangleq \{x \in \mathcal{X} \mid \kappa(t, x) \leq 0\}. \tag{7.2}$$

As in prior chapters, the objective of this chapter is to render trajectories always inside the *safe set*, specifically by rendering a subset of $\mathcal{S}$ forward invariant. To this end, I first generalize the definition of CBF [66, 169] to functions that are differentiable almost everywhere, to account for how the proposed P^{2}CBFs will have discontinuities at the beginning and end of the prediction horizon.

Definition 7.1 (Absolutely Continuous Control Barrier Function)**.** *Let $h : \mathcal{T} \times \mathcal{X} \to \mathbb{R}$ be absolutely continuous (i.e. continuously differentiable except on a set of Lebesgue measure zero) and*

let $\mathcal{H}$ and $\mathcal{H}^{\mathcal{T}}$ be as in (2.2)-(2.3)*. The function h is a* Control Barrier Functions (CBF) *for the system* (7.1) *if there exists a set $\mathcal{D} : \mathcal{T} \to 2^{\mathcal{X}}$ and a function $\alpha \in \mathcal{K}_e$ such that $\mathcal{H}(\tau) \subset \mathcal{D}(\tau)$ for all $\tau \in \mathcal{T}$ and*

$$\inf_{u \in \mathbb{R}^m} \Bigg[\underbrace{\frac{\partial h(t,x)}{\partial t} + \frac{\partial h(t,x)}{\partial x}\big(f(t,x) + g(t,x)u\big)}_{= \frac{d}{dt}[h(t,x)]} \Bigg] \leq \alpha(-h(t,x)) \tag{7.3}$$

for almost every $(t,x) \in \mathcal{H}^{\mathcal{T}}$.

One can then establish forward invariance of the CBF set $\mathcal{H}^{\mathcal{T}}$ similarly to prior chapters as follows.

Lemma 7.1. *Given a CBF $h : \mathcal{T} \times \mathcal{X} \to \mathbb{R}$ as in Definition 7.1, let $\mathcal{D} : \mathcal{T} \to 2^{\mathcal{X}}$ be an open set such that $\mathcal{H}(\tau) \subset \mathcal{D}(\tau)$ for all $\tau \in \mathcal{T}$, let $\mathcal{D}^{\mathcal{T}} = \{(t,x) \in \mathcal{T} \times \mathcal{X} \mid x \in \mathcal{D}(t)\}$, and let $\alpha \in \mathcal{K}_e$. Assume that for every trajectory $x(\mathcal{T})$, it holds that h is continuously differentiable almost everywhere on the curve $x(\mathcal{T})$ (i.e. this is a stricter condition that h is not only differentiable almost everywhere in $\mathcal{X}$ (of dimension n), but is furthermore differentiable almost everywhere on every feasible trajectory along $\mathcal{T}$ (of dimension 1) of* (7.1)*). Then any control law $u : \mathcal{T} \times \mathcal{X} \to \mathbb{R}^m$ that satisfies $u(t,x) \in K_{\text{cbf}}(t,x)$ for almost every $(t,x) \in \mathcal{H}^{\mathcal{T}}$, where*

$$K_{\text{cbf}}(t,x) \triangleq \left\{ u \in \mathbb{R}^m \mid \frac{\partial h(t,x)}{\partial t} + \frac{\partial h(t,x)}{\partial x}\big(f(t,x) + g(t,x)u\big) \leq \alpha(-h(t,x)) \right\}, \tag{7.4}$$

will render $\mathcal{H}^{\mathcal{T}}$ forward invariant.

Proof. The proof is identical to the proof of Theorem 2.2. ■

Note that Lemma 2.3 and Theorem 2.2 (invoked in Lemma 7.1) generalize [169, Lemma 1] and [169, Thm. 1], respectively, to allow for non-Lipschitz α. These extensions are important, because the main theorem of this section, Theorem 7.5, will need to make use of non-Lipschitz α functions for one of the proof cases.

Remark 7.1. *Note in particular the assumption in Lemma 7.1 about h being continuously differentiable almost everywhere along every feasible $x(\mathcal{T})$. To see why this is important, consider $x \in \mathbb{R}$ with dynamics $\dot{x} = u$ and the CBF $h(t,x) = |t - x|$. This h is absolutely continuous, because the manifold $\mathcal{X}_{\text{non-diff}} = \{(t,x) \in \mathbb{R}^2 \mid t = x\}$ is of Lebesgue measure zero in $\mathbb{R}^2$. For initial condition $x(0) = 0$ and control input $u \equiv 1$, the resulting trajectory evolves entirely along $\mathcal{X}_{\text{non-diff}}$. In this case, the function θ constructed in the proof of Theorem 2.2 is continuously differentiable nowhere along this trajectory. Thus, the assumption that h is continuously differentiable almost everywhere along each $x(\mathcal{T})$ is necessary to apply the logic of Theorem 2.2 to absolutely continuous CBFs.*

That said, even though $u \equiv 1$ does not meet the assumptions of Lemma 7.1, this choice of control law indeed renders $\mathcal{H}$ as above forward invariant. Intuitively, this is what one would expect from any system whose trajectories depend continuously on the system's initial conditions (see [71, Thm. 3.4]). Thus, I hypothesize that with the addition of some regularity assumptions on f, g, and u (e.g. local Lipschitz continuity) that this assumption on the trajectories $x(\mathcal{T})$ can be removed, though proving this is beyond the scope of this dissertation.

I am now ready to begin the presentation of the main topic of this chapter.

7.1.3 Predictive Control Barrier Functions

The P^{2}CBF has three components: a nominal trajectory (Section 7.1.3.1), points of interest on this trajectory (Section 7.1.3.2), and the function structure (Section 7.1.3.3). After introducing all three parts, I then analyze the the P^{2}CBF zero-sublevel set and present a computational lemma required for implementation. The main theoretical result of this section then shows that the P^{2}CBF satisfies Definition 7.1. I conclude the subsection with a discussion of the main theorem assumptions.

7.1.3.1 Path Functions

Suppose that an agent seeks to follow a nominal trajectory through $\mathcal{X}$, which is not necessarily safe. I call this trajectory a *path*, and assume that all potential paths of the system can be described by a *path function*, defined as follows.

Definition 7.2 (Path Function)**.** *A function $p : \mathcal{T} \times \mathcal{T} \times \mathcal{X} \to \mathcal{X}$, denoted $p(\tau, t, x)$, is called a* path function *if 1) it is continuously differentiable in all inputs, 2) $p(t; t, x) = x, \forall (t, x) \in \mathcal{T} \times \mathcal{X}$, and 3) for every $(t, x) \in \mathcal{T} \times \mathcal{X}$, there exists a control law $\mu : \mathcal{T} \times \mathcal{X} \to \mathbb{R}^m$, denoted $\mu(t, x)$, that satisfies*

$$\frac{\partial p(\tau, t, x)}{\partial \tau} = f(\tau, p(\tau, t, x)) + g(\tau, p(\tau, t, x))\mu(\tau, p(\tau, t, x)),\ \forall \tau \geq t\,. \tag{7.5}$$

That is, p is a function that from every (t, x) generates a trajectory $p(\tau, t, x)$ over future times τ starting from state x, where p must be feasible according to the dynamics (7.1). Note that τ here is defined as absolute time in this section, but τ could be equivalently defined as time since t (e.g. as in β in Section 4.3.3). Since generated paths $p(\tau, \cdot, \cdot)$ are not required to be safe, the following methodology decouples safety from the computation of p, and thus enables the use of very simple nominal paths. That is, one does not need to know a safety-encouraging backup law as in Section 4.3.3; indeed, one possible choice of p is the trajectory following from $\mu(t, x) \equiv 0$.

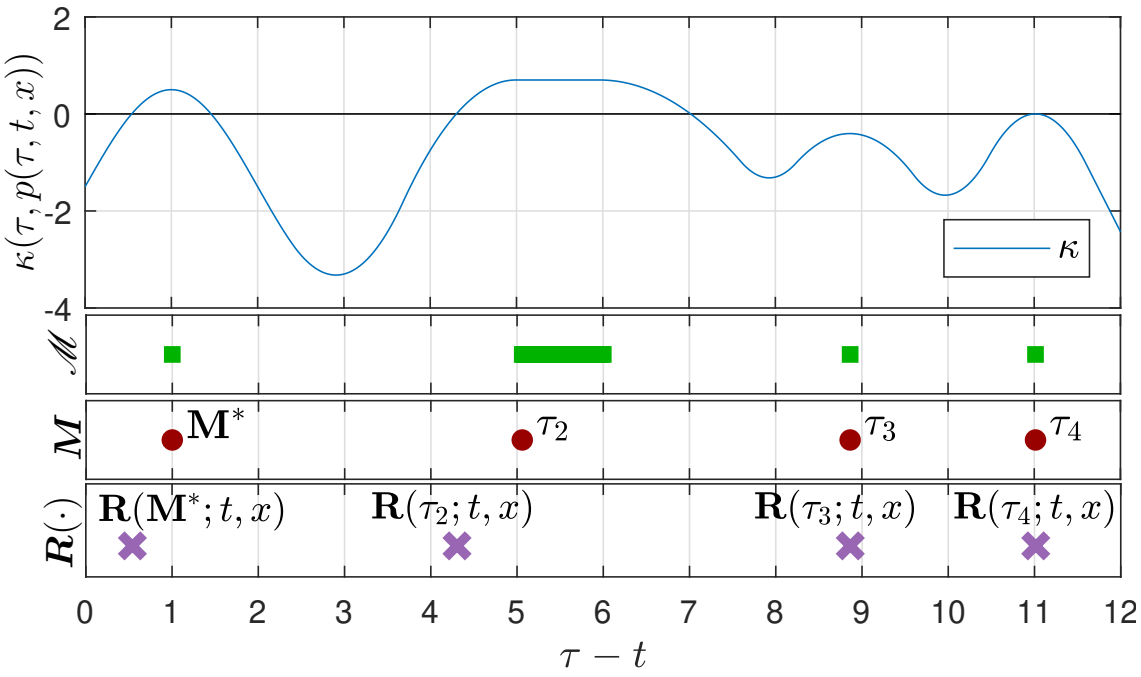

Figure 7.1: Illustration of Times of Interest on Propagated Trajectories. A sample trajectory of $\kappa(\tau, p(\tau, t, x))$ with the corresponding sets $\mathscr{M}$ and $\boldsymbol{M}$. The value of $\boldsymbol{R}$ corresponding to each element of $\boldsymbol{M}$ is also shown. The set of all local maximizers $\mathscr{M}$ is dense, so $\boldsymbol{M}$ extracts one element from every isolated subset of $\mathscr{M}$. Each element of $\boldsymbol{M}$ and its corresponding root $\boldsymbol{R}$ is then used to construct (7.12)-(7.13).

Remark 7.2. *Path functions can be defined either geometrically (as in Section 7.1.4.2), provided a feasible $\mu(t, x)$ in* (7.5) *exists, or as the explicit (as in Section 7.1.4.1) or numerical (as in* (3.22) *or [96, Eq. 23]) solution to the differential equation* (7.5) *given a nominal input $\mu(t, x)$. Note that while some of the following techniques appear complicated, they can be implemented straightforwardly using numerical solutions to* (7.5).

7.1.3.2 Future Times of Interest on the Propagated Trajectory

Given a path function, one can examine the safety of the agent forward in time along that path. To do this, I assume that the agent knows both the current and future construction of its safe set (e.g. future locations of obstacles) on a finite receding horizon $T > 0$, so that the agent is able to compute κ across this horizon. Given this, I then reduce the continuous trajectory $p(\tau, \cdot, \cdot)$ to a finite number of "times of interest", namely the maximizers and roots of $\kappa(\tau, p(\tau, \cdot, \cdot))$. These times are shown for a sample trajectory in Fig. 7.1, and defined mathematically as follows. First, define the set of all local maximizers within the horizon (for a specific path), $\mathscr{M} : \mathcal{T} \times \mathcal{X} \to 2^{\mathcal{T}}$,

as

$$\mathscr{M}(t,x) \triangleq \{\tau \in [t,t+T] \mid \exists \epsilon > 0 : \forall \sigma \in [\tau-\epsilon,\tau+\epsilon] \cap [t,t+T], \\ \kappa(\tau, p(\tau,t,x)) \geq \kappa(\sigma, p(\sigma,t,x))\}\,. \quad (7.6)$$

Note that the set $\mathscr{M}$ could be uncountable, so next define the countable subset $\boldsymbol{M} : \mathcal{T} \times \mathcal{X} \to 2^{\mathcal{T}}$ as

$$\boldsymbol{M}(t,x) \triangleq \{\tau \in \mathrm{cl}(\mathscr{M}(t,x)) \mid \tau = t \text{ or } \exists \epsilon > 0 : \sigma \notin \mathscr{M}(t,x), \forall \sigma \in (\tau-\epsilon,\tau)\}, \quad (7.7)$$

That is, $\boldsymbol{M}$ contains all the isolated elements of $\mathscr{M}$ and the first element of every interval in $\mathscr{M}$, as shown in Fig. 7.1. This extra step is needed to ensure that $\boldsymbol{M}$ has finitely many elements, so that $\boldsymbol{M}$ is tractable for computations. Additionally, let $\boldsymbol{M}^* : \mathcal{T} \times \mathcal{X} \to \mathbb{R}$, defined as

$$\boldsymbol{M}^*(t,x) \triangleq \min \boldsymbol{M}(t,x)\,, \quad (7.8)$$

denote the first element of $\boldsymbol{M}$, since I am most interested in the soonest future time where the trajectory could leave the safe set. Next, it is not enough to know just the maximizer times; the proposed methodology also needs to know when the predicted paths first become unsafe (if ever), i.e. the roots of $\kappa(\tau, p(\tau,\cdot,\cdot))$ over $\tau \in [t,t+T]$, so define $\boldsymbol{R} : \mathcal{T} \times \mathcal{T} \times \mathcal{X} \to \mathbb{R}$ as

$$\boldsymbol{R}(\tau,t,x) \triangleq \begin{cases} \boldsymbol{R}_1(\tau,t,x) & \kappa(\tau,p(\tau,t,x)) > 0 \\ \tau & \kappa(\tau,p(\tau,t,x)) \leq 0 \end{cases}, \quad (7.9)$$

$$\boldsymbol{R}_1(\tau,t,x) \triangleq \max\Big\{\eta \in [t,\tau] \mid \kappa(\eta,p(\eta,t,x)) = 0 \text{ and} \\ \exists \epsilon > 0 : \kappa(\sigma,p(\sigma,t,x)) < 0, \forall \sigma \in (\eta-\epsilon,\eta)\Big\}. \quad (7.10)$$

That is, for every propagated time $\tau \in [t,t+T]$, if $p(\tau,\cdot,\cdot)$ is unsafe, then $\boldsymbol{R}(\tau,\cdot,\cdot)$ is the root $\boldsymbol{R}_1 = \eta$ of $\kappa(\eta,p(\eta,\cdot,\cdot))$ immediately preceding τ, or if $p(\tau,\cdot,\cdot)$ is safe, then let $\boldsymbol{R}(\tau,\cdot,\cdot) = \tau$. The second case of (7.9) is defined as above because I will principally consider $\boldsymbol{R}(\tau,\cdot,\cdot)$ when the argument τ is a local maximizer of $\kappa(\tau,p(\tau,\cdot,\cdot))$. If $\tau \in \boldsymbol{M}(\cdot)$, then it follows that at points where (7.9) switches cases, i.e. where $\kappa(\tau,p(\tau,\cdot,\cdot)) = 0$ (e.g. τ_4 in Fig. 7.1), both cases of (7.9) are equivalent. That is, when the maximizer $\tau \in \boldsymbol{M}(\cdot)$ of the propagated trajectory takes on value zero, then the maximizer and root are equivalent. Thus, when the value at the maximizer τ is less than zero, choosing $\boldsymbol{R} = \tau$ yields a continuous definition of $\boldsymbol{R}$. Possible trajectories of $\kappa(\tau,p(\tau,\cdot,\cdot))$ and associated $\boldsymbol{M}$ and $\boldsymbol{R}$ values are shown in Fig. 7.2. Together, the local maximums

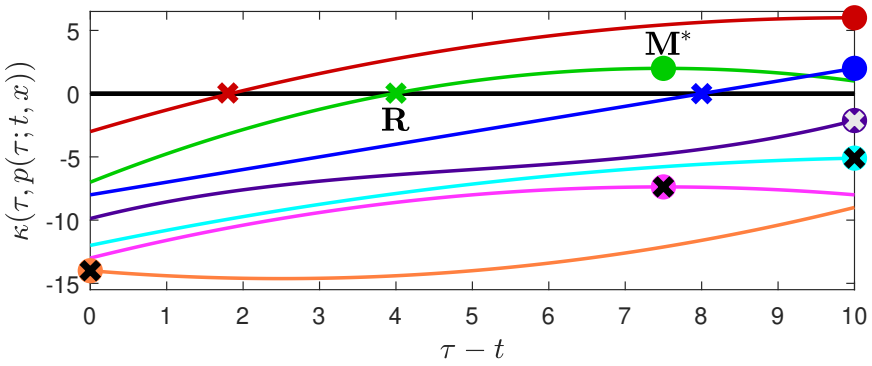

Figure 7.2: Illustration of Cases for Theorem Proof. Various possible trajectories of $\kappa(\tau, p(\tau, t, x))$ along a path p on a finite horizon $T = 10$. Each trajectory shown has only a single maximizer on $[t, t + T]$, so $\boldsymbol{M}^* = \boldsymbol{M} = \mathscr{M}$. Each maximizer $\boldsymbol{M}^*$ is annotated with a circle and each root $\boldsymbol{R}$ is annotated with an "x". When the maximum value of κ is nonpositive (bottom four lines), I define $\boldsymbol{R}(\boldsymbol{M}^*; \cdot) = \boldsymbol{M}^*$.

$\kappa(\tau, p(\tau, \cdot, \cdot)), \tau \in \boldsymbol{M}(\cdot)$ measure how unsafe a predicted trajectory is, and the roots $\boldsymbol{R}(\tau, \cdot, \cdot)$ measure how much time the controller has to make a correction.

7.1.3.3 Construction of the Path Predictive Control Barrier Function

Next, define the predictive safety function $\kappa_p : \mathcal{T} \times \mathcal{T} \times \mathcal{X} \to \mathbb{R}$ as

$$\kappa_p(\tau, t, x) \triangleq \kappa(\tau, p(\tau, t, x)) - m(\boldsymbol{R}(\tau, t, x) - t) \tag{7.11}$$

where $m : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is any continuously differentiable function that is nondecreasing and satisfies $m(0) = 0$. I also impose the following assumption.

Assumption 7.1. *Let* $\kappa_{\max} \triangleq \sup_{x \in \mathcal{S}(t), t \in \mathcal{T}} \sup_{\tau \in [t, t+T]} \kappa(\tau, p(\tau, t, x))$ *denote the maximum possible value of* κ *that may occur along propagated paths (e.g. the value of* κ *at the center of an obstacle). Assume that* $\kappa_{\max} < \infty$ *and* $m(T) \geq \kappa_{\max}$.

The first term of the function κ_p in (7.11) encodes the future safety of the system along p, while the second term encodes a relaxation (i.e. nonpositive number) based on the amount of time in the future $\boldsymbol{R} - t$ until the path p becomes unsafe. Here, m is a tunable margin function that converts this "time margin" to a "position margin" in the units of κ so as to allow for the combined metric κ_p to be nonpositive even if the predicted path is unsafe, i.e. if $\kappa(\tau, p(\tau, t, x)) > 0$. A steeper margin function m will allow κ_p to remain nonpositive when the agent is closer to obstacles (i.e. has a smaller $\boldsymbol{R} - t$) than would occur for a more gradual margin. In this context, Assumption 7.1 implies that even if a new maximizer $\tau = t + T$ is discovered at the end of the horizon, $\kappa_p(\tau, t, x)$ will initially be nonpositive. The proposed strategy is then to render $\kappa_p(\tau, t, x)$ always nonpositive,

which I will show 1) renders trajectories always inside a subset of $\mathcal{S}$ (Theorem 7.2) and 2) is feasible from a particular CBF set $\mathcal{H}$ under certain assumptions (Theorem 7.5).

Remark 7.3. *In the formulation* (7.11)*, I choose the argument of* m *to be the time until the nominal trajectory becomes unsafe (or the time until the maximizer* τ *if* $p(\eta, t, x)$ *is safe for all* $\eta \in [t, \tau]$*). Alternatively, [264, Eq. 38] chooses the argument of* m *to be the current safety distance* $\kappa(t, x(t))$*. I expect the time-based formulation in* (7.11) *to be more appropriate than [264, Eq. 22] in cases where* $\kappa(\tau, p(\tau, \cdot, \cdot))$ *is expected to vary rapidly with* τ *(e.g. as occurs for the low-Earth-orbiting system in Section 7.1.4.2). As* $m(\boldsymbol{R} - t)$ *serves as a relaxation on the first term of* (7.11)*, I suspect that there exist other useful relaxations, though using such a relaxation would require re-proving Theorem 7.2.*

I then apply (7.11) at the local maximizer times in $\boldsymbol{M}$ in (7.7) to define functions $h_{p,\text{all}} : \mathcal{T} \times \mathcal{X} \to 2^{\mathbb{R}}$ and $h_p : \mathcal{T} \times \mathcal{X} \to \mathbb{R}$ as

$$h_{p,\text{all}}(t, x) \triangleq \begin{bmatrix} \vdots \\ \kappa_p(\tau_i, t, x) \\ \vdots \end{bmatrix}, \forall \tau_i \in \boldsymbol{M}(t, x), \tag{7.12}$$

$$h_p(t, x) \triangleq \kappa_p(\boldsymbol{M}^*(t, x), t, x)\,. \tag{7.13}$$

I now present theorems relating the future safety encoded in $h_{p,\text{all}}$ and h_p to the present safety given by κ in (7.2).

Theorem 7.2. *For some constraint function* $\kappa : \mathcal{T} \times \mathcal{X} \to \mathbb{R}$*, let* $\mathcal{S}$ *be as in* (7.2) *and* h_p *as in* (7.13)*, with* $\mathcal{H}_p$ *defined as*

$$\mathcal{H}_p(t) \triangleq \{x \in \mathcal{X} \mid h_p(t, x) \leq 0\}\,. \tag{7.14}$$

Then $\mathcal{H}_p(t) \subseteq \mathcal{S}(t), \forall t \in \mathcal{T}$*.*

Proof. By construction, $\boldsymbol{M}$ in (7.7) is always nonempty, because if no local maxima occur for $\tau \in [t, t + T)$, then $\tau = t + T$ is guaranteed to be an element in (7.6). Thus, $\boldsymbol{M}^*$ always exists and h_p is well defined.

First, consider if $\boldsymbol{M}^*(t, x) = t$ (i.e. κ is initially decreasing along the path p). Then $h_p(t, x) = \kappa(t, x)$ and the result follows immediately.

Second, consider if $\boldsymbol{M}^*(t, x) \neq t$. This implies that κ is initially increasing along the path (e.g., as occurs in the top six lines of Fig. 7.2), so it must be that $\kappa(t, x) \leq \kappa(\tau, p(\tau, t, x))$ for all times $\tau \in [t, \boldsymbol{M}^*(t, x)]$, so to complete the proof, I only need to show that there exists such a time $\tau \in [t, \boldsymbol{M}^*(t, x)]$ where $\kappa(\tau, p(\tau, t, x)) \leq 0$. If $\kappa(\boldsymbol{M}^*(t, x), p(\boldsymbol{M}^*(t, x); t, x)) \leq 0$ (bottom four lines of Fig. 7.2), then the proof is done. If instead $\kappa(\boldsymbol{M}^*(t, x), p(\boldsymbol{M}^*(t, x); t, x)) > 0$ (top

three lines of Fig. 7.2), then by (7.11), $x \in \mathcal{H}_p(t)$ implies that $m(\boldsymbol{R}(\boldsymbol{M}^*(t,x);t,x) - t) > 0$. Since $m(0) = 0$ and m is nondecreasing, it follows that $\boldsymbol{R}(\boldsymbol{M}^*(t,x);t,x) > t$. Thus, there exists $\tau = \boldsymbol{R}(\boldsymbol{M}^*(t,x);t,x) \in (t, \boldsymbol{M}^*(t,x))$ such that $\kappa(\tau, p(\tau,t,x)) = 0$, so it must be that $\kappa(t,x) \leq \kappa(\tau, p(\tau,t,x)) \leq 0$. Thus, $x \in \mathcal{H}_p(t)$ implies $x \in \mathcal{S}(t)$. ■

Corollary 7.3. *Let $q(t,x) = |\boldsymbol{M}(t,x)|$ and let $h_{p,i}$ be the ith output of $h_{p,\text{all}}$ in* (7.12)*. Define the set $\mathcal{H}_{p,\text{all}}$ as*

$$\mathcal{H}_{p,\text{all}}(t) \triangleq \{x \in \mathcal{X} \mid h_{p,i}(t,x) \leq 0, \forall i = 1, \cdots, q(t,x)\}\,. \tag{7.15}$$

Then $\mathcal{H}_{p,\text{all}}(t) \subseteq \mathcal{S}(t), \forall t \in \mathcal{T}$.

Proof. The result follows immediately from Theorem 7.2 since $h_{p,1} \equiv h_p$. ■

That is, rendering either $\mathcal{H}_{p,\text{all}}$ or $\mathcal{H}_p$ forward invariant is sufficient to render trajectories always inside the safe set $\mathcal{S}$. The remaining question is under what conditions rendering $\mathcal{H}_{p,\text{all}}$ or $\mathcal{H}_p$ forward invariant will be possible. First, I make some regularity assumptions.

Assumption 7.2. *Assume that the functions $h_{p,\text{all}}$ and h_p in* (7.12) *and* (7.13)*, respectively, are absolutely continuous. Assume further that each maximizer location $\tau_i \in \boldsymbol{M}(t,x) \cap (t, t+T)$, where $\tau_i = \tau_i(t,x)$, is continuously differentiable.*

That is, each maximizer τ_i in $\boldsymbol{M}(t,x)$ is also a function of t and x, and I assume that the variation of $\tau_i(t,x)$ is regular when τ_i is in the open set $(t, t+T)$. When τ_i is at the start or end of the horizon, this will in general not be true, which is why I only assume $h_{p,\text{all}}$ and h_p to be absolutely continuous rather than continuously differentiable (and why I introduced Definition 7.1 and Lemma 7.1).

Next, I present a result on how to compute the derivatives of h_p in (7.13) and each output $h_{p,i}$ in (7.12). Going forward, to remove ambiguity, let $\frac{\partial p(a,b,c)}{\partial \lambda_\tau}$, $\frac{\partial p(a,b,c)}{\partial \lambda_t}$, and $\frac{\partial p(a,b,c)}{\partial \lambda_x}$ denote the partial derivative of p with respect to its first, second, and third argument, respectively, and evaluated at (a,b,c), similar to the notation used in Section 4.3.3. Similarly, let $\frac{\partial \kappa(a,b)}{\partial \lambda_t}$ and $\frac{\partial \kappa(a,b)}{\partial \lambda_x}$ denote the partial derivative with respect to the first and second argument of κ, respectively, evaluated at (a,b), i.e. $\frac{\partial \kappa(a,b)}{\partial \lambda_t} = \left.\frac{\partial \kappa(t,x)}{\partial t}\right|_{t=a,x=b}$ and $\frac{\partial \kappa(a,b)}{\partial \lambda_x} = \left.\frac{\partial \kappa(t,x)}{\partial x}\right|_{t=a,x=b}$.

Lemma 7.4. *Let $\tau \in \boldsymbol{M}(t,x)$ and let $m' : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ be the derivative of m in* (7.11)*. Assume $x \in \mathcal{H}_p(t)$.*

Case i: *If $\tau \in (t, t+T)$, then*

$$\frac{d}{dt}[\kappa_p(\tau, p(\tau,t,x))] = m'(\boldsymbol{R}(\tau,t,x) - t) + B(\tau,t,x)(u - \mu(t,x))\,. \tag{7.16}$$

Case ii: *If* $\tau = t + T$ *and* $\boldsymbol{R}(\tau, t, x) < \tau$*, then*

$$\frac{d}{dt}[\kappa_p(\tau, p(\tau, t, x))] = m'(\boldsymbol{R}(\tau, t, x) - t) + B(\tau, t, x)(u - \mu(t, x)) + \frac{d}{d\tau}[\kappa(\tau, p(\tau, t, x))]\frac{d\tau(t, x)}{dt}\,. \quad (7.17)$$

Case iii: *If* $\tau \in \{t, t + T\}$ *and* $\boldsymbol{R}(\tau, t, x) = \tau$*, then*

$$\frac{d}{dt}[\kappa_p(\tau, p(\tau, t, x))] = \frac{\partial\kappa(\tau, p(\tau, t, x))}{\partial\lambda_x}\frac{\partial p(\tau, t, x)}{\partial\lambda_x}g(t, x)(u - \mu(t, x)) + \frac{d}{d\tau}[\kappa(\tau, p(\tau, t, x))]\frac{d\tau(t, x)}{dt} - m'(\boldsymbol{R}(\tau, t, x) - t)\left(\frac{d\tau(t, x)}{dt} - 1\right)\,, \quad (7.18)$$

where

$$B(\tau, t, x) = \left[\frac{\partial\kappa(\tau, p(\tau, t, x))}{\partial\lambda_x}\frac{\partial p(\tau, t, x)}{\partial\lambda_x} - m'(\boldsymbol{R}(\tau, t, x) - t)C(\tau, t, x)\right]g(t, x)\,, \quad (7.19)$$

$$C(\tau, t, x) = \begin{cases} \frac{\partial\tau(t,x)}{\partial\lambda_x} & \boldsymbol{R}(\tau, t, x) = \tau \\ C_1(\boldsymbol{R}_1(\tau, t, x), t, x) & \boldsymbol{R}(\tau, t, x) \neq \tau \end{cases}, \quad (7.20)$$

$$C_1(\eta, t, x) = -\frac{\partial\kappa(\eta, p(\eta, t, x))}{\partial\lambda_x}\frac{\partial p(\eta, t, x)}{\partial\lambda_x} \cdot \left[\frac{\partial\kappa(\eta, p(\eta, t, x))}{\partial\lambda_t} + \frac{\partial\kappa(\eta, p(\eta, t, x))}{\partial\lambda_x}\frac{\partial p(\eta, t, x)}{\partial\lambda_\tau}\right]^{-1} \quad (7.21)$$

and where $\frac{d\tau(t,x)}{dt} = \partial_t\tau(t, x) + L_{f(t,x)+g(t,x)u}\tau(t, x)$ *describes the sensitivity of the maximizer time to the current time* $t \in \mathcal{T}$ *and state* $x \in \mathcal{X}$*.*

Proof. First, note that every τ in $\boldsymbol{M}(t, x)$ is a function $\tau(t, x)$ defined as the maximizer of κ along the path originating from (t, x) and restricted to a sufficiently small domain around τ (see Fig. 4.3 for a related visualization).

I start by deriving a relationship between the partial derivatives of p in (7.5). For a fixed time $\sigma > t$, by definition $p(\sigma, t, x)$ is constant so long as $\dot{x} = f(t, x) + g(t, x)\mu(t, x)$ as in (7.5). Thus, it follows that

$$c = p(\sigma, t, x) \text{ if } \dot{x} = f(t, x) + g(t, x)\mu(t, x) \quad (7.22)$$

for some constant $c \in \mathbb{R}^n$. Thus,

$$0 = \frac{d}{dt}\Big[p(\sigma, t, x)\Big] \text{ if } \dot{x} = f(t,x) + g(t,x)\mu(t,x) \tag{7.23a}$$

$$= \frac{\partial p(\sigma,t,x)}{\partial \lambda_\tau} \underbrace{\frac{d\sigma}{dt}}_{=0} + \frac{\partial p(\sigma,t,x)}{\partial t} + \frac{\partial p(\sigma,t,x)}{\partial x}\left[f(t,x) + g(t,x)\mu(t,x)\right] \tag{7.23b}$$

$$\implies \frac{\partial p(\sigma,t,x)}{\partial \lambda_t} = -\frac{\partial p(\sigma,t,x)}{\partial \lambda_x}\left[f(t,x) + g(t,x)\mu(t,x)\right] \tag{7.23c}$$

That is, (7.23c) provides a simplification for the partial derivative of p in (7.5) with respect to its second argument, the initial time. This result is not strictly necessary to apply the CBF (7.11) (i.e. one could simply differentiate (7.11) directly), but this result is convenient because it reveals the structure of $\frac{dh_p}{dt}$ as a difference between u and μ.

Next, I compute the total derivative of the first term of (7.11), $\kappa(\tau, t, x)$, as follows

$$\frac{d\kappa(\tau, p(\tau,t,x))}{dt} = \frac{\partial \kappa(\tau, p(\tau,t,x))}{\partial \lambda_t}\frac{d\tau}{dt} + \frac{\partial \kappa(\tau, p(\tau,t,x))}{\partial \lambda_x}\left[\frac{\partial p(\tau,t,x)}{\partial \lambda_\tau}\frac{d\tau}{dt} + \frac{\partial p(\tau,t,x)}{\partial \lambda_t} + \frac{\partial p(\tau,t,x)}{\partial \lambda_x}\frac{dx}{dt}\right] \tag{7.24a}$$

$$\overset{(7.23c)}{=} \left[\frac{\partial \kappa(\tau, p(\tau,t,x))}{\partial \lambda_t} + \frac{\partial \kappa(\tau, p(\tau,t,x))}{\partial \lambda_x}\frac{\partial p(\tau,t,x)}{\partial \lambda_\tau}\right]\frac{d\tau}{dt} + \frac{\partial \kappa(\tau, p(\tau,t,x))}{\partial \lambda_x}\frac{\partial p(\tau,t,x)}{\partial \lambda_x}\left[\frac{dx}{dt} - \Big(f(t,x) + g(t,x)\mu(t,x)\Big)\right] \tag{7.24b}$$

$$\overset{(7.1),(7.5)}{=} \frac{d}{d\tau}[\kappa(\tau, p(\tau,t,x))]\frac{d\tau}{dt} + \frac{\partial \kappa(\tau, p(\tau,t,x))}{\partial \lambda_x}\frac{\partial p(\tau,t,x)}{\partial \lambda_x} g(t,x)(u - \mu(t,x)). \tag{7.24c}$$

Note that (7.24c) holds in all three cases in Lemma 7.4.

Next, consider the second term of (7.11), which has total derivative

$$\frac{dm(\boldsymbol{R}(\tau,t,x) - t)}{dt} = m'(\boldsymbol{R}(\tau,t,x) - t)\left(\frac{d\boldsymbol{R}(\tau,t,x)}{dt} - 1\right), \tag{7.25}$$

which further requires one to compute $\frac{d}{dt}[\boldsymbol{R}(\tau,t,x)]$. Note that $x \in \mathcal{H}_p(t)$ implies that $\boldsymbol{R}(\tau,t,x)$ is well defined for all $\tau \geq t$ (i.e. $\kappa(t,x) \leq 0$, so (7.9) always returns a value).

In the case where $\boldsymbol{R}(\tau,t,x) = \tau$, then $\frac{d}{dt}[\boldsymbol{R}(\tau,t,x)] = \frac{d\tau(t,x)}{dt}$. If instead $\boldsymbol{R}(\tau,t,x) \neq \tau$, then by (7.9), $\eta = \boldsymbol{R}_1(\tau,t,x)$ is a zero of $\kappa(\eta, p(\eta,t,x))$ over η in a neighborhood preceding τ. That is,

$$0 = \kappa(\boldsymbol{R}_1(\tau,t,x), p(\boldsymbol{R}_1(\tau,t,x); t, x)), \tag{7.26}$$

which when differentiated with respect to t yields

$$0 = \frac{\partial \kappa(\boldsymbol{R}_1, p(\boldsymbol{R}_1,t,x))}{\partial \lambda_t}\frac{d\boldsymbol{R}_1}{dt} + \frac{\partial \kappa(\boldsymbol{R}_1, p(\boldsymbol{R}_1,t,x))}{\partial \lambda_x}\left[\frac{\partial p(\boldsymbol{R}_1,t,x)}{\partial \lambda_\tau}\frac{d\boldsymbol{R}_1}{dt} + \frac{\partial p(\boldsymbol{R}_1,t,x)}{\partial \lambda_t} + \frac{\partial p(\boldsymbol{R}_1,t,x)}{\partial \lambda_x}\frac{dx}{dt}\right] \tag{7.27a}$$

$$\implies \frac{d\boldsymbol{R}_1(\tau,t,x)}{dt} \overset{(7.27a)}{=} -\left(\frac{\partial \kappa(\boldsymbol{R}_1, p(\boldsymbol{R}_1,t,x))}{\partial \lambda_t} + \frac{\partial \kappa(\boldsymbol{R}_1, p(\boldsymbol{R}_1,t,x))}{\partial \lambda_x}\frac{\partial p(\boldsymbol{R}_1,t,x)}{\partial \lambda_\tau}\right)^{-1} \cdot \frac{\partial \kappa(\boldsymbol{R}_1, p(\boldsymbol{R}_1,t,x))}{\partial \lambda_x}\left[\frac{\partial p(\boldsymbol{R}_1,t,x)}{\partial \lambda_t} + \frac{\partial p(\boldsymbol{R}_1,t,x)}{\partial \lambda_x}\frac{dx}{dt}\right] \tag{7.27b}$$

$$\overset{(7.23c)}{=} -\left(\frac{\partial \kappa(\boldsymbol{R}_1, p(\boldsymbol{R}_1,t,x))}{\partial \lambda_t} + \frac{\partial \kappa(\boldsymbol{R}_1, p(\boldsymbol{R}_1,t,x))}{\partial \lambda_x}\frac{\partial p(\boldsymbol{R}_1,t,x)}{\partial \lambda_\tau}\right)^{-1} \cdot \frac{\partial \kappa(\boldsymbol{R}_1, p(\boldsymbol{R}_1,t,x))}{\partial \lambda_x}\frac{\partial p(\boldsymbol{R}_1,t,x)}{\partial \lambda_x}\left[\frac{dx}{dt} - \Big(f(t,x) + g(t,x)\mu(t,x)\Big)\right] \tag{7.27c}$$

$$\overset{(7.1),(7.5)}{=} -\left(\frac{\partial \kappa(\boldsymbol{R}_1, p(\boldsymbol{R}_1,t,x))}{\partial \lambda_t} + \frac{\partial \kappa(\boldsymbol{R}_1, p(\boldsymbol{R}_1,t,x))}{\partial \lambda_x}\frac{\partial p(\boldsymbol{R}_1,t,x)}{\partial \lambda_\tau}\right)^{-1} \cdot \frac{\partial \kappa(\boldsymbol{R}_1, p(\boldsymbol{R}_1,t,x))}{\partial x}\frac{\partial p(\boldsymbol{R}_1,t,x)}{\partial x} g(t,x)(u-\mu(t,x)) \tag{7.27d}$$

where (7.27d) is used to define $C_1(\boldsymbol{R}_1,t,x)$ in (7.21).

The total derivative $\frac{d}{dt}[\kappa(\tau(t,x), p(\tau(t,x);t,x))]$ in (7.16)-(7.18) is then the difference between (7.24c) and (7.25) where I make the following simplifications.

Case i: In this case, τ is a maximizer of $\kappa(\tau, p(\tau,t,x))$ on an open interval, and κ and p are assumed to be continuously differentiable, so a necessary condition for $\tau \in \boldsymbol{M}(t,x)$ is that $\frac{d}{d\tau}[\kappa(\tau, p(\tau,t,x))] = 0$. Moreover, since $\tau(t,x)$ is assumed to be continuously differentiable, the term $\frac{d\tau}{dt}$ in the first line of (7.24c) is finite. Therefore, the first line of (7.24c) is zero in Case i and does not appear in (7.16). The remaining second line of (7.24c) is incorporated into the first term of (7.19).

Next, since $\tau \in (t, t+T)$, by Assumption 7.2, the maximizer $\tau(t,x)$ is continuously differentiable. Similar to (7.22) and (4.44), the maximizer time will also not change if the trajectory continues along the nominal path. That is, for σ in a sufficiently small neighborhood of t, there exists $c \in \mathbb{R}$ such that

$$c = \tau(t,x) = \tau(\sigma, p(\sigma,t,x))\,, \tag{7.28}$$

which implies

$$0 = \frac{d}{d\sigma}\Big[\tau(\sigma, p(\sigma,t,x))\Big] = \frac{\partial \tau(\sigma, p(\sigma,t,x))}{\partial \lambda_t} + \frac{\partial \tau(\sigma, p(\sigma,t,x))}{\partial \lambda_x}\frac{\partial p(\sigma,t,x)}{\partial \lambda_\tau} \tag{7.29a}$$

$$\implies \frac{\partial \tau(t, p(\sigma, t, x))}{\partial \lambda_t} = -\frac{\partial \tau(\sigma, p(\sigma, t, x))}{\partial \lambda_x} \frac{\partial p(\sigma, t, x)}{\partial \lambda_\tau} . \tag{7.29b}$$

I then select $\sigma = t$ to get

$$\begin{aligned} \frac{\partial \tau(t, p(t, t, x))}{\partial \lambda_t} &= -\frac{\partial \tau(t, p(t, t, x))}{\partial \lambda_x} \frac{\partial p(t, t, x)}{\partial \lambda_\tau} \\ \frac{\partial \tau(t, x)}{\partial \lambda_t} &= -\frac{\partial \tau(t, x)}{\partial \lambda_x} \Big(f(t, x) + g(t, x)\mu(t, x) \Big) . \end{aligned} \tag{7.29c}$$

Thus, the variation of the maximizer time $\tau(t, x) \in \boldsymbol{M}(t, x)$ is

$$\frac{d\tau(t, x)}{dt} = \frac{\partial \tau(t, x)}{\partial \lambda_t} + \frac{\partial \tau(t, x)}{\partial \lambda_x} \frac{dx}{dt} \tag{7.30a}$$

$$\overset{(7.29c)}{=} \frac{\partial \tau(t, x)}{\partial \lambda_x} \left[\frac{dx}{dt} - \Big(f(t, x) + g(t, x)\mu(t, x) \Big) \right] \tag{7.30b}$$

$$\overset{(7.1),(7.5)}{=} \frac{\partial \tau(t, x)}{\partial \lambda_x} g(t, x)(u - \mu(t, x)) . \tag{7.30c}$$

Note that (7.30c) is only valid if $\tau(t, x)$ is continuously differentiable, which is why Cases ii-iii (when τ is an endpoint) simply include $\frac{d\tau(t,x)}{dt}$ rather than using the simplification (7.30c).

When $\boldsymbol{R}(\tau, t, x) = \tau$, the simplification (7.30c) yields the first case of $C(\tau, t, x)$ in (7.20). When instead $\boldsymbol{R}(\tau, t, x) \neq \tau$, the simplification (7.27d) yields the second case of $C(\tau, t, x)$ in (7.20). Both cases are then incorporated into the second term of (7.19). Thus, (7.16) accounts for all the terms of the difference between (7.24c) and (7.25).

Case ii: In this case, $\frac{d}{d\tau}[\kappa(\tau, p(\tau, t, x))]$ is no longer guaranteed to be zero, so I add the first line of (7.24c) back to (7.17) compared to (7.16). Since τ is an endpoint of the horizon, τ is no longer assumed continuously differentiable so $\frac{\partial \tau(t,x)}{\partial \lambda_x}$ may no longer be well-defined. Thus, the simplification in (7.30c) may no longer hold, so I do not simplify $\frac{d\tau(t,x)}{dt}$ from (7.24c) any further.

Next, looking at the second term of (7.19), since Case ii assumes that $\boldsymbol{R}(\tau, t, x) < \tau$, the second case of (7.20) (i.e. $\boldsymbol{R} = \tau$, also the first case of (7.9)) will always hold in Case ii. Thus, even though C in (7.20) includes the term $\frac{\partial \tau(t,x)}{\partial \lambda_x}$ (which by the above logic is no longer well-defined), I can use the same formula for B and C in (7.19) and (7.20) as in Case i, because the case involving $\frac{\partial \tau(t,x)}{\partial x}$ is excluded. Thus, (7.17) accounts for all the terms of the difference between (7.24c) and (7.25).

Case iii: This case is identical to Case ii, except that $\boldsymbol{R}(\tau, t, x) = \tau$ so $\frac{d}{dt}[\boldsymbol{R}(\tau, t, x)]$ is equivalent to $\frac{d\tau(t,x)}{dt}$ instead of coming from (7.27d) via $C(\tau, t, x)$ in (7.20). Thus, the first two terms of (7.18) account for all the terms of (7.24c), while the last term of (7.18) accounts for all the terms of (7.25). ■

Remark 7.4. *In* (7.18)*, if $\tau = t$, then $\frac{d\tau(t,x)}{dt}$ is either equal to 1 or is undefined. That is, the maximizer time advances at the same rate as the time variable, or else advances in a discontinuous*

fashion.

In (7.17) *and* (7.18)*, if* $\tau = t + T$ *and if* $\frac{d\tau(t,x)}{dt}$ *is well-defined, then either* $\frac{d\tau(t,x)}{dt} = 1$ *or* $\frac{d}{d\tau}[\kappa(\tau, p(\tau, t, x))] = 0$*. The case* $\frac{d}{d\tau}[\kappa(\tau, p(\tau, t, x))] = 0$ *implies that on a slightly larger interval* $[t, t + T + \epsilon)$*, the time* $t + T$ *is a local maximizer. The case* $\frac{d\tau(t,x)}{dt} = 1$ *implies that the maximizer time advances at the same rate as the horizon progresses forward.*

Note that the case $\boldsymbol{R} = \tau$ *in* (7.20) *only gets invoked by Case i of Lemma 7.4, where* $\tau \in (t+T)$*. In this case,* $\frac{d\tau(t,x)}{dt}$ *could take on any value (i.e. not just 0 and 1). It follows that* $\frac{\partial \tau(t,x)}{\partial \lambda_x}$ *in* (7.20) *could take on any value and thus must be numerically computed. Lemma 7.4 assumes that* $\tau \in (t, t + T)$ *are continuously differentiable so that this value is well-defined, though proving this is difficult and is an area for future work.*

I am now ready to prove that, under mild assumptions elaborated upon after the proof, h_p in (7.13) is a CBF, and thus can be used to ensure safety via the CBF condition.

Theorem 7.5 (Main Theorem: h_p is a CBF)**.** *Let the derivative* $m' : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ *of* m *in* (7.11) *be strictly positive on* $(0, T)$*. Suppose that there exists* γ *such that* $\frac{d}{d\tau}[\kappa(\tau, p(\tau, t, x))] \leq \gamma$ *for all* $\tau \in \mathcal{T}, t \in \mathcal{T}, x \in \mathcal{X}$*. Suppose also that* $\| \frac{\partial \kappa(\eta, p(\eta,t,x))}{\partial \lambda_x} \frac{\partial p(\eta,t,x)}{\partial \lambda_x} g(t, x) \| \neq 0$ *for all* $\eta \in (t, t+T) \backslash \mathscr{M}(t, x)$ *and for all* $t \in \mathcal{T}, x \in \mathcal{X}$*, and that* $\frac{\partial \kappa(\tau, p(\tau,t,x))}{\partial \lambda_x}^{\mathrm{T}} \frac{\partial \kappa(\eta, p(\eta,t,x))}{\partial \lambda_x} \geq 0$ *whenever* $\eta = \boldsymbol{R}(\tau, t, x)$ *for all* $\tau \in \mathcal{T}, t \in \mathcal{T}, x \in \mathcal{X}$*. Then* h_p *in* (7.13) *is a CBF as in Definition 7.1.*

Proof. By Assumption 7.2, h_p is absolutely continuous, and therefore differentiable almost everywhere. Thus, h_p is a CBF as in Definition 7.1 if there exists $\alpha \in \mathcal{K}_e$ satisfying condition (7.3) almost everywhere. I will show that such an $\alpha \in \mathcal{K}_e$ always exists, by breaking this proof into cases where A) $\boldsymbol{R}(\boldsymbol{M}^*(t, x); t, x) = \boldsymbol{M}^*(t, x)$ and B) $\boldsymbol{R}(\boldsymbol{M}^*(t, x); t, x) \neq \boldsymbol{M}^*(t, x)$. I further break both A and B into the cases 1) $\boldsymbol{M}^*(t, x) = t$, 2) $\boldsymbol{M}^*(t, x) \in (t, t + T)$, and 3) $\boldsymbol{M}^*(t, x) = t + T$. Note that while Lemma 7.4 applies to any $\tau \in \boldsymbol{M}(t, x)$, this theorem only considers $\tau = \boldsymbol{M}^*(t, x)$, so $\frac{d\tau(t,x)}{dt} \equiv \frac{d\boldsymbol{M}^*(t,x)}{dt}$ in this proof.

In cases A1-A3, the path $p(\tau, t, x)$ is safe for future times τ until at least $\boldsymbol{M}^*(t, x)$, so intuitively choosing $u = \mu(t, x)$ should render the system safe, thereby satisfying (7.3). I show this formally as follows. First, case A1 (orange line in Fig. 7.2) implies that $\tau = \boldsymbol{M}^*(t, x) = t$ is a local maximizer of $\kappa(\tau, p(\tau, t, x))$ on $\tau \in [t, t + T]$. This implies that κ is initially nonincreasing along the path p beginning from (t, x), so $\frac{d}{d\tau}\left[\kappa(\tau, p(\tau, t, x)]\right|_{\tau=t} \leq 0$. Moreover, if the agent continues along the path (i.e. chooses $u = \mu(t, x)$), then at an arbitrarily small time $t + \epsilon$ in the future, $\boldsymbol{M}^*(t + \epsilon, p(t + \epsilon; t, x)) = t + \epsilon$ will be the first local maximizer on the horizon $[t + \epsilon, t + \epsilon + T]$ (see also $\mathcal{B}_{loc}(0)$ in Fig. 4.3), so $\frac{d\boldsymbol{M}^*(t,x)}{dt} = 1$ under $u = \mu(t, x)$. Thus, (7.18) implies that $\frac{d}{dt}[h_p(t, x)] = \frac{d}{d\tau}\left[\kappa(\tau, p(\tau, t, x)]\right|_{\tau=t} \leq 0$ under $u = \mu(t, x)$, which satisfies condition (7.3) for any $\alpha \in \mathcal{K}_e$.

Next, in case A2 (magenta line in Fig. 7.2), (7.16) implies that under $u = \mu(t,x)$, it follows that $\frac{d}{dt}[h_p(t,x)] = m'(\boldsymbol{M}^*(t,x) - t)$. Moreover, case A assumes that $\boldsymbol{R}(\boldsymbol{M}^*(t,x);t,x) = \boldsymbol{M}^*(t,x)$, which by (7.9) implies that the predicted maximum value $\kappa(\boldsymbol{M}^*(t,x), p(\boldsymbol{M}^*(t,x);t,x))$ is at most zero, so $h_p(t,x) \leq -m(\boldsymbol{M}^*(t,x) - t) < 0$. Thus, choose $\alpha \in \mathcal{K}_e$ such that $\alpha(m(\lambda)) \geq m'(\lambda), \forall \lambda \in [0,T]$, and then (7.3) is satisfied under $u = \mu(t,x)$. Note that if $\kappa(\boldsymbol{M}^*(t,x), p(\boldsymbol{M}^*(t,x);t,x)) = 0$, then $u = \mu(t,x)$ will cause h_p to reach (but not exceed since $\boldsymbol{M}^*$ is a maximizer) the origin in finite time. Thus, a function $\alpha \in \mathcal{K}_e$ satisfying the above condition will not be Lipschitz continuous (recall that a Lipschitz α will not allow finite-time convergence to the origin).

Next, in case A3, two behaviors are possible at the end of the horizon. First, if κ is nonincreasing along the path at the end of the horizon (cyan line in Fig. 7.2), i.e. $\frac{d}{d\tau}\left[\kappa(\tau, p(\tau,t,x))\right]|_{\tau=t+T} = 0$, then under $u = \mu(t,x)$, at some arbitrarily small time $t+\epsilon$ in the future, $\boldsymbol{M}^*(t+\epsilon, p(t+\epsilon;t,x)) = \boldsymbol{M}^*(t,x)$ will still be a local maximizer. Thus, $\frac{d\boldsymbol{M}^*(t,x)}{dt} = 0$ under $u = \mu(t,x)$, so (7.18) implies that $\frac{d}{dt}[h_p(t,x)] = m'(\boldsymbol{M}^*(t,x) - t)$, and the same logic as in case A2 implies satisfaction of condition (7.3). Second, if instead $\frac{d}{d\tau}\left[\kappa(\tau, p(\tau,t,x))\right]|_{\tau=t+T} > 0$ (purple line in Fig. 7.2), then under $u = \mu(t,x)$, at some arbitrarily small time $t+\epsilon$ in the future, $\boldsymbol{M}^*(t+\epsilon, p(t+\epsilon;t,x)) = \boldsymbol{M}^*(t,x) + \epsilon$ will be the local maximizer on the horizon $[t+\epsilon, t+\epsilon+T]$. Thus, $\frac{d\boldsymbol{M}^*(t,x)}{dt} = 1$ under $u = \mu(t,x)$, so (7.18) implies that $\frac{d}{dt}[h_p(t,x)] = \frac{d}{d\tau}\left[\kappa(\tau, p(\tau,t,x))\right]|_{\tau=t+T} \leq \gamma$. Similar to case A2, $h_p(t,x) \leq -m(\boldsymbol{M}^*(t,x) - t) = -m(T) < 0$, so choose α so that $\alpha(m(T)) \geq \gamma$, and then condition (7.3) is satisfied for $u = \mu(T,x)$.

Next, note that case B1 is not possible, since (7.9) implies that $\boldsymbol{R}(t;t,x) = t$, which would imply case A1. Next, cases B2-B3 imply that $\kappa(\boldsymbol{M}^*(t,x), p(\boldsymbol{M}^*(t,x);t,x)) > 0$; i.e the path $p(\tau,t,x)$ becomes unsafe on the horizon $[t, \boldsymbol{M}^*] \subseteq [t, t+T]$, so unlike cases A1-A3, the system will need to take a control action $u \neq \mu(t,x)$ to ensure safety. Since $\kappa(\boldsymbol{M}^*(t,x), p(\boldsymbol{M}^*(t,x);t,x)) > 0$, any $x \in \mathcal{H}_p(t)$ must satisfy $\boldsymbol{R}(\boldsymbol{M}^*(t,x);t,x) > t$, so the system has until time $\boldsymbol{R}(\boldsymbol{M}^*(t,x);t,x)$ to take corrective action. First, in case B2, both the maximizer time $\boldsymbol{M}^*$ and the root $\boldsymbol{R}$ occur within the time horizon (green line in Fig. 7.2), and $\frac{d}{dt}[h_p(t,x)]$ is given by (7.16). Since m' is assumed to be strictly positive and $\frac{\partial \kappa(\eta, p(\eta,t,x))}{\partial \lambda_x}\frac{\partial p(\eta,t,x)}{\partial \lambda_x} g(t,x)$ is assumed to not be the zero vector, the second line of (7.19) is guaranteed nonzero. Additionally, since the theorem assumed that the inner product of $\frac{\partial \kappa(\tau, p(\tau,t,x))}{\partial \lambda_x}$ and $\frac{\partial \kappa(\eta, p(\eta,t,x))}{\partial \lambda_x}$ is nonnegative for $\tau = \boldsymbol{M}^*(t,x)$ and $\eta = \boldsymbol{R}(\boldsymbol{M}^*(t,x);t,x)$, the first term of (7.19) cannot cancel with the second term, so the matrix $B(\tau,t,x)$ in (7.19) is guaranteed nonzero. Since $u \in \mathbb{R}^m$ is unconstrained, one can thus choose u to make $\frac{d}{dt}[h_p(t,x)]$ in (7.16) arbitrarily negative, so (7.3) can be satisfied for any $\alpha \in \mathcal{K}_e$.

Lastly, case B3 represents the scenario where the path becomes unsafe for some time $\boldsymbol{R}(\boldsymbol{M}^*(t,x);t,x) \in (t, t+T)$ in the prediction horizon, and the maximizer of $\kappa(\tau, p(\tau,t,x))$

occurs either at (red line in Fig. 7.2) or after (blue line in Fig. 7.2) the end of the horizon. For this case to occur, it must be that $\frac{d}{d\tau}\left[\kappa(\tau, p(\tau, t, x))\right]|_{\tau=t+T} \geq 0$. However, by assumption, $\frac{d}{d\tau}[\kappa(\tau, p(\tau, t, x))] \leq \gamma$ also. Since the prediction horizon T is constant, it follows that $\frac{d\boldsymbol{M}^*(t,x)}{dt} \leq 1$, so the second line of (7.17) is upper bounded by γ. The term $m'(\boldsymbol{R}(\tau, t, x) - t)$ is also independent of u and is bounded since m is assumed to be continuously differentiable. Therefore, by the same argument as in case B2, one can choose u to make $\frac{d}{dt}[h_p(t, x)]$ in (7.17) arbitrarily negative, so (7.3) can be satisfied for any $\alpha \in \mathcal{K}_e$.

Thus, in every case, a function $\alpha \in \mathcal{K}_e$ satisfying $\alpha(m(\lambda)) \geq m'(\lambda), \forall \lambda \in [0, T]$ and $\alpha(m(T)) \geq \gamma$ will satisfy condition (7.3), so h_p is indeed a CBF. Note that these are sufficient, not necessary conditions on α. ■

That is, under mild assumptions, it is always possible to render the set $\mathcal{H}_p$ forward invariant, so I call h_p as in (7.13) a P^{2}CBF. To elaborate on these assumptions, first, γ represents a bound on the rate of change of κ along the nominal path p, which is a reasonable assumption for most paths. Next, intuitively, the assumption that $\|\frac{\partial \kappa(\eta, p(\eta,t,x))}{\partial \lambda_x}\frac{\partial p(\eta,t,x)}{\partial \lambda_x} g(t, x)\| \neq 0$ is analogous to controllability and observability. This quantity is the sensitivity of the future value of κ to the current control input, so this assumption encodes that both the state trajectory $p(\eta, \cdot, \cdot)$ and the observed values $\kappa(\eta, p(\eta, \cdot, \cdot))$ along that trajectory must be controllable. If instead future κ values are not controllable along p, then one cannot expect such a predictive strategy using the path function p to be useful. That said, Theorem 7.5 makes an exception to allow for $\|\frac{\partial \kappa(\tau, p(\tau,t,x))}{\partial \lambda_x}\frac{\partial p(\tau,t,x)}{\partial \lambda_x} g(t, x)\| = 0$ to occur when τ is a maximizer time (i.e. $\tau \in \mathscr{M}(t, x)$). This is because one can easily construct a path for which the maximum value $\max_{\tau \in [t,t+T]} \kappa(\tau, p(\tau, t, x))$ of κ is constant with t and x; for example, a path that attempts to converge to an unsafe state. In this case, no control input will change the maximum value of κ along the path, but the assumption that $\|\frac{\partial \kappa(\eta, p(\eta,t,x))}{\partial \lambda_x}\frac{\partial p(\eta,t,x)}{\partial \lambda_x} g(t, x)\| \neq 0$ for all $\eta \notin \boldsymbol{M}(t, x)$ encodes that it is still possible to delay the time $\boldsymbol{R}(\tau, t, x)$ at which the state trajectory first becomes unsafe by choosing a control input u different from the nominal input $\mu(t, x)$.

The related assumption that $\frac{\partial \kappa(\tau, p(\tau,t,x))}{\partial \lambda_x}^{\mathrm{T}} \frac{\partial \kappa(\eta, p(\eta,t,x))}{\partial \lambda_x} \geq 0$ means that if the path is such that it is possible to change both 1) the maximum value of κ achieved along the path (first term of (7.11)) and 2) the first time along the path at which the trajectory becomes unsafe (second term of (7.11)), then there exists a control action which decreases both terms of (7.11) simultaneously. For example, a car moving towards a stop sign and decelerating at $|u|$ larger than $|\mu|$ will stop both *shorter* and *sooner* than the same car decelerating at $|\mu|$. That is, this assumption excludes the use of dynamics that would cause the car to instead stop *further* and *sooner*, or *shorter* and *later*. This is also reasonable mathematically, since, for $\boldsymbol{R} > \boldsymbol{M}$, the two terms of B in (7.19) (i.e. the first term of (7.19) and the negation of the first line of (7.21)) approach the same direction as $\boldsymbol{R}$ approaches $\boldsymbol{M}$.

In particular, I note that this section has not made any assumptions on the relative-degree of κ. For many systems, such as those in Section 7.1.4, this method will generate a CBF h_p even when κ is not a CBF. That is, using only the constraint function κ, the dynamics f and g, and a nominal, not necessarily safe, control law $\mu(t, x)$, one can directly utilize the sensitivities (7.16)-(7.18) of the nominal trajectory p to construct a P^{2}CBF and compute a safe trajectory. Note also that Theorems 7.2-7.5 do not assume that the system actually follows the "nominal" trajectory $p(\tau, t, x)$, even when this trajectory is safe. Rather, this trajectory is just used to generate the P^{2}CBF. However, I only *expect* a controller to proactively ensure safety better than traditional CBFs when h_p is constructed using a path that is relevant to the system's intended behavior (note that I use the term "expect" because I have not provided any guarantees of less myopic behavior, and have not even precisely defined "myopic", though the simulations in Section 7.1.4 show very promising results). Thus, one possible safe controller is

$$u(t, x) = \underset{u \in K_{\text{cbf}}(t,x)}{\arg\min} \|u - \mu(t, x)\|^2, \tag{7.31}$$

where K_{cbf} in (7.4) uses $h = h_p$ and where I assume that (7.31) is sufficiently regular for Lemma 7.1.

Next, while h_p in (7.13) is sufficient for enforcing safety, to make the controller even more proactive, one may want to consider future safety for all elements of $h_{p,\text{all}}$ in (7.12) simultaneously. However, it is in general difficult to show that $h_{p,\text{all}}$ in (7.12) is also a CBF (with appropriate generalizations for $h_{p,\text{all}}$ being set-valued), because there may not exist control inputs such that the CBF condition can be satisfied simultaneously for every output $h_{p,i}$. That said, it may still be advantageous to construct the CBF condition for every future time-of-interest and use slack penalties for violation of the CBF condition for $\tau_i \in \boldsymbol{M} \setminus \boldsymbol{M}^*$ in case of a conflict. Additionally, instead of enforcing the CBF condition on every $h_{p,i}$, one can reduce the size of $h_{p,\text{all}}$ by instead considering the maximal value of $h_{p,\text{all}}$ [151], the smooth maximum of $h_{p,\text{all}}$ [169], or by using hysteresis tolerances to make certain elements of $h_{p,\text{all}}$ inactive (see Section 4.4). In essence, the above considerations are analogous to controller tuning.

Remark 7.5. *To avoid needing to run a line-search for maximizers and roots in* (7.6)-(7.10)*, one could instead sample $\boldsymbol{M}$ as many discrete time points in the finite horizon (e.g. every ΔT time point), similar to MPC. With a suitable robustness margin for the discretization, this could also be used to guarantee safety in a similar manner. This strategy would increase the number of constraints on the control input, but would still be simpler than MPC approaches because only the current control input is an optimization variable in* (7.31)*.*

I now present simulations demonstrating the utility of the P^{2}CBF h_p.

7.1.4 Simulations and Comparisons

7.1.4.1 Autonomous Vehicles

The first case study presented involves two cars passing through a four-way intersection. I assume that the cars are fixed in their lanes $l_1 : \mathbb{R} \to \mathbb{R}^2$ and $l_2 : \mathbb{R} \to \mathbb{R}^2$, respectively, with locations z_1 and z_2 along their lanes. Thus, the position of car 1 on the road is $l_1(z_1)$ and the position of car 2 is $l_2(z_2)$. For simplicity, I model the cars as double-integrators: $\ddot{z}_1 = u_1$ and $\ddot{z}_2 = u_2$, resulting in state vector $x = [z_1;\, \dot{z}_1;\, z_2;\, \dot{z}_2]$ and control vector $u = [u_1; u_2]$. Suppose the cars nominally want to travel in their lanes at velocities v_1 and v_2, respectively, so the nominal control input is $\mu = [\mu_1; \mu_2]$ where $\mu_i(t, x) = k(v_i - \dot{z}_i)$ for some gain $k > 0$. The path function is then $p = [p_1;\, p_2]$ where p_i is the solution to (7.5) under μ:

$$p_i(\tau, t, x) = \begin{bmatrix} z_i + v_i(\tau - t) + \frac{\dot{z}_i - v_i}{k}(1 - e^{-k(\tau - t)}) \\ v_i + (\dot{z}_i - v_i)e^{-k(\tau - t)} \end{bmatrix}. \tag{7.32}$$

Let the safety constraint be $\kappa = \rho - \|l_1(z_1) - l_2(z_2)\|$.

I then constructed a P^{2}CBF of the form h_p, and simulated two intersection scenarios using the controller (7.31), one with two cars intersecting perpendicularly, and one with one car driving straight and a second car turning left and passing through l_1. The results were very similar, so I focus on the left-turn case. For comparison, I also simulated this case with safety governed by an Exponential CBF [85] (ECBF) and via Nonlinear MPC (NMPC) using the same prediction horizon. The control inputs are shown in Fig. 7.3 and the safety constraint is shown in Fig. 7.4. Videos of every simulation can be found at https://youtu.be/0tVUAX6MCno and all parameters and simulation code can be found at https://github.com/jbreeden-um/phd-code/tree/main/2022/CDC%20Predictive%20CBFs. Figs. 7.3-7.4 show that the P^{2}CBF and MPC performed generally similarly, with both cars approaching close and then continuing on opposite sides of the intersection. On the other hand, the ECBF caused both agents to come to a complete stop, so neither agent made it through the intersection. On average, the computation time per control computation was 0.0011 s for the ECBF, 0.0061 s for the P^{2}CBF, and 0.40 s for the NMPC approach, all on a 3.5 GHz CPU, though the code for all three cases could likely be further optimized.

7.1.4.2 Satellite Collision Avoidance

The second case study presented involves two satellites in Low Earth Orbit avoiding collision. Let r_1 and r_2 denote the satellite positions and let $\kappa(t, x) = \rho - \|r_1 - r_2\|$ where $\rho = 1$ km. Let the satellites be governed by two body dynamics. This scenario is not well solved by existing CBF

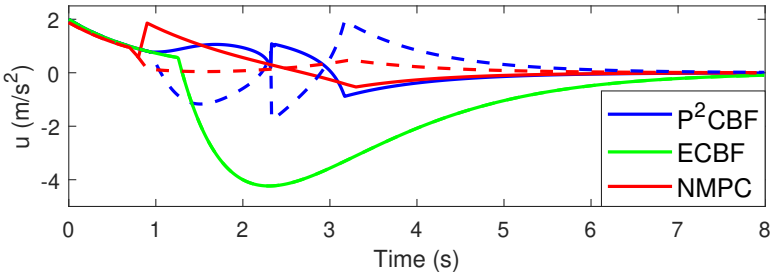

Figure 7.3: Control Inputs of Ground Vehicles. Control inputs of two vehicles with safety determined by an ECBF, P^{2}CBF, or by NMPC. The solid lines are u_1 (acceleration of vehicle 1) and the dashed lines are u_2 (acceleration of vehicle 2).

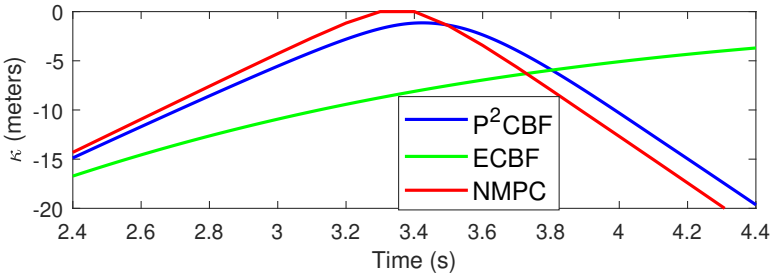

Figure 7.4: Safety Metric of Ground Vehicles. The values of κ during the three simulations in Fig. 7.3. The P^{2}CBF and MPC trajectories are similar, whereas the ECBF trajectory slowly converges to zero as the vehicles come to a complete stop. See also the animation: https://youtu.be/0tVUAX6MCno.

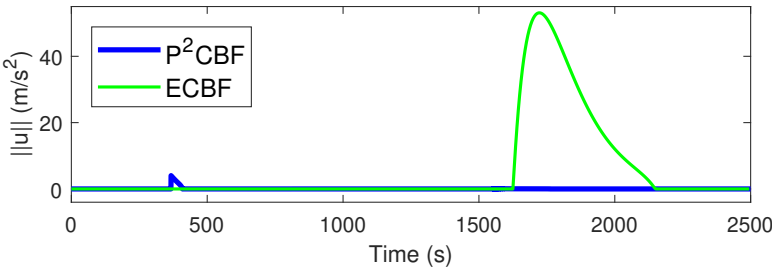

Figure 7.5: Control Inputs of Satellites. Control thrusts of a satellite with safety determined by an ECBF or P^{2}CBF.

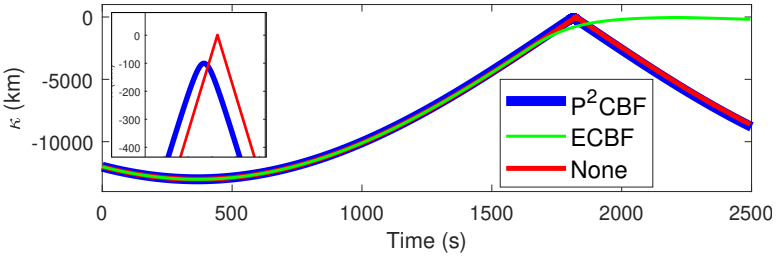

Figure 7.6: Safety Metric of Satellites. The values of κ during the two simulations in Fig. 7.5 and a simulation with zero control input. The P^{2}CBF minimally modifies the original trajectory so the red and blue lines are very similar, but the red line exceeds the safe set by 1 km (see zoomed in inset or the animation: https://youtu.be/HhtWUG63BWY), while the blue and green lines remain safe.

methods, or gradient methods in general, because the satellites necessarily orbit at large velocities, so κ may rapidly change from very negative to zero. If the satellites wait to take evasive maneuvers until κ is nearly zero, then large control actions will be necessary to maintain safety. However, very small control actions taken approximately a half orbit preceding a predicted collision (when κ is most negative) will result in large changes to r_1 and/or r_2 at the predicted collision time.

For this case study, let $x = [r_1; \dot{r}_1]$ represent the state of a controlled satellite and let r_2 be an uncontrolled piece of debris with a known orbit. Let p be the trajectory of the controlled satellite under zero control input, which is a well-characterized elliptical orbit. The simulation scenario places the controlled satellite and the debris initially very far apart, but in orbits that eventually intersect if no control action is taken. Simulations under h_p, under an ECBF, and under no control action are shown in Figs. 7.5-7.6 and in the video at https://youtu.be/HhtWUG63BWY. Note how the P^{2}CBF trajectory (blue) takes a small control action as soon as the unsafe prediction enters the horizon at $t = 367$ s, and then is very similar to the nominal trajectory (red). On the other hand, the ECBF trajectory (green) takes control action much later, when over 10 times as much thrust is required. This avoidance problem could in theory also be solved with NMPC, but would require a very fine discretization, because $\kappa > 0$ occurs for only 0.14 seconds since the satellites

are moving so fast. Thus, utilizing the same length of prediction horizon would require more than 10^4 samples, making the NMPC problem (and most other online trajectory search algorithms) intractable.

7.1.5 Conclusions

This section has developed a systematic approach for propagating state trajectories into the future along a nominal path, and then adjusting the trajectory if the propagated path is found to be unsafe. The approach reduces the degree to which the CBF condition intrudes on the nominal path compared to a traditional CBF in simulation, and performs similarly to an NMPC approach while reducing computation times by an order of magnitude and avoiding the need for linearizations and/or other approximations. The satellite collision avoidance simulation in particular shows how the P^{2}CBF can also solve problems that are not well handled by either traditional CBFs or MPC. Remaining questions include whether this approach can be extended to consider input constraints, similar to the Backup CBF, and extending the safety guarantees to collections of agents with distributed controllers.

7.2 Remarks on Performance, Assumptions, and Extensions

The above methodology presents a universal structure of CBF, and the simulation results in Section 7.1.4 suggest a substantial increase in performance for the cases tested. Thus, the P^{2}CBF on first reading appears to be a very powerful tool for control design. However, with any powerful tool, it is important to understand the assumptions and possible limitations. Since publishing this initial work, I have worked on better understanding these limitations, and in this subsection, I now discuss some (probably not *all*) of the non-obvious properties of the CBF (7.13) that I have since discovered. Some of these findings may be presented in a more organized fashion in future academic publications.

7.2.1 Performance

First, I revisit the question of input constraints. The main limitation of Theorem 7.5 is that it assumes the set of allowable control inputs is $\mathbb{R}^m$. However, Section 7.1.4 showed that the P^{2}CBF actually resulted in lesser control input usage. This begs the question: "can one design the P^{2}CBF to satisfy specified input constraints?". To answer this, recall that the zero sublevel set of a CBF must be a viable set. Thus, given a set of input constraints $\mathcal{U} \subset \mathbb{R}^m$, the set $\mathcal{H}_p$ in (7.14) should be designed to be a viable set. The primary "knobs" one can modify to change the size/shape of $\mathcal{H}_p$ are the function $m : \mathbb{R} \to \mathbb{R}$ in (7.11) and the prediction horizon T. In effect, the function m converts

the time until the system becomes unsafe $\boldsymbol{R} - t$ (e.g. units of seconds) into a relaxation on how large $\kappa(\boldsymbol{M}^*, p(\boldsymbol{M}^*, t, x))$ (e.g. units of meters) is permitted to be. A steeper choice of m will allow for initial conditions $x \in \mathcal{H}_p(t)$ that require more aggressive actions to remain within the CBF set. A more gradual choice of m will result in a smaller CBF set. At present, the choice of m has been made arbitrarily, and thus Theorem 7.5 has only been proven for the case of unconstrained control inputs. I hypothesize that there exists a more principled method of choosing m that will lead to provable input constraint satisfaction for certain systems. Even lacking provable guarantees, one can manually tune the m function to achieve more or less aggressive responses. That said, note that tuning m to achieve earlier, more fuel-efficient responses is inherently in opposition to deriving the largest possible controlled-invariant set, so an engineer must balance these competing objectives. Finally, as this chapter lacks an analytic proof of input constraint satisfaction regardless of how m is chosen, this is one area where learning a CBF, specifically learning the m function in (7.11), might be an advantageous method of tuning (in the prior chapters, I eschewed learning approaches because of the lack of provable guarantees, but in this case, I speculate that machine learning of m under the form (7.11) might provide a probabilistic guarantee of input constraint satisfaction that would enhance the present theory).

Likewise, the horizon T affects performance in multiple ways. Firstly, the maximum value of $\boldsymbol{R} - t$ in (7.11) is T, so $m(T)$ is the maximum relaxation afforded by the second term of (7.11). Thus, states (t, x) where $\kappa(\boldsymbol{M}^*, p(\boldsymbol{M}^*, t, x)) > m(T)$ are automatically excluded from the CBF set $\mathcal{H}_p$, whereas such states might still be allowable if a longer horizon were used. Secondly, I note that the controller (7.31) performs very differently depending on whether $\boldsymbol{M}^*$ is in $(t, t+T)$ or is one of the endpoints. If $\boldsymbol{M}^*(t, x) = t$ and κ is of high relative degree, then B in (7.19) is zero, so $u(t, x) = \mu(t, x)$ in (7.31). If κ represents obstacle avoidance and p provides a path through the obstacle, and if $\boldsymbol{M}^*(t, x) \in (t, t+T)$, then the first term of B in (7.19) will be a vector pointing orthogonal to p. This will encourage the controller (7.31) to maneuver around the obstacle. If, on the other hand, $\boldsymbol{M}^*(t, x) = t + T$, then the first term of B in (7.19) will be a vector pointing from the center of the obstacle to $p(t+T, t, x)$. This vector will not necessarily be orthogonal to p, and thus might encourage the controller (7.31) to decelerate ahead of the obstacle (as the ECBF did in Section 7.1.4) rather than to maneuver around it. Thus, the P^{2}CBF and the controller (7.31) perform better when T is sufficiently large that $\boldsymbol{M}^*$ is generally not equal to T. Alternatively, one might choose α sufficiently steep that the QP (7.31) outputs $u(t, x) = \mu(t, x)$ until the state gets close enough to the obstacle that $\boldsymbol{M}^*(t, x) < t + T$. For this reason, the P^{2}CBF is well suited to obstacle avoidance problems, where the system can navigate around a small excluded region of the state space, but the P^{2}CBF behaves more similarly to conventional CBFs when the unsafe set is similar to a long wall rather than a small excluded region.

The above paragraph also suggests two more peculiarities of the controller (7.31). Firstly, it is

not clear what is a good choice of α in (7.31). As mentioned in Section 2.4.2, this is true in general of CBF-based methods. However, the requirements on α summarized at the end of the proof of Theorem 7.5 further complicates this question. I have not yet studied this question sufficiently to make a recommendation, besides referring to the analysis surrounding (4.61). Secondly, it is worth studying the direction of the second term of B in (7.19) as well. This second term is a vector indicating which direction will cause $\boldsymbol{R}$ to most increase. In the context of obstacle avoidance, delaying $\boldsymbol{R}$ most often means decelerating. Thus, to avoid this deceleration, one seeks that the first term of B (i.e. the sensitivity of $\kappa(\boldsymbol{M}^*, p(\boldsymbol{M}^*, t, x))$) be greater in magnitude than the second term of B (i.e. the sensitivity of $m(\boldsymbol{R}-t)$). How to achieve this is also an open question for future work.

Finally, I emphasize that, despite some of the assumptions elaborated upon in Section 7.2.2, the P^{2}CBF is indeed a very generally applicable tool. Besides the way it improves obstacle avoidance specifically, it is also extremely useful for time-varying applications when the system needs to respond to things that may not happen until far into the future.

7.2.2 Assumptions

I now revisit and re-explain some of the assumptions and their implications. Firstly, the most limiting assumption is the assumption that the set of allowable control inputs is $\mathbb{R}^m$. That said, as in most control systems, one can tune the controller parameters, particularly the m function in (7.11), to achieve input constraint satisfaction. I also suspect that for certain systems one may be able to find more principled methods of choosing m that lead to provable input constraint satisfaction as in the prior chapters of this dissertation. Recall that this assumption was only invoked in cases B2 and B3 of the proof of Theorem 7.5. I suspect that case B3, the endpoint case, will be the more challenging case both for tuning and for finding provable extensions of Theorem 7.5 with $\mathcal{U} \neq \mathbb{R}^m$. Motivated by this, if one instead assumes that the prediction horizon is sufficiently long that $\boldsymbol{M}^*$ always occurs inside it, then one can ignore case B3.

Next, recall the assumption in Theorem 7.5 that $\frac{\partial \kappa(\tau, p(\tau,t,x,)) }{\partial \lambda_x}^{\mathrm{T}} \frac{\partial \kappa(\eta, p(\eta,t,x))}{\partial \lambda_x} \geq 0$. In my experience (i.e. working with physical agents in Newtonian environments and with second-order constraint functions) this assumption that has never been limiting, and I explained the intuitive meaning of it after presenting Theorem 7.5. That said, I note here that other researchers suggested that this may conflict with the dynamics of certain electronic systems. Thus, the reader should remember to verify this and other assumptions before applying this work.

Next, recall that Assumption 7.1 assumed that the maximum possible value of κ along any propagated trajectory originating from a state in $\mathcal{S}$ was bounded. A careful reader might notice that this assumption was never invoked in Theorem 7.2 or Theorem 7.5. As explained after the

assumption, the reason for this assumption is that if $\boldsymbol{M}^*(t,x) = t$ is one local maximizer and there exists another local maximizer at $t+T$, then there must be a local minimum between these two local maximizers. When the system passes this local minimum, the values of $\boldsymbol{M}^*$ and of $\kappa(\boldsymbol{M}^*, p(\boldsymbol{M}^*, t, x))$ jump, so the method requires that they jump in such a manner that h_p is less than zero before and after the jump. At this point, the reader may identify two flaws in the above logic. Firstly, Assumption 7.2 further assumed that h_p was absolutely continuous. This essentially prevents the phenomenon of $\boldsymbol{M}^*$ jumping values, as this would usually cause h_p to be discontinuous. Thus, one may conclude that Assumption 7.1 is unnecessary, because the phenomenon that it is intended to address is already forbidden by Assumption 7.2. Moreover, while Fig. 7.1 illustrates multiple maximizers and roots, the absolute continuity assumption makes this analysis irrelevant for the same reason. Section 7.2.3 discusses a fix for this logical oversight, and also explains the second logical flaw in Assumption 7.1.

Next, the strongest assumption of this section is the assumption that h_p is absolutely continuous in Assumption 7.2. While this holds for the simulated cases in Section 7.1.4, one can construct simple κ and p functions for which this does not hold. In general, the maximum of a function over an interval is usually continuous, and under fairly weak assumptions (e.g. see [205, Ch. 3]) is also directionally differentiable (which is nearly equivalent to absolute continuity). However, the maximizer (i.e. the point(s) at which the maximum occurs) may in general be discontinuous, and thus not differentiable either. Thus, to generalize Lemma 7.4, the first term of (7.11) should be treated as a single quantity and differentiated directly, rather than differentiating κ, p, and $\boldsymbol{M}$ separately. Similarly, the root $\boldsymbol{R}$ along a trajectory in general may be discontinuous and/or non-differentiable. For simple κ and p functions (e.g. straight line paths through concave κ functions), Lemma 7.4 and Theorem 7.5 often apply nicely, but I emphasize that they are only applicable under the assumption that $\boldsymbol{M}$ and $\boldsymbol{R}$ are sufficiently regular. Deriving easy-to-check conditions that certify this is one open area for future work.

Indeed, even when Assumption 7.2 does apply, absolute continuous functions may cause numerical issues. For instance, the formula (7.27d) contains an inverse, and the quantity being inverted may be zero. Indeed, the quantity being inverted is equivalent to $\frac{d}{d\boldsymbol{R}}[\kappa(\boldsymbol{R}, p(\boldsymbol{R}, t, x))]$. Since, $\kappa(\tau, p(\tau, t, x))$ is assumed continuously differentiable, this quantity is always zero when $\boldsymbol{R} \in \boldsymbol{M} \cap (t, t+T)$, i.e. on the boundary between the cases of (7.20). Thus, the matrices B and C in (7.19)-(7.21) become unbounded. In practice, this is not an issue because 1) an unbounded sensitivity matrix allows for the use of an arbitrarily small control input to satisfy (7.4), and 2) the system spends very little time on this boundary between the cases of (7.21). However, it is still undesirable to possibly have division by zero in the controller code.

Finally, one might note that the choice of κ used in the simulations in Section 7.1.4 is actually not continuously differentiable. There exists a single point where the norm function is not dif-

ferentiable. In the context of the P^{2}CBF, this non-differentiable point arises along a manifold of points where it is equally advantageous to go left or right (or an infinite number of directions if the space is three-dimensional, as in the satellite simulations) around an obstacle. In practice, numerical errors in the simulation cause the system to quickly leave this manifold of states for which $l_1(z_1) = l_2(z_2)$ occurs in the prediction horizon, but nonetheless, it is still undesirable to have such a non-differentiable manifold in the controller code. This can be fixed for instance by choosing $\kappa(t, x) = \rho^2 - \|l_1(z_1) - l_2(z_2)\|^2$, which is continuously differentiable. However, this choice results in the first term of B in (7.19) being very small near $\|l_1(z_1) - l_2(z_2)\| \approx 0$. As explained above, one generally achieves better performance when the first term of (7.19) is larger in magnitude than the second term. Thus, I suggest that it is better to account for this non-differentiability directly (e.g. by establishing an assumed direction for the gradient vector at this manifold) than to try to eliminate the manifold by changing κ.

7.2.3 Extensions

The prior subsection identified some potential flaws in the logic of Section 7.1.3, so this subsection now identifies some corrections. Firstly, in practice, the assumption that h_p is differentiable almost everywhere (absolutely continuous) is not satisfactory for practical controller design. Instead, one prefers that the P^{2}CBF and the controller based on the P^{2}CBF sensitivities should be well-defined everywhere (i.e. not almost everywhere). One can make progress towards accomplishing this by working with directional derivatives (also called "sub-gradients") instead of regular derivatives. However, this also introduces complexity, and the tools to work with directional derivatives in math and in code may be less developed. In any case, establishing differentiability of $\boldsymbol{M}^*$ and $\boldsymbol{R}$ is extremely challenging, so I suggest that it is better to work with $\kappa(\boldsymbol{M}^*, p(\boldsymbol{M}^*, t, x))$ directly and to develop additional tools to aid in the analysis of $\boldsymbol{R}$. Alternatively, I am also working on developing a hybrid system extension of this method that uses jumps to avoid areas where $k(\boldsymbol{M}^*, p(\boldsymbol{M}^*, t, x))$ and/or $\boldsymbol{R}$ are not continuously differentiable. While this then requires one to verify a hybrid logic, it has the advantage of avoiding the need for directional derivatives.

Next, the assumption that h_p is absolutely continuous should be relaxed, so that the method can handle multiple maximizers. I suspect that this extension will be rather straightforward to write out mathematically, though possibly difficult to guarantee in practice. In a hybrid system (e.g. see Section 5.3), the CBF conditions are that 1) the flows of the system satisfy the typical continuous-time CBF condition, e.g. (7.4), and 2) the jumps of the system always map states in the CBF set to states in the CBF set. That is, the image of $\mathcal{H}$ through the jump map must be a subset of $\mathcal{H}$. The same logic can be applied to the P^{2}CBF. One can relax Definition 7.1 to instead allow for jumps, so long as these jumps stay within the CBF set. This will then make Assumption 7.1 more

meaningful, and would allow for trajectories that pass through nonconvex obstacles or multiple obstacles in this framework. That said, even with such a relaxation, I note that Assumption 7.1 is still not sufficient. To see this, note that even if $\boldsymbol{M}^*$ jumps to $\boldsymbol{M}^* = t + T$, it may be the case that $\boldsymbol{R}(\boldsymbol{M}^*, t, x) < t+T$. Thus, the relaxation $m(\boldsymbol{R}-t)$ will be less than $m(T)$, so the assumption that $m(T) \geq h_{\text{max}}$ in Assumption 7.1 does not fulfill its intended purpose. This is the second logical flaw mentioned in Section 7.2.2. Determining the appropriate replacement for Assumption 7.1 to properly fulfill this purpose is an area for future work.

CHAPTER 8

Conclusion

8.1 Conclusions

This dissertation presents a series of tools that advance the state of the art of control barrier functions and quadratic-program-based safety filters for constrained control design. Recall that the problem that motivated this work was the goal of constructing and verifying computationally efficient control strategies for space systems with constraints. Control barrier functions (CBFs) as safety filters provide solutions to both problems simultaneously. Given a CBF, one can always construct a safety-preserving controller using the quadratic program in Section 2.4, so CBFs are constructive. Equally important, this quadratic program provably yields set invariance of the CBF set, so the work in this dissertation removes (or at least lessens) the need for additional controller verification. That said, at the start of this work—i.e. at the end of Chapter 2—this seemingly powerful method was difficult to apply to space systems. Now, with all the methods in this dissertation taken together, engineers have the tools to apply CBF-based safety filters to a range of space systems, and have a basis for deriving further extensions if necessary.

Specifically, Chapter 1 presented an overview of the constrained control literature and a deeper review of the CBF literature in particular. Chapter 2 presented a comprehensive technical review of the general CBF method to 1) establish a baseline for the rest of the dissertation, and 2) clarify certain non-obvious technicalities of this method for the reader. Chapter 2 also identified the connection between CBFs and viability theory; this connection is useful because it allows engineers to compare what behaviors are allowed by physics and what behaviors are allowed by a particular safety-filter (which is usually much more conservative).

Chapter 3 presented two constructive tools for designing CBFs for systems with inertia and input constraints (i.e. any real space system), and these tools were improved upon and added to in Chapter 4. These chapters thus provide much-needed practical tools to construct CBFs for space systems problems.

Chapter 4 then extended CBFs from the deterministic setting to the setting of systems with

bounded disturbances. This chapter introduced a new "robust CBF condition" in place of the condition in Chapter 2, and re-proved the results on set invariance in that chapter now for perturbed systems. Chapter 4 also extended the tools in Chapter 3 to the robust case, and importantly, the methods in Chapter 4 maintain input constraint satisfaction for any disturbance in the assumed bounds. Finally, Chapter 4 introduces the notion of a "tight-tolerance problem", and shows how robust CBFs can be used to provably achieve such tolerances. This chapter thus makes the provable guarantee of set invariance provided by CBFs and safety filters more relevant in practice, especially in light of how controls engineers almost always work with simplified models and possibly other uncertainties.

Chapter 5 extends all of the work up to this point to the sampled-data regime. Section 5.1 first extends the results in Chapter 2 to sampled-data systems, and introduces two metrics of conservatism. I emphasize that these metrics are also generalizable to other types of robustness. Section 5.2 then generalizes the new developments in Chapter 4 to sampled-data systems as well. Finally, Section 5.3 shows how the same tools can be applied to systems with impulsive actuators with a minimum dwell-time between impulses. This chapter thus generalizes safety filters from the continuous domain to some of the actuators more commonly deployed on spacecraft, and again makes the provable guarantee of set invariance provided by this method more relevant.

Chapter 6 addresses a technical problem in the scalability of safety filters to multiple constraints, and provides constructive tools for how to modify individual CBFs to ensure that all CBFs are compatible together in a single safety filter. The eventual goal of safety filters is to allow agents to be more autonomous in complex environments, so this chapter thus helps elevate CBFs from simple constraints to more complicated and possibly multi-agent environments.

Chapter 7 proposes a new form of CBF that both adds to the methods in Chapters 3-4 and solves new problems. In particular, CBFs (and constrained control methodologies in general) tend to be conservative in the presence of time-varying obstacles, and difficult to apply when the magnitude of the uncontrolled dynamics is much larger than the magnitude of the control input. Chapter 7 solves these problems by proposing a computationally-efficient way to consider the safety of future trajectories of the system when choosing the current control action. As a result, this is the only method in this dissertation that is applicable (without substantial conservatism) to satellites in dissimilar low Earth orbits (i.e. where relative motion approximations do not apply), such as might occur for a constellation of communication satellites. While it is too soon to know which method in this dissertation will prove most useful in practice, I can say now that this is the most theoretically novel tool and the most significant "leap" in control algorithm design accomplished in this dissertation. Chapter 7 thus generalizes CBFs to another important domain for space systems.

In summary, the prior chapters all present a piece of the many new tools needed to apply CBFs and safety filters to space systems. Additional pieces remain unsolved, but the above nonetheless

provides a foundation for many practical problems and from which to tackle the remaining theoretical problems. While this work was all motivated by the goal of enhancing space systems, the theoretical problems encountered along the way are not specific to space systems. Thus, the above tools are all contributions to general control theory and can be applied to other systems meeting the theorem requirements as well. Many of the observations made in this dissertation are also relevant to other constrained control methodologies, such as how the sampling margins studied in Chapter 5 are equally relevant to model predictive control. I hope that these tools are useful for future engineers looking to solve space systems challenges, and to future researchers looking for constrained control tools for problems that I have not yet even imagined.

8.2 Future Work

While this work is expansive, there are certainly many areas still open for improvement. Some specific areas were suggested at the ends of Chapters 3-7, so here I highlight the most important (i.e. the most frequently recurring) and the most high-level of these topics. The most glaring area for future work is the need for greater fuel optimality. Because safety filters of the sort introduced in Section 2.4 only consider the current state, this framework does not immediately admit any way to consider fuel optimality over a trajectory. Thus, future researchers may consider studying more computationally complex tools that yield optimality or near-optimality guarantees, the potential role of CBFs in multi-rate planning frameworks, and/or whether safety filters can be used to "gracefully degrade" online a pre-computed optimal trajectory. When studying this question, researchers will likely contend with the "no free lunch theorem"—i.e. how no gain in performance in one area comes without costs in another area—and thus will have to sacrifice computational simplicity for greater fuel optimality. The proper balance between these two goals will likely be determined by industry engineers rather than researchers. Another consideration is that a continuously running computation like a safety-filter is more practical to run on an embedded computer, whereas large, infrequent trajectory planning computations might require scheduled pauses in nominal operations. Note that Chapter 7 indeed improved fuel optimality substantially, but this improvement was largely coincidental. That is, there is no way (that I know of) to prove that the method in Chapter 7 will yield lower fuel consumption. Still, knowing that there exist CBFs that reduce fuel consumption makes me hopeful that some variation of the safety filter method can be practically applied in fuel-critical scenarios.

Next, this dissertation only considered a subset of possible spacecraft dynamics. As long as there are mission concepts that researchers have still not imagined, there will always be a need to derive CBFs (and other Lyapunov tools) for those specific systems. The tools in Chapters 3-4 and Chapter 7 (and the broader CBF literature) form a baseline for these future developments. Like-

wise, Section 4.7 considered the possibility that a conservative robust safety filter might conflict with main mission objectives. Along this line of thought, future engineers will need to design tools for other varieties of CBFs to ensure that these CBFs do not conflict with objectives of the missions to which these CBFs are applied. Tools such as the "feasibility margin" in Section 4.7 are particularly relevant here, as they inform the engineer of what is maximally possible with a method. Similar tools should be developed for other mission scenarios.

The tools in Chapters 4-5 considered safety filter robustness to two specific phenomena, in this case bounded disturbances and controller sampling. Future work might consider robustness to other phenomena, such as measurement uncertainty, input delay, large persistent disturbances (e.g. wind gusts, periods of high solar activity, etc.), and others. In particular, current filtering techniques often lead to Gaussian measurement uncertainties, so many requirements are written in terms of 3σ behaviors. Such disturbance models are not immediately compatible with the bounded disturbances work in this dissertation. Thus, an open problem is going from bounded disturbances to stochastic distributions with tails. This has partially been studied via "risk-bounded CBFs", but there is still work to be done in relating pointwise statistics to statistics of a full trajectory, and in doing this analysis in a minimally conservative manner. A method for bridging this gap from pointwise to full-trajectory statistics may also be relevant to methods for considering full-trajectory fuel-optimality. Additionally, zero-order-hold and impulsive actuators are only a subset of possible actuator behaviors. Other common constraints include actuators that only operate on-off—i.e. without continuous variation between these endpoints—and impulsive actuators with a minimum impulse magnitude.

Next, one of the motivating applications for safety filters that was not studied in this dissertation is multi-agent control. I qualify this by pointing out that "multi-agent control" is a very vague term, especially in the spacecraft domain. The term can apply to satellites in passive relative orbits, in high-precision formations, in loose formations, in distributed interferometers, in a network of resource-scouting spacecraft with no hard separation requirements, or to a group of planetary rovers, ice-tunneling snake robots, etc.. The tools to achieve all of these tasks look very different from each other. The motivation for using safety-filters for multi-agent applications in general is that by giving each agent responsibility for its own safety, engineers can reduce the need for expensive centralized computation (in particular, engineers might be able to avoid the "hole" phenomenon that exclusion constraints cause, and thus be able run convex centralized planning routines instead of non-convex planning-and-avoidance routines) and make the system more robust to single-agent failures. However, the hypothesis that safety filters will help achieve this is limited by (at least) two observations. First, designing safety filters that enforce multiple constraints simultaneously is a nontrivial problem, as shown in Chapter 6. Enforcing multiple constraints required redesigning every CBF in the safety filter to be compatible with every other CBF. This

process is in direct opposition to the goal of a scalable, flexible multi-agent system. Second, it is difficult to generalize local safety constraints to whole-system safety constraints. Indeed, with only three or four double-integrator agents, it is easy to construct examples where one agent becomes trapped between other agents with no viable control action. Two solutions to this are to perform some amount of centralized planning and deconfliction (possibly in a multi-rate architecture), or to apply system-level rules or heuristics, analogous to the rules of driving on a shared road. On a road, drivers agree to stay in their lines and obey traffic rules; these rules may make individual drivers' routes less efficient, and may in fact make the net system's operations suboptimal, but similar rules will be necessary for distributed safety-filter applications to exhibit provable safety guarantees. The alternative of having a centralized planner dictate every possible action to agents will require much more computational power than is likely to exist soon. Thus, there exists a need for further and more complex extensions to Chapter 6.

One should also consider the types of problems to which CBFs and safety filters apply well, and those to which they do not. For one, the safety filters in this thesis all implicitly required that the system be fully controllable. For example, the simulations in Chapters 3-4 all assumed that the system had a thruster pointing in every direction. In practice, satellites may have a single large thruster that requires rotating the satellite to thrust in different directions, thus creating a nonholonomic system. There is some work on nonholonmic and mixed relative-degree CBFs, but this has yet to be tested on satellite simulations. CBFs could also be further extended to underactuated systems, where the system needs to stay in a particular configuration (i.e. a particular subset of the state space that needs to be computed) to maintain feasible operations (consider for instance the James Webb Space Telescope, which must always stay on a particular side of the Sun-Earth Lagrange point because it possesses no thrusters pointing towards the Sun; this could be a natural CBFs problem but would likely need further tools). Next, CBFs and safety filters often work best when there is a clear set of states to be rendered forward invariant. In problems where such a set is not clearly defined, or where such a set can be violated under certain conditions or for limited amounts of time, the CBFs studied in this work may not apply well. The constraint sets used in the simulations throughout this dissertation were also relatively simple—usually two-norm exclusion constraints—whereas it may be more difficult to apply these methods to more complex constraint set geometries. Finally, one should consider how most satellite systems operate without breaks every day of the year. As a result, satellites spend a majority of that time holding their position/orientation or applying no active control. For example, though this dissertation studied the satellite reorientation case study in Section 5.2, a satellite is likely to spend far more time holding an orientation than reorienting. Likewise, a satellite formation mission that uses a reasonable amount of fuel will need to spend far more time resting in orbit than thrusting. The safety filters studied in this dissertation are an inherently "active" method, so the control of the satellite during

these more "passive" phases of operation should also be considered, especially for multi-agent applications.

This dissertation made contributions to deriving practical CBFs for space systems, but there are still other obstacles to using CBFs in practice. For one, the need for high expertise in this subject area to easily apply these methods is one obstacle. Moreover, all practical control systems need some amount of tuning. For safety filters, much of this tuning appears via the choice of the class-$\mathcal{K}$ function. At present, it is not clear how this function should be chosen. Section 4.4 provides one suggestion, and other works provide other reasons for specific choices of this function, but ultimately choosing this function still largely depends on the engineer's own experience with safety filters. Other guidelines/heuristics or easily tunable parameters would also make CBFs more widely useful.

Finally, recall that one of the objectives of this work was to enable increased trust in machine-learning-based control laws, by wrapping such control laws in a model-based and provably-safe safety filter. Though this was a major inspiration for this work, the nominal control laws in all the case studies in this dissertation were always very simple control laws (usually control laws deliberately intended to cause safety violations). Having established functionality under this setup, the next step is to replace this nominal control law with a learning-based control law, and to conduct further experiments that may identify other interesting behaviors arising from this combination. Such behaviors could also arise in other more in-depth mission applications with safety filters, so this is a wide area for future experimentation.

8.3 Closing Practical Remarks

I conclude with some of my own predictions for how safety filters and constrained control methodologies may be used in the future. First, computational hardware continues to advance quickly, and optimization tools are also continuing to improve. Much of this dissertation concerned itself with alternatives to model predictive control (MPC), under the assumption that MPC might be too computationally complex for space systems. On the other hand, if one possesses the computational power to run nonlinear MPC (i.e. without linearization approximation error that would remove the provable guarantees of safety), then this strategy will almost certainly yield better fuel efficiency than the pointwise safety-filter design as in Section 2.4. That said, I emphasize again my point from Section 1.2.2 that MPC should always be used with a constraint that the terminal state belong in a known controlled-invariant set, i.e. a CBF set. Thus, the tools for computing such a set in Chapters 3-4 and Chapter 6 are still relevant to MPC. The robustness margins in Chapters 4-5 are also relevant to MPC, and I am curious how the continuous safety prediction and search suggested in Chapter 7 might also be relevant to new MPC strategies.

Next, I repeat my prediction in Section 1.2.2 that machine-learning will be one of the primary approaches for constrained control in the future. As such, safety filters that work well with learning-based control laws may also become widespread. That said, despite the increases in algorithmic capabilities and computational power coming to fruition, to some degree engineers will likely always want to know in advance (i.e. on the ground) that the systems they launch will indeed work as expected. Thus, ground-based planning will not entirely disappear. For similar reasons, engineers will likely derive the explicit piecewise solutions to the above QPs a priori rather than relying on online optimization solvers. That said, as the number of active satellites increases, engineers will have more incentive to use control laws with higher degrees of autonomy and more onboard computations. As a result, engineers will likely gradually become more comfortable trusting such methods, except in the most critical, or the most uncertain, applications.

Finally, recall that space systems design is inherently a systems engineering problem, so engineers often have to take a systems-level perspective. A more powerful computer to run an advanced algorithm will require more mass and power (which will require larger solar arrays or power-generation equipment, which will also add mass), but may alleviate other complexities. Based on my past experience, I am generally of the opinion that relegating problems to software (i.e. control) that could have been solved by better hardware is a bad idea, but nonetheless that is a tradeoff that systems engineers can make. And indeed one of the objectives of control theory, and this dissertation, is to make software more capable so that hardware can be made cheaper. In conclusion, the tools proposed in this dissertation add to an already large set of controls tools to which engineers can refer to potentially make different systems tradeoffs than were made in past missions, and to find solutions for future engineering problems.

BIBLIOGRAPHY

[1] Notebaert, O., "On-board payload data processing requirements and technology trends," *European Workshop on On-Board Data Processing*, 2019. URL https://indico.esa.int/event/225/contributions/4298/attachments/3359/4397/OBDP2019-S01-05-Airbus_Notebaert_On-Board_Payload_Data_Processing_Requirements_and_Technology_Trends.pdf.

[2] Lentaris, G., Maragos, K., Stratakos, I., Papadopoulos, L., Papanikolaou, O., Soudris, D., Lourakis, M., Zabulis, X., Gonzalez-Arjona, D., and Furano, G., "High-Performance Embedded Computing in Space: Evaluation of Platforms for Vision-Based Navigation," *Journal of Aerospace Information Systems*, Vol. 15, No. 4, 2018, pp. 178–192. https://doi.org/10.2514/1.I010555.

[3] Seto, D., Krogh, B., Sha, L., and Chutinan, A., "The Simplex architecture for safe online control system upgrades," *Proceedings of the 1998 American Control Conference. ACC (IEEE Cat. No.98CH36207)*, Vol. 6, 1998, pp. 3504–3508. https://doi.org/10.1109/ACC.1998.703255.

[4] Vivekanandan, P., Garcia, G., Yun, H., and Keshmiri, S., "A Simplex Architecture for Intelligent and Safe Unmanned Aerial Vehicles," *2016 IEEE 22nd International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA)*, 2016, pp. 69–75. https://doi.org/10.1109/RTCSA.2016.17.

[5] Flower, J. O., and Parr, E. A., "Control Systems," *Electrical Engineer's Reference Book (Sixteenth Edition)*, edited by M. Laughton and D. Warne, Newnes, 2003, Chap. 13. https://doi.org/10.1016/B978-075064637-6/50013-7.

[6] Schlereth, F. H., "A method of calculating gain and phase margins from pole-zero diagrams," *Transactions of the American Institute of Electrical Engineers, Part I: Communication and Electronics*, Vol. 80, No. 2, 1961, pp. 139–143. https://doi.org/10.1109/TCE.1961.6373090.

[7] Goddard Space Flight Center, "Rules for the Design, Development, Verification, and Operation of Flight Systems," Tech. Rep. GSFC-STD-1000H, National Aeronautics and Space Administration, March 2023. URL https://standards.nasa.gov/sites/default/files/standards/GSFC/H/0/GSFC-STD-1000RevH_Approved.pdf.

[8] Clohessy, W. H., and Wiltshire, R. S., "Terminal Guidance System for Satellite Rendezvous," *Journal of the Aerospace Sciences*, Vol. 27, No. 9, 1960, pp. 653–658. https://doi.org/10.2514/8.8704.

[9] Hill, G. W., "Researches in the Lunar Theory," *American Journal of Mathematics*, Vol. 1, No. 1, 1878, pp. 5–26. https://doi.org/10.2307/2369430.

[10] Davis, W. R., "Control of the Relative Motion Between Satellites in Neighboring Elliptical Orbits," Ph.D. thesis, Stanford University, 1966. URL https://ntrs.nasa.gov/api/citations/19660022831/downloads/19660022831.pdf.

[11] Lane, C., and Axelrad, P., "Formation Design in Eccentric Orbits Using Linearized Equations of Relative Motion," *Journal of Guidance, Control, and Dynamics*, Vol. 29, No. 1, 2006, pp. 146–160. https://doi.org/10.2514/1.13173.

[12] Hamel, J.-F., and de Lafontaine, J., "Linearized Dynamics of Formation Flying Spacecraft on a J2-Perturbed Elliptical Orbit," *Journal of Guidance, Control, and Dynamics*, Vol. 30, No. 6, 2007, pp. 1649–1658. https://doi.org/10.2514/1.29438.

[13] Wei, C., Park, S.-Y., and Park, C., "Linearized dynamics model for relative motion under a J2-perturbed elliptical reference orbit," *International Journal of Non-Linear Mechanics*, Vol. 55, 2013, pp. 55–69. https://doi.org/10.1016/j.ijnonlinmec.2013.04.016.

[14] Markley, F. L., and Crassidis, J. L., *Fundamentals of Spacecraft Attitude Determination and Control*, Springer-Verlag New York, 2014. https://doi.org/10.1007/978-1-4939-0802-8.

[15] Warren, C., "Global path planning using artificial potential fields," *1989 IEEE International Conference on Robotics and Automation*, IEEE Computer Society, Los Alamitos, CA, USA, 1989, pp. 316,317,318,319,320,321. https://doi.org/10.1109/ROBOT.1989.100007.

[16] Warren, C., "Multiple robot path coordination using artificial potential fields," *Proceedings., IEEE International Conference on Robotics and Automation*, 1990, pp. 500–505 vol.1. https://doi.org/10.1109/ROBOT.1990.126028.

[17] Panagou, D., Stipanovič, D. M., and Voulgaris, P. G., "Multi-objective control for multi-agent systems using Lyapunov-like barrier functions," *2013 IEEE 52nd Conference on Decision and Control*, 2013, pp. 1478–1483. https://doi.org/10.1109/CDC.2013.6760091.

[18] Panagou, D., Stipanović, D. M., and Voulgaris, P. G., "Distributed Coordination Control for Multi-Robot Networks Using Lyapunov-Like Barrier Functions," *IEEE Transactions on Automatic Control*, Vol. 61, No. 3, 2016, pp. 617–632. https://doi.org/10.1109/TAC.2015.2444131.

[19] Lee, U., and Mesbahi, M., "Constrained consensus via logarithmic barrier functions," *2011 IEEE 50th Conference on Decision and Control and European Control Conference*, 2011, pp. 3608–3613. https://doi.org/10.1109/CDC.2011.6161496.

[20] Den Hertog, D., Roos, C., and Terlaky, T., "On the classical logarithmic barrier function method for a class of smooth convex programming problems," *Journal of Optimization Theory and Applications*, Vol. 73, 1992. https://doi.org/10.1007/BF00940075.

[21] Lopez, I., and Mclnnes, C. R., "Autonomous rendezvous using artificial potential function guidance," *Journal of Guidance, Control, and Dynamics*, Vol. 18, No. 2, 1995, pp. 237–241. https://doi.org/10.2514/3.21375.

[22] Gazi, V., and Passino, K. M., "Stability analysis of swarms," *IEEE Transactions on Automatic Control*, Vol. 48, No. 4, 2003, pp. 692–697. https://doi.org/10.1109/TAC.2003.809765.

[23] Saaj, C. M., Lappas, V., and Gazi, V., "Spacecraft Swarm Navigation and Control Using Artificial Potential Field and Sliding Mode Control," *2006 IEEE International Conference on Industrial Technology*, 2006, pp. 2646–2651. https://doi.org/10.1109/ICIT.2006.372712.

[24] An, M., Wang, Z., and Zhang, Y., "Self-organizing control strategy for asteroid intelligent detection swarm based on attraction and repulsion," *Acta Astronautica*, Vol. 130, 2017, pp. 84–96. https://doi.org/10.1016/j.actaastro.2016.10.038.

[25] Dong, H., Hu, Q., and Akella, M. R., "Safety Control for Spacecraft Autonomous Rendezvous and Docking Under Motion Constraints," *Journal of Guidance, Control, and Dynamics*, Vol. 40, No. 7, 2017, pp. 1680–1692. https://doi.org/10.2514/1.G002322.

[26] Mclnnes, C. R., "Large angle slew maneuvers with autonomous sun vector avoidance," *Journal of Guidance, Control, and Dynamics*, Vol. 17, No. 4, 1994, pp. 875–877. https://doi.org/10.2514/3.21283.

[27] Lee, U., and Mesbahi, M., "Spacecraft reorientation in presence of attitude constraints via logarithmic barrier potentials," *Proceedings of the 2011 American Control Conference*, 2011, pp. 450–455. https://doi.org/10.1109/ACC.2011.5991284.

[28] Wieland, P., and Allgöwer, F., "Constructive Safety Using Control Barrier Functions," *7th IFAC Symposium on Nonlinear Control Systems*, Vol. 40, No. 12, 2007, pp. 462–467. https://doi.org/10.3182/20070822-3-ZA-2920.00076.

[29] Ames, A. D., Xu, X., Grizzle, J. W., and Tabuada, P., "Control Barrier Function Based Quadratic Programs for Safety Critical Systems," *IEEE Transactions on Automatic Control*, Vol. 62, No. 8, 2017, pp. 3861–3876. https://doi.org/10.1109/TAC.2016.2638961.

[30] Agrawal, A., and Sreenath, K., "Discrete Control Barrier Functions for Safety-Critical Control of Discrete Systems with Application to Bipedal Robot Navigation," *Proceedings of Robotics: Science and Systems*, Cambridge, Massachusetts, 2017. https://doi.org/10.15607/RSS.2017.XIII.073.

[31] Singletary, A., Chen, Y., and Ames, A. D., "Control Barrier Functions for Sampled-Data Systems with Input Delays," *2020 IEEE 59th Conference on Decision and Control*, 2020, pp. 804–809. https://doi.org/10.1109/CDC42340.2020.9304281.

[32] Chai, J., and Sanfelice, R. G., "Forward Invariance of Sets for Hybrid Dynamical Systems (Part I)," *IEEE Transactions on Automatic Control*, Vol. 64, No. 6, 2019, pp. 2426–2441. https://doi.org/10.1109/TAC.2018.2882158.

[33] Chai, J., and Sanfelice, R. G., "Forward Invariance of Sets for Hybrid Dynamical Systems (Part II)," *IEEE Transactions on Automatic Control*, Vol. 66, No. 1, 2021, pp. 89–104. https://doi.org/10.1109/TAC.2020.2990665.

[34] Srinivasan, M., Hyun, N. P., and Coogan, S., "Weighted Polar Finite Time Control Barrier Functions With Applications To Multi-Robot Systems," *2019 IEEE Conference on Decision and Control*, 2019, pp. 7031–7036. https://doi.org/10.1109/CDC40024.2019.9029263.

[35] Xiao, W., Cassandras, C. G., Belta, C. A., and Rus, D., "Control Barrier Functions for Systems with Multiple Control Inputs," *2022 American Control Conference*, 2022, pp. 2221–2226. https://doi.org/10.23919/ACC53348.2022.9867612.

[36] Wang, L., Ames, A. D., and Egerstedt, M., "Safety Barrier Certificates for Collisions-Free Multirobot Systems," *IEEE Transactions on Robotics*, Vol. 33, No. 3, 2017, pp. 661–674. https://doi.org/10.1109/TRO.2017.2659727.

[37] Grover, J. S., Liu, C., and Sycara, K., "Deadlock Analysis and Resolution for Multi-robot Systems," *Algorithmic Foundations of Robotics XIV*, edited by S. M. LaValle, M. Lin, T. Ojala, D. Shell, and J. Yu, Springer International Publishing, Cham, 2021, pp. 294–312. https://doi.org/10.1007/978-3-030-66723-8_18.

[38] Singletary, A., Klingebiel, K., Bourne, J., Browning, A., Tokumaru, P., and Ames, A., "Comparative Analysis of Control Barrier Functions and Artificial Potential Fields for Obstacle Avoidance," *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2021, pp. 8129–8136. https://doi.org/10.1109/IROS51168.2021.9636670.

[39] Kapasouris, P., Athans, M., and Stein, G., "Design of feedback control systems for unstable plants with saturating actuators," *Symposium on Nonlinear Control Systems Design*, 1988. URL https://ntrs.nasa.gov/citations/19890005006.

[40] Kolmanovsky, I., Garone, E., and Di Cairano, S., "Reference and command governors: A tutorial on their theory and automotive applications," *2014 American Control Conference*, 2014, pp. 226–241. https://doi.org/10.1109/ACC.2014.6859176.

[41] Bemporad, A., "Reference governor for constrained nonlinear systems," *IEEE Transactions on Automatic Control*, Vol. 43, No. 3, 1998, pp. 415–419. https://doi.org/10.1109/9.661611.

[42] Nicotra, M. M., and Garone, E., "The Explicit Reference Governor: A General Framework for the Closed-Form Control of Constrained Nonlinear Systems," *IEEE Control Systems Magazine*, Vol. 38, No. 4, 2018, pp. 89–107. https://doi.org/10.1109/MCS.2018.2830081.

[43] Rabiee, P., and Hoagg, J. B., "Soft-Minimum Barrier Functions for Safety-Critical Control Subject to Actuation Constraints," *2023 American Control Conference*, 2023, pp. 2646–2651. https://doi.org/10.23919/ACC55779.2023.10156245.

[44] Nicotra, M. M., Liao-McPherson, D., Burlion, L., and Kolmanovsky, I. V., "Spacecraft Attitude Control With Nonconvex Constraints: An Explicit Reference Governor Approach," *IEEE Transactions on Automatic Control*, Vol. 65, No. 8, 2020, pp. 3677–3684. https://doi.org/10.1109/TAC.2019.2951303.

[45] Kalabić, U., Gupta, R., Di Cairano, S., Bloch, A., and Kolmanovsky, I., "Constrained spacecraft attitude control on SO(3) using reference governors and nonlinear model predictive control," *2014 American Control Conference*, 2014, pp. 5586–5593. https://doi.org/10.1109/ACC.2014.6858865.

[46] Semeraro, S., Kolmanovsky, I., and Garone, E., "Reference governor for constrained spacecraft orbital transfers," *Advanced Control for Applications*, 2024. https://doi.org/10.1002/adc2.179.

[47] Tan, X., and Dimarogonas, D. V., "Construction of control barrier function and C2 reference trajectory for constrained attitude maneuvers," *2020 IEEE 59th Conference on Decision and Control*, 2020, pp. 3329–3334. https://doi.org/10.1109/CDC42340.2020.9304324.

[48] Souissi, O., Benatitallah, R., Duvivier, D., Artiba, A., Belanger, N., and Feyzeau, P., "Path planning: A 2013 survey," *Proceedings of 2013 International Conference on Industrial Engineering and Systems Management*, 2013. URL https://ieeexplore.ieee.org/abstract/document/6761521.

[49] Karur, K., Sharma, N., Dharmatti, C., and Siegel, J. E., "A Survey of Path Planning Algorithms for Mobile Robots," *Vehicles*, Vol. 3, No. 3, 2021, pp. 448–468. https://doi.org/10.3390/vehicles3030027.

[50] Costa, M. M., and Silva, M. F., "A Survey on Path Planning Algorithms for Mobile Robots," *2019 IEEE International Conference on Autonomous Robot Systems and Competitions*, 2019. https://doi.org/10.1109/ICARSC.2019.8733623.

[51] Phillips, J., Kavraki, L., and Bedrossian, N., "Spacecraft Rendezvous and Docking with Real-Time, Randomized Optimization," *AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2003. https://doi.org/10.2514/6.2003-5511.

[52] Frazzoli, E., "Quasi-random algorithms for real-time spacecraft motion planning and coordination," *Acta Astronautica*, Vol. 53, No. 4, 2003, pp. 485–495. https://doi.org/10.1016/S0094-5765(03)80009-7.

[53] Rybus, T., Seweryn, K., and Sąsiadek, J. Z., "Optimal detumbling of defunct spacecraft using space robots," *2014 19th International Conference on Methods and Models in Automation and Robotics*, 2014, pp. 64–69. https://doi.org/10.1109/MMAR.2014.6957326.

[54] Rybus, T., and Seweryn, K., "Application of Rapidly-exploring Random Trees (RRT) algorithm for trajectory planning of free-floating space manipulator," *2015 10th International Workshop on Robot Motion and Control*, 2015, pp. 91–96. https://doi.org/10.1109/RoMoCo.2015.7219719.

[55] Huntington, G. T., and Rao, A. V., "Optimal Reconfiguration of Spacecraft Formations Using the Gauss Pseudospectral Method," *Journal of Guidance, Control, and Dynamics*, Vol. 31, No. 3, 2008, pp. 689–698. https://doi.org/10.2514/1.31083.

[56] Banerjee, A., Amrr, S. M., and Nabi, M., "A pseudospectral method based robust-optimal attitude control strategy for spacecraft," *Advances in Space Research*, Vol. 64, No. 9, 2019, pp. 1688–1700. https://doi.org/10.1016/j.asr.2019.08.008.

[57] Hart, S. T., Ayoubi, M. A., and Naseradinmousavi, P., "Comparative Study of Pseudospectral Methods for Spacecraft Optimal Attitude Maneuvers," *Journal of Spacecraft and Rockets*, Vol. 59, No. 1, 2022, pp. 178–189. https://doi.org/10.2514/1.A35081.

[58] Tu, L., Wang, Y., Yan, C., and Yang, Y., "Optimal transfer orbit design of spacecraft with finite thrust based on Legendre pseudospectral method," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, Vol. 237, No. 8, 2023, pp. 1791–1797. https://doi.org/10.1177/09544100221138164.

[59] Morari, M., Garcia, C. E., and Prett, D. M., "Model predictive control: Theory and practice," *IFAC Proceedings Volumes*, Vol. 21, No. 4, 1988. https://doi.org/10.1016/B978-0-08-035735-5.50006-1.

[60] Rawlings, J., Meadows, E., and Muske, K., "Nonlinear Model Predictive Control: A Tutorial and Survey," *IFAC Proceedings Volumes*, Vol. 27, No. 2, 1994, pp. 185–197. https://doi.org/10.1016/S1474-6670(17)48151-1.

[61] Rawlings, J., "Tutorial overview of model predictive control," *IEEE Control Systems Magazine*, Vol. 20, No. 3, 2000, pp. 38–52. https://doi.org/10.1109/37.845037.

[62] Allgöwer, F., and Zheng, A. (eds.), *Nonlinear Model Predictive Control*, Birkhäuser Basel, 2000. https://doi.org/10.1007/978-3-0348-8407-5.

[63] Zhang, X., Farina, M., Spinelli, S., and Scattolini, R., "Multi-rate model predictive control algorithm for systems with fast-slow dynamics," *IET Control Theory & Applications*, Vol. 12, No. 18, 2018, pp. 2468–2477. https://doi.org/10.1049/iet-cta.2018.5220.

[64] Garg, K., Cosner, R. K., Rosolia, U., Ames, A. D., and Panagou, D., "Multi-Rate Control Design Under Input Constraints via Fixed-Time Barrier Functions," *IEEE Control Systems Letters*, Vol. 6, 2022, pp. 608–613. https://doi.org/10.1109/LCSYS.2021.3084322.

[65] Weiss, A., Baldwin, M., Erwin, R. S., and Kolmanovsky, I., "Model Predictive Control for Spacecraft Rendezvous and Docking: Strategies for Handling Constraints and Case Studies," *IEEE Transactions on Control Systems Technology*, Vol. 23, No. 4, 2015, pp. 1638–1647. https://doi.org/10.1109/TCST.2014.2379639.

[66] Ames, A. D., Coogan, S., Egerstedt, M., Notomista, G., Sreenath, K., and Tabuada, P., "Control Barrier Functions: Theory and Applications," *2019 European Control Conference*, 2019, pp. 3420–3431. https://doi.org/10.23919/ECC.2019.8796030.

[67] Shaw Cortez, W., Tan, X., and Dimarogonas, D. V., "A Robust, Multiple Control Barrier Function Framework for Input Constrained Systems," *IEEE Control Systems Letters*, Vol. 6, 2022, pp. 1742–1747. https://doi.org/10.1109/LCSYS.2021.3133418.

[68] Luo, B., Wu, H.-N., Huang, T., and Liu, D., "Reinforcement learning solution for HJB equation arising in constrained optimal control problem," *Neural Networks*, Vol. 71, 2015, pp. 150–158. https://doi.org/10.1016/j.neunet.2015.08.007.

[69] Adib Yaghmaie, F., and Braun, D. J., "Reinforcement learning for a class of continuous-time input constrained optimal control problems," *Automatica*, Vol. 99, 2019, pp. 221–227. https://doi.org/10.1016/j.automatica.2018.10.038.

[70] Han, M., Tian, Y., Zhang, L., Wang, J., and Pan, W., "Reinforcement learning control of constrained dynamic systems with uniformly ultimate boundedness stability guarantee," *Automatica*, Vol. 129, 2021, p. 109689. https://doi.org/10.1016/j.automatica.2021.109689.

[71] Khalil, H. K., *Nonlinear Systems, Third Edition*, Prentice Hall, 2002.

[72] Prajna, S., Jadbabaie, A., and Pappas, G. J., "A Framework for Worst-Case and Stochastic Safety Verification Using Barrier Certificates," *IEEE Transactions on Automatic Control*, Vol. 52, No. 8, 2007, pp. 1415–1428. https://doi.org/10.1109/TAC.2007.902736.

[73] Lin, Y., and Sontag, E. D., "A universal formula for stabilization with bounded controls," *Systems & Control Letters*, Vol. 16, No. 6, 1991, pp. 393–397. https://doi.org/10.1016/0167-6911(91)90111-Q.

[74] Prajna, S., and Jadbabaie, A., "Safety Verification of Hybrid Systems Using Barrier Certificates," *Hybrid Systems: Computation and Control*, edited by R. Alur and G. J. Pappas, Springer Berlin Heidelberg, Berlin, Heidelberg, 2004, pp. 477–492. https://doi.org/10.1007/978-3-540-24743-2_32.

[75] Prajna, S., Jadbabaie, A., and Pappas, G., "Stochastic safety verification using barrier certificates," *2004 43rd IEEE Conference on Decision and Control*, 2004, pp. 929–934. https://doi.org/10.1109/CDC.2004.1428804.

[76] Prajna, S., and Rantzer, A., "On the Necessity of Barrier Certificates," *16th IFAC World Congress*, Vol. 38, 2005, pp. 526–531. https://doi.org/10.3182/20050703-6-CZ-1902.00743.

[77] Prajna, S., "Barrier certificates for nonlinear model validation," *Automatica*, Vol. 42, 2006, pp. 117–126. https://doi.org/10.1016/j.automatica.2005.08.007.

[78] Yang, G., Belta, C., and Tron, R., "Continuous-time Signal Temporal Logic Planning with Control Barrier Functions," *2020 American Control Conference*, 2020, pp. 4612–4618. https://doi.org/10.23919/ACC45564.2020.9147387.

[79] Garg, K., and Panagou, D., "Control-Lyapunov and Control-Barrier Functions based Quadratic Program for Spatio-temporal Specifications," *2019 IEEE 58th Conference on Decision and Control*, 2019, pp. 1422–1429. https://doi.org/10.1109/CDC40024.2019.9029666.

[80] Lindemann, L., and Dimarogonas, D. V., "Control Barrier Functions for Multi-Agent Systems Under Conflicting Local Signal Temporal Logic Tasks," *IEEE Control Systems Letters*, Vol. 3, No. 3, 2019, pp. 757–762. https://doi.org/10.1109/LCSYS.2019.2917975.

[81] Wang, L., Ames, A. D., and Egerstedt, M., "Safe certificate-based maneuvers for teams of quadrotors using differential flatness," *2017 IEEE International Conference on Robotics and Automation*, 2017, pp. 3293–3298. https://doi.org/10.1109/ICRA.2017.7989375.

[82] Doeser, L., Nilsson, P., Ames, A. D., and Murray, R. M., "Invariant Sets for Integrators and Quadrotor Obstacle Avoidance," *2020 American Control Conference*, 2020, pp. 3814–3821. https://doi.org/10.23919/ACC45564.2020.9147872.

[83] Wu, G., and Sreenath, K., "Safety-critical and constrained geometric control synthesis using control Lyapunov and control Barrier functions for systems evolving on manifolds," *2015 American Control Conference (ACC)*, 2015, pp. 2038–2044. https://doi.org/10.1109/ACC.2015.7171033.

[84] Hsu, S.-C., Xu, X., and Ames, A. D., "Control barrier function based quadratic programs with application to bipedal robotic walking," *2015 American Control Conference*, 2015, pp. 4542–4548. https://doi.org/10.1109/ACC.2015.7172044.

[85] Nguyen, Q., and Sreenath, K., "Exponential Control Barrier Functions for enforcing high relative-degree safety-critical constraints," *2016 American Control Conference*, 2016, pp. 322–328. https://doi.org/10.1109/ACC.2016.7524935.

[86] Squires, E., Pierpaoli, P., and Egerstedt, M., "Constructive Barrier Certificates with Applications to Fixed-Wing Aircraft Collision Avoidance," *2018 IEEE Conference on Control Technology and Applications*, 2018, pp. 1656–1661. https://doi.org/10.1109/CCTA.2018.8511342.

[87] Xiao, W., and Belta, C., "Control Barrier Functions for Systems with High Relative Degree," *2019 IEEE 58th Conference on Decision and Control*, 2019, pp. 474–479. https://doi.org/10.1109/CDC40024.2019.9029455.

[88] Xiao, W., and Belta, C., "High-Order Control Barrier Functions," *IEEE Transactions on Automatic Control*, Vol. 67, No. 7, 2022, pp. 3655–3662. https://doi.org/10.1109/TAC.2021.3105491.

[89] Tan, X., Cortez, W. S., and Dimarogonas, D. V., "High-Order Barrier Functions: Robustness, Safety, and Performance-Critical Control," *IEEE Transactions on Automatic Control*, Vol. 67, No. 6, 2022, pp. 3021–3028. https://doi.org/10.1109/TAC.2021.3089639.

[90] Xiao, W., Belta, C. A., and Cassandras, C. G., "Sufficient conditions for feasibility of optimal control problems using Control Barrier Functions," *Automatica*, Vol. 135, 2022, p. 109960. https://doi.org/10.1016/j.automatica.2021.109960.

[91] Xiao, W., Belta, C., and Cassandras, C. G., "Adaptive Control Barrier Functions," *IEEE Transactions on Automatic Control*, Vol. 67, No. 5, 2022, pp. 2267–2281. https://doi.org/10.1109/TAC.2021.3074895.

[92] Cortez, W. S., and Dimarogonas, D. V., "Correct-by-Design Control Barrier Functions for Euler-Lagrange Systems with Input Constraints," *2020 American Control Conference*, 2020, pp. 950–955. https://doi.org/10.23919/ACC45564.2020.9147367.

[93] Shaw Cortez, W., and Dimarogonas, D. V., "Safe-by-design control for Euler–Lagrange systems," *Automatica*, Vol. 146, 2022, p. 110620. https://doi.org/10.1016/j.automatica.2022.110620.

[94] Singletary, A., Nilsson, P., Gurriet, T., and Ames, A. D., "Online Active Safety for Robotic Manipulators," *2019 IEEE/RSJ Conference on Intelligent Robots and Systems*, 2019, pp. 173–178. https://doi.org/10.1109/IROS40897.2019.8968231.

[95] Chen, Y., Jankovic, M., Santillo, M., and Ames, A. D., "Backup Control Barrier Functions: Formulation and Comparative Study," *2021 IEEE 60th Conference on Decision and Control*, 2021, pp. 6835–6841. https://doi.org/10.1109/CDC45484.2021.9683111.

[96] Gurriet, T., Mote, M., Ames, A. D., and Feron, E., "An Online Approach to Active Set Invariance," *2018 IEEE Conference on Decision and Control*, 2018, pp. 3592–3599. https://doi.org/10.1109/CDC.2018.8619139.

[97] Chen, Y., Singletary, A., and Ames, A. D., "Guaranteed Obstacle Avoidance for Multi-Robot Operations With Limited Actuation: A Control Barrier Function Approach," *IEEE Control Systems Letters*, Vol. 5, No. 1, 2021, pp. 127–132. https://doi.org/10.1109/LCSYS.2020.3000748.

[98] Wiltz, A., Tan, X., and Dimarogonas, D. V., "Construction of Control Barrier Functions Using Predictions with Finite Horizon," *2023 IEEE 62nd Conference on Decision and Control*, 2023. https://doi.org/10.48550/arXiv.2305.05294v3.

[99] Wabersich, K. P., and Zeilinger, M. N., "Predictive control barrier functions: Enhanced safety mechanisms for learning-based control," *IEEE Transactions on Automatic Control*, 2022. https://doi.org/10.1109/TAC.2022.3175628.

[100] Gurriet, T., Mote, M., Singletary, A., Nilsson, P., Feron, E., and Ames, A. D., "A Scalable Safety Critical Control Framework for Nonlinear Systems," *IEEE Access*, Vol. 8, 2020, pp. 187249–187275. https://doi.org/10.1109/ACCESS.2020.3025248.

[101] Robey, A., Lindemann, L., Tu, S., and Matni, N., "Learning Robust Hybrid Control Barrier Functions for Uncertain Systems," *7th IFAC Conference on Analysis and Design of Hybrid Systems*, 2021. https://doi.org/10.1016/j.ifacol.2021.08.465.

[102] Choi, J., Castañeda, F., Tomlin, C. J., and Sreenath, K., "Reinforcement Learning for Safety-Critical Control under Model Uncertainty, using Control Lyapunov Functions and Control Barrier Functions," *arXiv*, 2020. https://doi.org/10.48550/arXiv.2004.07584, v2.

[103] Wang, C., Meng, Y., Li, Y., Smith, S. L., and Liu, J., "Learning Control Barrier Functions with High Relative Degree for Safety-Critical Control," *2021 European Control Conference*, 2021, pp. 1459–1464. https://doi.org/10.23919/ECC54610.2021.9655206.

[104] Taylor, A., Singletary, A., Yue, Y., and Ames, A., "Learning for Safety-Critical Control with Control Barrier Functions," *Proceedings of the 2nd Conference on Learning for Dynamics and Control*, Proceedings of Machine Learning Research, Vol. 120, edited by A. M. Bayen, A. Jadbabaie, G. Pappas, P. A. Parrilo, B. Recht, C. Tomlin, and M. Zeilinger, PMLR, 2020, pp. 708–717. URL https://proceedings.mlr.press/v120/taylor20a.html.

[105] Black, M., Arabi, E., and Panagou, D., "A Fixed-Time Stable Adaptation Law for Safety-Critical Control under Parametric Uncertainty," *2021 European Control Conference*, 2021, pp. 1328–1333. https://doi.org/10.23919/ECC54610.2021.9655080.

[106] Taylor, A. J., and Ames, A. D., "Adaptive Safety with Control Barrier Functions," *2020 American Control Conference*, 2020, pp. 1399–1405. https://doi.org/10.23919/ACC45564.2020.9147463.

[107] Cohen, M. H., and Belta, C., "High Order Robust Adaptive Control Barrier Functions and Exponentially Stabilizing Adaptive Control Lyapunov Functions," *2022 American Control Conference*, 2022, pp. 2233–2238. https://doi.org/10.23919/ACC53348.2022.9867633.

[108] Lopez, B. T., Slotine, J.-J. E., and How, J. P., "Robust Adaptive Control Barrier Functions: An Adaptive and Data-Driven Approach to Safety," *IEEE Control Systems Letters*, Vol. 5, No. 3, 2021, pp. 1031–1036. https://doi.org/10.1109/LCSYS.2020.3005923.

[109] Miller, J. K., Weeks, C. J., and Wood, L. J., "Orbit determination strategy and accuracy for a Comet Rendezvous mission," *Journal of Guidance, Control, and Dynamics*, Vol. 13, No. 5, 1990, pp. 775–784. https://doi.org/10.2514/3.25402.

[110] Schatten, K., "Solar activity and the solar cycle," *Advances in Space Research*, Vol. 32, No. 4, 2003, pp. 451–460. https://doi.org/10.1016/S0273-1177(03)00328-4.

[111] Cook, G., "Satellite drag coefficients," *Planetary and Space Science*, Vol. 13, No. 10, 1965, pp. 929–946. https://doi.org/10.1016/0032-0633(65)90150-9.

[112] Alan, A., Taylor, A. J., He, C. R., Orosz, G., and Ames, A. D., "Safe Controller Synthesis With Tunable Input-to-State Safe Control Barrier Functions," *IEEE Control Systems Letters*, Vol. 6, 2022, pp. 908–913. https://doi.org/10.1109/LCSYS.2021.3087443.

[113] Kolathaya, S., and Ames, A. D., "Input-to-State Safety With Control Barrier Functions," *IEEE Control Systems Letters*, Vol. 3, No. 1, 2019, pp. 108–113. https://doi.org/10.1109/LCSYS.2018.2853698.

[114] Alan, A., Taylor, A. J., He, C. R., Ames, A. D., and Orosz, G., "Control Barrier Functions and Input-to-State Safety With Application to Automated Vehicles," *IEEE Transactions on Control Systems Technology*, Vol. 31, No. 6, 2023, pp. 2744–2759. https://doi.org/10.1109/TCST.2023.3286090.

[115] Alan, A., Molnar, T. G., Ames, A. D., and Orosz, G., "Parameterized Barrier Functions to Guarantee Safety Under Uncertainty," *IEEE Control Systems Letters*, Vol. 7, 2023, pp. 2077–2082. https://doi.org/10.1109/LCSYS.2023.3285188.

[116] Xu, X., Tabuada, P., Grizzle, J. W., and Ames, A. D., "Robustness of Control Barrier Functions for Safety Critical Control," *Analysis and Design of Hybrid Systems*, Vol. 48, No. 27, 2015, pp. 54–61. https://doi.org/10.1016/j.ifacol.2015.11.152.

[117] Jankovic, M., "Robust control barrier functions for constrained stabilization of nonlinear systems," *Automatica*, Vol. 96, 2018, pp. 359–367. https://doi.org/10.1016/j.automatica.2018.07.004.

[118] Garg, K., and Panagou, D., "Robust Control Barrier and Control Lyapunov Functions with Fixed-Time Convergence Guarantees," *2021 American Control Conference*, 2021, pp. 2292–2297. https://doi.org/10.23919/ACC50511.2021.9482751.

[119] Shaw Cortez, W., Oetomo, D., Manzie, C., and Choong, P., "Control Barrier Functions for Mechanical Systems: Theory and Application to Robotic Grasping," *IEEE Transactions on Control Systems Technology*, Vol. 29, No. 2, 2021, pp. 530–545. https://doi.org/10.1109/TCST.2019.2952317.

[120] Wang, C., Bahreinian, M., and Tron, R., "Chance Constraint Robust Control with Control Barrier Functions," *2021 American Control Conference*, 2021, pp. 2315–2322. https://doi.org/10.23919/ACC50511.2021.9482973.

[121] Yaghoubi, S., Majd, K., Fainekos, G., Yamaguchi, T., Prokhorov, D., and Hoxha, B., "Risk-Bounded Control Using Stochastic Barrier Functions," *IEEE Control Systems Letters*, Vol. 5, No. 5, 2021, pp. 1831–1836. https://doi.org/10.1109/LCSYS.2020.3043287.

[122] Singletary, A., Ahmadi, M., and Ames, A. D., "Safe Control for Nonlinear Systems With Stochastic Uncertainty via Risk Control Barrier Functions," *IEEE Control Systems Letters*, Vol. 7, 2023, pp. 349–354. https://doi.org/10.1109/LCSYS.2022.3187458.

[123] Wang, C., Meng, Y., Smith, S. L., and Liu, J., "Safety-Critical Control of Stochastic Systems using Stochastic Control Barrier Functions," *2021 IEEE 60th Conference on Decision and Control*, 2021, pp. 5924–5931. https://doi.org/10.1109/CDC45484.2021.9683349.

[124] Usevitch, J., and Panagou, D., "Adversarial Resilience for Sampled-Data Systems using Control Barrier Function Methods," *2021 American Control Conference*, 2021, pp. 758–763. https://doi.org/10.23919/ACC50511.2021.9482659.

[125] Niu, L., Zhang, H., and Clark, A., "Safety-Critical Control Synthesis for Unknown Sampled-Data Systems via Control Barrier Functions," *2021 IEEE 60th Conference on Decision and Control*, 2021, pp. 6806–6813. https://doi.org/10.1109/CDC45484.2021.9683019.

[126] Cosner, R. K., Singletary, A. W., Taylor, A. J., Molnar, T. G., Bouman, K. L., and Ames, A. D., "Measurement-Robust Control Barrier Functions: Certainty in Safety with Uncertainty in State," *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2021, pp. 6286–6291. https://doi.org/10.1109/IROS51168.2021.9636584.

[127] Vahs, M., Pek, C., and Tumova, J., "Belief Control Barrier Functions for Risk-Aware Control," *IEEE Robotics and Automation Letters*, Vol. 8, No. 12, 2023, pp. 8565–8572. https://doi.org/10.1109/LRA.2023.3330662.

[128] Zeng, J., Zhang, B., and Sreenath, K., "Safety-Critical Model Predictive Control with Discrete-Time Control Barrier Function," *2021 American Control Conference*, 2021, pp. 3882–3889. https://doi.org/10.23919/ACC50511.2021.9483029.

[129] Khajenejad, M., Cavorsi, M., Niu, R., Shen, Q., and Yong, S. Z., "Tractable Compositions of Discrete-Time Control Barrier Functions with Application to Driving Safety Control," *2021 European Control Conference*, 2021, pp. 1303–1309. https://doi.org/10.23919/ECC54610.2021.9655012.

[130] Taylor, A. J., Dorobantu, V. D., Cosner, R. K., Yue, Y., and Ames, A. D., "Safety of Sampled-Data Systems with Control Barrier Functions via Approximate Discrete Time Models," *2022 IEEE 61st Conference on Decision and Control*, 2022, pp. 7127–7134. https://doi.org/10.1109/CDC51059.2022.9993226.

[131] Marley, M., Skjetne, R., and Teel, A. R., "Synergistic control barrier functions with application to obstacle avoidance for nonholonomic vehicles," *2021 American Control Conference*, 2021, pp. 243–249. https://doi.org/10.23919/ACC50511.2021.9482979.

[132] Maghenem, M., and Sanfelice, R. G., "Sufficient conditions for forward invariance and contractivity in hybrid inclusions using barrier functions," *Automatica*, Vol. 124, 2021, p. 109328. https://doi.org/10.1016/j.automatica.2020.109328.

[133] Aubin, J.-P., Lygeros, J., Quincampoix, M., Sastry, S., and Seube, N., "Impulse differential inclusions: a viability approach to hybrid systems," *IEEE Transactions on Automatic Control*, Vol. 47, No. 1, 2002, pp. 2–20. https://doi.org/10.1109/9.981719.

[134] Braun, P., and Zaccarian, L., "Augmented obstacle avoidance controller design for mobile robots," *IFAC-PapersOnLine*, Vol. 54, No. 5, 2021, pp. 157–162. https://doi.org/10.1016/j.ifacol.2021.08.491.

[135] Glotfelter, P., Buckley, I., and Egerstedt, M., "Hybrid Nonsmooth Barrier Functions With Applications to Provably Safe and Composable Collision Avoidance for Robotic Systems," *IEEE Robotics and Automation Letters*, Vol. 4, No. 2, 2019, pp. 1303–1310. https://doi.org/10.1109/LRA.2019.2895125.

[136] Rauscher, M., Kimmel, M., and Hirche, S., "Constrained robot control using control barrier functions," *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2016, pp. 279–285. https://doi.org/10.1109/IROS.2016.7759067.

[137] Aali, M., and Liu, J., "Multiple Control Barrier Functions: An Application to Reactive Obstacle Avoidance for a Multi-steering Tractor-trailer System," *2022 IEEE 61st Conference on Decision and Control*, 2022, pp. 6993–6998. https://doi.org/10.1109/CDC51059.2022.9992820.

[138] Xu, X., "Constrained control of input–output linearizable systems using control sharing barrier functions," *Automatica*, Vol. 87, 2018, pp. 195–201. https://doi.org/10.1016/j.automatica.2017.10.005.

[139] Black, M., and Panagou, D., "Adaptation for Validation of a Consolidated Control Barrier Function based Control Synthesis," *arXiv*, 2022. https://doi.org/10.48550/ARXIV.2209.08170, v1.

[140] Tan, X., and Dimarogonas, D. V., "Compatibility checking of multiple control barrier functions for input constrained systems," *2022 IEEE 61st Conference on Decision and Control*, 2022, pp. 939–944. https://doi.org/10.1109/CDC51059.2022.9993001.

[141] Rosolia, U., and Ames, A. D., "Multi-Rate Control Design Leveraging Control Barrier Functions and Model Predictive Control Policies," *IEEE Control Systems Letters*, Vol. 5, No. 3, 2021, pp. 1007–1012. https://doi.org/10.1109/LCSYS.2020.3008326.

[142] Cohen, M. H., and Belta, C., "Approximate Optimal Control for Safety-Critical Systems with Control Barrier Functions," *2020 IEEE 59th Conference on Decision and Control*, 2020, pp. 2062–2067. https://doi.org/10.1109/CDC42340.2020.9303896.

[143] Almubarak, H., Theodorou, E. A., and Sadegh, N., "HJB Based Optimal Safe Control using Control Barrier Functions," *2021 IEEE 60th Conference on Decision and Control*, 2021, pp. 6829–6834. https://doi.org/10.1109/CDC45484.2021.9683655.

[144] Krstic, M., "Inverse Optimal Safety Filters," *IEEE Transactions on Automatic Control*, Vol. 69, No. 1, 2024, pp. 16–31. https://doi.org/10.1109/TAC.2023.3278788.

[145] Majd, K., Yaghoubi, S., Yamaguchi, T., Hoxha, B., Prokhorov, D., and Fainekos, G., "Safe Navigation in Human Occupied Environments Using Sampling and Control Barrier Functions," *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2021, pp. 5794–5800. https://doi.org/10.1109/IROS51168.2021.9636406.

[146] Xiao, W., Cassandras, C. G., and Belta, C. A., "Bridging the gap between optimal trajectory planning and safety-critical control with applications to autonomous vehicles," *Automatica*, Vol. 129, 2021, p. 109592. https://doi.org/10.1016/j.automatica.2021.109592.

[147] Choi, J. J., Lee, D., Sreenath, K., Tomlin, C. J., and Herbert, S. L., "Robust Control Barrier–Value Functions for Safety-Critical Control," *2021 IEEE 60th Conference on Decision and Control*, 2021, pp. 6814–6821. https://doi.org/10.1109/CDC45484.2021.9683085.

[148] Grandia, R., Taylor, A. J., Ames, A. D., and Hutter, M., "Multi-Layered Safety for Legged Robots via Control Barrier Functions and Model Predictive Control," *2021 IEEE International Conference on Robotics and Automation*, 2021, pp. 8352–8358. https://doi.org/10.1109/ICRA48506.2021.9561510.

[149] Parwana, H., Mustafa, A., and Panagou, D., "Trust-based Rate-Tunable Control Barrier Functions for Non-Cooperative Multi-Agent Systems," *2022 IEEE 61st Conference on Decision and Control*, 2022, pp. 2222–2229. https://doi.org/10.1109/CDC51059.2022.9992744.

[150] Glotfelter, P., Cortés, J., and Egerstedt, M., "Nonsmooth Barrier Functions With Applications to Multi-Robot Systems," *IEEE Control Systems Letters*, Vol. 1, No. 2, 2017, pp. 310–315. https://doi.org/10.1109/LCSYS.2017.2710943.

[151] Glotfelter, P., Cortés, J., and Egerstedt, M., "Boolean Composability of Constraints and Control Synthesis for Multi-Robot Systems via Nonsmooth Control Barrier Functions," *2018 IEEE Conference on Control Technology and Applications*, 2018, pp. 897–902. https://doi.org/10.1109/CCTA.2018.8511471.

[152] Lindemann, L., and Dimarogonas, D. V., "Decentralized Control Barrier Functions for Coupled Multi-Agent Systems under Signal Temporal Logic Tasks," *2019 European Control Conference*, 2019, pp. 89–94. https://doi.org/10.23919/ECC.2019.8796109.

[153] Usevitch, J., Garg, K., and Panagou, D., "Strong Invariance Using Control Barrier Functions: A Clarke Tangent Cone Approach," *2020 IEEE 59th Conference on Decision and Control*, 2020, pp. 2044–2049. https://doi.org/10.1109/CDC42340.2020.9303873.

[154] Thirugnanam, A., Zeng, J., and Sreenath, K., "Duality-based Convex Optimization for Real-time Obstacle Avoidance between Polytopes with Control Barrier Functions," *2022 American Control Conference*, 2022, pp. 2239–2246. https://doi.org/10.23919/ACC53348.2022.9867246.

[155] Cortes, J., "Discontinuous dynamical systems," *IEEE Control Systems Magazine*, Vol. 28, No. 3, 2008, pp. 36–73. https://doi.org/10.1109/MCS.2008.919306.

[156] Robey, A., Hu, H., Lindemann, L., Zhang, H., Dimarogonas, D. V., Tu, S., and Matni, N., "Learning Control Barrier Functions from Expert Demonstrations," *2020 IEEE 59th Conference on Decision and Control*, 2020, pp. 3717–3724. https://doi.org/10.1109/CDC42340.2020.9303785.

[157] Jin, W., Wang, Z., Yang, Z., and Mou, S., "Neural Certificates for Safe Control Policies," *arXiv*, 2020. https://doi.org/10.48550/arXiv.2006.08465, v1.

[158] Liu, S., Liu, C., and Dolan, J., "Safe Control Under Input Limits with Neural Control Barrier Functions," *Proceedings of the 6th Conference on Robot Learning*, Proceedings of Machine Learning Research, Vol. 205, edited by K. Liu, D. Kulic, and J. Ichnowski, PMLR, 2023, pp. 1970–1980. URL https://proceedings.mlr.press/v205/liu23e.html.

[159] Tan, D. C. H., Acero, F., McCarthy, R., Kanoulas, D., and Li, Z., "Value Functions are Control Barrier Functions: Verification of Safe Policies using Control Theory," *arXiv*, 2023. https://doi.org/10.48550/arXiv.2306.04026, v4.

[160] Clark, A., "Verification and Synthesis of Control Barrier Functions," *2021 IEEE 60th Conference on Decision and Control*, 2021, pp. 6105–6112. https://doi.org/10.1109/CDC45484.2021.9683520.

[161] Zhao, W., He, T., Wei, T., Liu, S., and Liu, C., "Safety Index Synthesis via Sum-of-Squares Programming," *2023 American Control Conference*, 2023, pp. 732–737. https://doi.org/10.23919/ACC55779.2023.10156463.

[162] Schneeberger, M., Dörfler, F., and Mastellone, S., "SOS Construction of Compatible Control Lyapunov and Barrier Functions," *arXiv*, 2023. https://doi.org/10.48550/arXiv.2305.01222, v1.

[163] Zhang, H., Wu, J., Vorobeychik, Y., and Clark, A., "Exact Verification of ReLU Neural Control Barrier Functions," , 2023. https://doi.org/10.48550/arXiv.2310.09360, v1.

[164] Srinivasan, M., Coogan, S., and Egerstedt, M., "Control of Multi-Agent Systems with Finite Time Control Barrier Certificates and Temporal Logic," *2018 IEEE 57th Conference on Decision and Control*, 2018, pp. 1991–1996. https://doi.org/10.1109/CDC.2018.8619113.

[165] Zehfroosh, A., and Tanner, H. G., "Control Barrier Navigation Functions for STL Motion Planning," *2022 30th Mediterranean Conference on Control and Automation*, 2022, pp. 682–687. https://doi.org/10.1109/MED54222.2022.9837126.

[166] Charitidou, M., and Dimarogonas, D. V., "Receding Horizon Control With Online Barrier Function Design Under Signal Temporal Logic Specifications," *IEEE Transactions on Automatic Control*, Vol. 68, No. 6, 2023, pp. 3545–3556. https://doi.org/10.1109/TAC.2022.3195470.

[167] Yang, G., Belta, C., and Tron, R., "Continuous-time Signal Temporal Logic Planning with Control Barrier Functions," *2020 American Control Conference*, 2020, pp. 4612–4618. https://doi.org/10.23919/ACC45564.2020.9147387.

[168] Bhat, S. P., and Bernstein, D. S., "Finite-Time Stability of Continuous Autonomous Systems," *SIAM Journal on Control and Optimization*, Vol. 38, No. 3, 2000, pp. 751–766. https://doi.org/10.1137/S0363012997321358.

[169] Lindemann, L., and Dimarogonas, D. V., "Control Barrier Functions for Signal Temporal Logic Tasks," *IEEE Control Systems Letters*, Vol. 3, No. 1, 2019, pp. 96–101. https://doi.org/10.1109/LCSYS.2018.2853182.

[170] Nilsson, P., and Ames, A. D., "Barrier Functions: Bridging the Gap between Planning from Specifications and Safety-Critical Control," *2018 IEEE Conference on Decision and Control*, 2018, pp. 765–772. https://doi.org/10.1109/CDC.2018.8619142.

[171] Pati, T., Hwang, S., and Yong, S. Z., "Preview Control Barrier Functions for Linear Continuous-Time Systems with Previewable Disturbances," *2023 European Control Conference*, 2023. https://doi.org/10.23919/ECC57647.2023.10178355.

[172] Goldberg, A., Havelund, K., and McGann, C., "Runtime verification for autonomous spacecraft software," *2005 IEEE Aerospace Conference*, 2005, pp. 507–516. https://doi.org/10.1109/AERO.2005.1559341.

[173] Breeden, J., and Panagou, D., "High Relative Degree Control Barrier Functions Under Input Constraints," *2021 IEEE 60th Conference on Decision and Control*, 2021, pp. 6119–6124. https://doi.org/10.1109/CDC45484.2021.9683705.

[174] Breeden, J., and Panagou, D., "Robust Control Barrier Functions under high relative degree and input constraints for satellite trajectories," *Automatica*, Vol. 155, 2023, p. 111109. https://doi.org/10.1016/j.automatica.2023.111109.

[175] Breeden, J., and Panagou, D., "Guaranteed Safe Spacecraft Docking With Control Barrier Functions," *IEEE Control Systems Letters*, Vol. 6, 2022, pp. 2000–2005. https://doi.org/10.1109/LCSYS.2021.3136813.

[176] Breeden, J., Garg, K., and Panagou, D., "Control Barrier Functions in Sampled-Data Systems," *IEEE Control Systems Letters*, Vol. 6, 2022, pp. 367–372. https://doi.org/10.1109/LCSYS.2021.3076127.

[177] Breeden, J., and Panagou, D., "Autonomous Spacecraft Attitude Reorientation Using Robust Sampled-Data Control Barrier Functions," *Journal of Guidance, Control, and Dynamics*, Vol. 46, No. 10, 2023, pp. 1874–1891. https://doi.org/10.2514/1.G007456.

[178] Breeden, J., and Panagou, D., "Safety-Critical Control for Systems With Impulsive Actuators and Dwell Time Constraints," *IEEE Control Systems Letters*, Vol. 7, 2023, pp. 2119–2124. https://doi.org/10.1109/LCSYS.2023.3285141.

[179] Breeden, J., and Panagou, D., "Compositions of Multiple Control Barrier Functions Under Input Constraints," *2023 American Control Conference*, 2023, pp. 3688–3695. https://doi.org/10.23919/ACC55779.2023.10156625.

[180] Breeden, J., and Panagou, D., "Predictive Control Barrier Functions for Online Safety Critical Control," *2022 IEEE 61st Conference on Decision and Control*, 2022, pp. 924–931. https://doi.org/10.1109/CDC51059.2022.9992926.

[181] Garg, K., Usevitch, J., Breeden, J., Black, M., Agrawal, D., Parwana, H., and Panagou, D., "Advances in the Theory of Control Barrier Functions: Addressing Practical Challenges in Safe Control Synthesis for Autonomous and Robotic Systems," *arXiv*, 2023. https://doi.org/10.48550/arXiv.2312.16719, v1.

[182] Monnet, D., Ninin, J., and Jaulin, L., "Computing an inner and an outer approximation of the viability kernel," *Reliable Computing*, Vol. 22, 2016. URL https://hal.science/hal-01366752/file/Viability_Kernel_FINAL.pdf.

[183] Saint-Pierre, P., "Approximation of the viability kernel," *Applied Mathematics and Optimization*, Vol. 29, 1994, pp. 187–2019. https://doi.org/10.1007/BF01204182.

[184] Kaynama, S., Maidens, J., Oishi, M., Mitchell, I. M., and Dumont, G. A., "Computing the Viability Kernel Using Maximal Reachable Sets," *Proceedings of the 15th ACM International Conference on Hybrid Systems: Computation and Control*, Association for Computing Machinery, New York, NY, USA, 2012, p. 55–64. https://doi.org/10.1145/2185632.2185644.

[185] Martin, B., and Mullier, O., "Improving Validated Computation of Viability Kernels," *Proceedings of the 21st International Conference on Hybrid Systems: Computation and Control*, Association for Computing Machinery, New York, NY, USA, 2018, p. 227–236. https://doi.org/10.1145/3178126.3178141.

[186] Mitchell, I. M., Budzis, J., and Bolyachevets, A., "Invariant, Viability and Discriminating Kernel under-Approximation via Zonotope Scaling: Poster Abstract," *Proceedings of the 22nd ACM International Conference on Hybrid Systems: Computation and Control*, Association for Computing Machinery, New York, NY, USA, 2019, p. 268–269. https://doi.org/10.1145/3302504.3313354.

[187] Bouguerra, M. A., Fraichard, T., and Fezari, M., "Viability-Based Guaranteed Safe Robot Navigation," *Journal of Intelligent & Robotic Systems*, Vol. 95, 2019, pp. 459–471. https://doi.org/10.1007/s10846-018-0955-9.

[188] Rakovic, S. V., and Baric, M., "Parameterized Robust Control Invariant Sets for Linear Systems: Theoretical Advances and Computational Remarks," *IEEE Transactions on Automatic Control*, Vol. 55, No. 7, 2010, pp. 1599–1614. https://doi.org/10.1109/TAC.2010.2042341.

[189] Ames, A. D., Grizzle, J. W., and Tabuada, P., "Control barrier function based quadratic programs with application to adaptive cruise control," *53rd IEEE Conference on Decision and Control*, 2014, pp. 6271–6278. https://doi.org/10.1109/CDC.2014.7040372.

[190] Fitzpatrick, P. M., *Advanced Calculus*, 2nd ed., American Mathematical Society, 2006.

[191] Rouche, N., Habets, P., and Laloy, M., *Stability Theory by Liapunov's Direct Method*, Springer-Verlag, 1977. https://doi.org/10.1007/978-1-4684-9362-7, URL https://link.springer.com/book/10.1007/978-1-4684-9362-7.

[192] Nagumo, M., "Über die Lage der Integralkurven gewöhnlicher Differentialgleichungen," *Proceedings of the Physico-Mathematical Society of Japan*, Vol. 24, 1942, pp. 551–559. https://doi.org/10.11429/ppmsj1919.24.0_551.

[193] Bouligand, G., *Introducion a la geometrie inxnitesimale directe*, Libraire Vuibert, 1932.

[194] Blanchini, F., and Miani, S., *Set-Theoretic Methods in Control*, Birkhäuser Cham, 2015. https://doi.org/10.1007/978-3-319-17933-9.

[195] Blanchini, F., "Set invariance in control," *Automatica*, Vol. 35, No. 11, 1999, pp. 1747 – 1767. https://doi.org/10.1016/S0005-1098(99)00113-2.

[196] Stellato, B., Banjac, G., Goulart, P., Bemporad, A., and Boyd, S., "OSQP: an operator splitting solver for quadratic programs," *Mathematical Programming Computation*, Vol. 12, No. 4, 2020, pp. 637–672. https://doi.org/10.1007/s12532-020-00179-2.

[197] Mangasarian, O., and Fromovitz, S., "The Fritz John necessary optimality conditions in the presence of equality and inequality constraints," *Journal of Mathematical Analysis and Applications*, Vol. 17, 1967, pp. 37–47. https://doi.org/10.1016/0022-247X(67)90163-1.

[198] Gauvin, J., "A necessary and sufficient regularity condition to have bounded multipliers in nonconvex programming," *Mathematical Programming*, Vol. 12, 1977, pp. 136–138. https://doi.org/10.1007/BF01593777.

[199] Dempe, S., "Directional differentiability of optimal solutions under Slater's condition," *Mathematical Programming*, Vol. 59, 1993, pp. 49–69. https://doi.org/10.1007/BF01581237.

[200] Lee, G. M., Tam, N. N., and Yen, N. D., "Continuity of the Solution Map in Quadratic Programs under Linear Perturbations," *Journal of Optimization Theory and Applications*, Vol. 129, 2006, p. 415–423. https://doi.org/10.1007/s10957-006-9076-x.

[201] Morris, B. J., Powell, M. J., and Ames, A. D., "Continuity and smoothness properties of nonlinear optimization-based feedback controllers," *2015 IEEE 54th Conference on Decision and Control*, 2015, pp. 151–158. https://doi.org/10.1109/CDC.2015.7402101.

[202] Morris, B., Powell, M. J., and Ames, A. D., "Sufficient conditions for the Lipschitz continuity of QP-based multi-objective control of humanoid robots," *2013 IEEE 52nd Conference on Decision and Control*, 2013, pp. 2920–2926. https://doi.org/10.1109/CDC.2013.6760327.

[203] Breeden, J., and Panagou, D., "Quadratic Programs for High Relative Degree Spatial Constraints and Spatiotemporal Specifications with Spacecraft Applications," *2020 IEEE 59th Conference on Decision and Control*, 2020, pp. 1496–1502. https://doi.org/10.1109/CDC42340.2020.9304162.

[204] Wang, L., Ames, A. D., and Egerstedt, M., "Multi-objective compositions for collision-free connectivity maintenance in teams of mobile robots," *2016 IEEE 55th Conference on Decision and Control*, 2016, pp. 2659–2664. https://doi.org/10.1109/CDC.2016.7798663.

[205] Pshenichnyi, B. N., *Necessary Conditions for an Extremum*, Marcel Dekker, Inc., 1971.

[206] Sideris, T. C., *Ordinary Differential Equations and Dynamical Systems*, Atlantis Press, 2013.

[207] Healy, L. M., "Space Flight Dynamics and Navigation," , August 2018. Draft: August 26, 2018.

[208] Thomas, P. C., and Carcich, B. T., "MSI EROS SHAPE MODEL," Shape model of eros derived from msi data, Cornell University Center for Radiophysics and Space Research, June 2011. URL https://sbnarchive.psi.edu/pds3/near/NEAR_A_MSI_5_EROS_SHAPE_MODELS_V1_0/data/eros_2001012_007790.tab, eros_2001012_007790.tab.

[209] Torrence, M., "NLR393V3 GRAV POTENTIAL COEFFICIENTS," Gravity model, Goddard Space Flight Center, August 2001. URL https://sbnarchive.psi.edu/pds3/near/NEAR_A_5_COLLECTED_MODELS_V1_0/data/rss/n393coeff.tab, rss/n393coeff.tab.

[210] Azimi, V., and Hutchinson, S., "Exponential Control Lyapunov-Barrier Function Using a Filtering-Based Concurrent Learning Adaptive Approach," *IEEE Transactions on Automatic Control*, Vol. 67, No. 10, 2022, pp. 5376–5383. https://doi.org/10.1109/TAC.2021.3120622.

[211] Chinelato, C. I. G., and Angélico, B. A., "Robust Exponential Control Barrier Functions for Safety-Critical Control," *2021 American Control Conference*, 2021, pp. 2342–2347. https://doi.org/10.23919/ACC50511.2021.9482899.

[212] Sontag, E. D., *Mathematical Control Theory*, Springer Science+Business Media, 1998. https://doi.org/10.1007/978-1-4612-0577-7.

[213] Luenberger, D. G., and Ye, Y., *Linear and Nonlinear Programming*, Springer International Publishing Switzerland, 2016. https://doi.org/10.1007/978-3-319-18842-3.

[214] Yeomans, D., Giorgini, J. D., Konopliv, A., and Barriot, J.-P., "EROS POLE, SPIN AND LANDMARK DATA," Landmark data, Jet Propulsion Laboratory, July 2001. URL https://sbnarchive.psi.edu/pds3/near/NEAR_A_5_COLLECTED_MODELS_V1_0/data/rss/landmark.tab.

[215] Dong, H., Hu, Q., and Akella, M. R., "Dual-Quaternion-Based Spacecraft Autonomous Rendezvous and Docking Under Six-Degree-of-Freedom Motion Constraints," *Journal of Guidance, Control, and Dynamics*, Vol. 41, No. 5, 2018, pp. 1150–1162. https://doi.org/10.2514/1.G003094.

[216] Zappulla, R., Park, H., Virgili-Llop, J., and Romano, M., "Real-Time Autonomous Spacecraft Proximity Maneuvers and Docking Using an Adaptive Artificial Potential Field Approach," *IEEE Transactions on Control Systems Technology*, Vol. 27, No. 6, 2019, pp. 2598–2605. https://doi.org/10.1109/TCST.2018.2866963.

[217] Ventura, J., Ciarcià, M., Romano, M., and Walter, U., "Fast and Near-Optimal Guidance for Docking to Uncontrolled Spacecraft," *Journal of Guidance, Control, and Dynamics*, Vol. 40, No. 12, 2017, pp. 3138–3154. https://doi.org/10.2514/1.G001843.

[218] Lee, D., and Vukovich, G., "Robust adaptive terminal sliding mode control on SE(3) for autonomous spacecraft rendezvous and docking," *Nonlinear Dynamics*, Vol. 83, 2016, pp. 2263–2279. https://doi.org/10.1007/s11071-015-2479-1.

[219] Oestreich, C. E., Linares, R., and Gondhalekar, R., "Autonomous Six-Degree-of-Freedom Spacecraft Docking Maneuvers via Reinforcement Learning," *arXiv*, 2020. https://doi.org/10.48550/arXiv.2008.03215, v1.

[220] Sun, L., and Jiang, J., "Saturated Adaptive Relative Motion Coordination of Docking Ports in Space Close-Range Rendezvous," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 56, No. 6, 2020, pp. 4889–4898. https://doi.org/10.1109/TAES.2020.3003961.

[221] Kim, Y., and Mesbahi, M., "Quadratically constrained attitude control via semidefinite programming," *IEEE Transactions on Automatic Control*, Vol. 49, No. 5, 2004, pp. 731–735. https://doi.org/10.1109/TAC.2004.825959.

[222] Kim, Y., Mesbahi, M., Singh, G., and Hadaegh, F. Y., "On the Convex Parameterization of Constrained Spacecraft Reorientation," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 46, No. 3, 2010, pp. 1097–1109. https://doi.org/10.1109/TAES.2010.5545176.

[223] Weiss, A., Leve, F., Baldwin, M., Forbes, J. R., and Kolmanovsky, I., "Spacecraft constrained attitude control using positively invariant constraint admissible sets on SO(3) x R3," *2014 American Control Conference*, 2014, pp. 4955–4960. https://doi.org/10.1109/ACC.2014.6858878.

[224] Tan, X., Berkane, S., and Dimarogonas, D. V., "Constrained attitude maneuvers on SO(3): Rotation space sampling, planning and low-level control," *Automatica*, Vol. 112, 2020, p. 108659. https://doi.org/10.1016/j.automatica.2019.108659.

[225] Reynolds, T. P., Szmuk, M., Malyuta, D., Mesbahi, M., Açıkmeşe, B., and Carson, J. M., "Dual Quaternion-Based Powered Descent Guidance with State-Triggered Constraints," *Journal of Guidance, Control, and Dynamics*, Vol. 43, No. 9, 2020, pp. 1584–1599. https://doi.org/10.2514/1.G004536.

[226] Sun, C., and Dai, R., "Spacecraft Attitude Control under Constrained Zones via Quadratically Constrained Quadratic Programming," *AIAA Guidance, Navigation, and Control Conference*, 2015. https://doi.org/10.2514/6.2015-2010.

[227] Calaon, R., and Schaub, H., "Constrained Attitude Maneuvering via Modified-Rodrigues-Parameter-Based Motion Planning Algorithms," *Journal of Spacecraft and Rockets*, Vol. 59, No. 4, 2022, pp. 1342–1356. https://doi.org/10.2514/1.A35294.

[228] Kjellberg, H. C., and Lightsey, E. G., "Discretized Constrained Attitude Pathfinding and Control for Satellites," *Journal of Guidance, Control, and Dynamics*, Vol. 36, No. 5, 2013, pp. 1301–1309. https://doi.org/10.2514/1.60189.

[229] Tanygin, S., "Fast Autonomous Three-Axis Constrained Attitude Pathfinding and Visualization for Boresight Alignment," *Journal of Guidance, Control, and Dynamics*, Vol. 40, No. 2, 2017, pp. 358–370. https://doi.org/10.2514/1.G001801.

[230] Feron, E., Dahleh, M., Frazzoli, E., and Kornfeld, R., "A randomized attitude slew planning algorithm for autonomous spacecraft," *AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2001. https://doi.org/10.2514/6.2001-4155.

[231] Gupta, R., Kalabić, U. V., Di Cairano, S., Bloch, A. M., and Kolmanovsky, I. V., "Constrained spacecraft attitude control on SO(3) using fast nonlinear model predictive control," *2015 American Control Conference*, 2015, pp. 2980–2986. https://doi.org/10.1109/ACC.2015.7171188.

[232] Lee, D. Y., Gupta, R., Kalabić, U. V., Di Cairano, S., Bloch, A. M., Cutler, J. W., and Kolmanovsky, I. V., "Constrained attitude maneuvering of a spacecraft with reaction wheel assembly by Nonlinear Model Predictive Control," *2016 American Control Conference*, 2016, pp. 4960–4965. https://doi.org/10.1109/ACC.2016.7526139.

[233] Lee, D. Y., Gupta, R., Kalabić, U. V., Di Cairano, S., Bloch, A. M., Cutler, J. W., and Kolmanovsky, I. V., "Geometric Mechanics Based Nonlinear Model Predictive Spacecraft Attitude Control with Reaction Wheels," *Journal of Guidance, Control, and Dynamics*, Vol. 40, No. 2, 2017, pp. 309–319. https://doi.org/10.2514/1.G001923.

[234] Lourenço, P., Franco, J., Milhano, T., and Branco, J., "Model Predictive Control for On-Board-Optimized Attitude Guidance for Cometary Fly-By-Problem Statement, Solutions and V&V Process," *Proceedings of the 8th International Conference on Astrodynamics Tools and Techniques*, 2021. https://doi.org/10.5281/zenodo.6411680.

[235] Yang, J., Duan, Y., Ben-Larbi, M. K., and Stoll, E., "Potential Field-Based Sliding Surface Design and Its Application in Spacecraft Constrained Reorientation," *Journal of Guidance, Control, and Dynamics*, Vol. 44, No. 2, 2021, pp. 399–409. https://doi.org/10.2514/1.G005026.

[236] Shen, Q., Yue, C., Goh, C. H., Wu, B., and Wang, D., "Rigid-body attitude stabilization with attitude and angular rate constraints," *Automatica*, Vol. 90, 2018, pp. 157–163. https://doi.org/10.1016/j.automatica.2017.12.029.

[237] Kang, Z., Shen, Q., and Wu, S., "Constrained Attitude Control of Over-Actuated Spacecraft Subject to Instrument Pointing Direction Deviation," *IEEE Control Systems Letters*, Vol. 5, No. 6, 2021, pp. 1958–1963. https://doi.org/10.1109/LCSYS.2020.3044984.

[238] Lee, U., and Mesbahi, M., "Feedback control for spacecraft reorientation under attitude constraints via convex potentials," *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 50, No. 4, 2014, pp. 2578–2592. https://doi.org/10.1109/TAES.2014.120240.

[239] Dong, H., Zhao, X., and Yang, H., "Reinforcement Learning-Based Approximate Optimal Control for Attitude Reorientation Under State Constraints," *IEEE Transactions on Control Systems Technology*, Vol. 29, No. 4, 2021, pp. 1664–1673. https://doi.org/10.1109/TCST.2020.3007401.

[240] Diaz Ramos, M., and Schaub, H., "Kinematic Steering Law for Conically Constrained Torque-Limited Spacecraft Attitude Control," *Journal of Guidance, Control, and Dynamics*, Vol. 41, No. 9, 2018, pp. 1990–2001. https://doi.org/10.2514/1.G002873.

[241] Ibuki, T., Wilson, S., Ames, A. D., and Egerstedt, M., "Distributed Collision-Free Motion Coordination on a Sphere: A Conic Control Barrier Function Approach," *IEEE Control Systems Letters*, Vol. 4, No. 4, 2020, pp. 976–981. https://doi.org/10.1109/LCSYS.2020.2997952.

[242] Karpenko, M., Bhatt, S., Bedrossian, N., and Ross, I. M., "Flight Implementation of Shortest-Time Maneuvers for Imaging Satellites," *Journal of Guidance, Control, and Dynamics*, Vol. 37, No. 4, 2014, pp. 1069–1079. https://doi.org/10.2514/1.62867.

[243] Stoneking, E., "42: An Open-Source Simulation Tool For Study And Design Of Spacecraft Attitude Control Systems," Tech. Rep. GSFC-E-DAA-TN52069, NASA Goddard Space Flight Center, 2018. URL https://ntrs.nasa.gov/citations/20180000954.

[244] Yang, G., Belta, C., and Tron, R., "Self-triggered Control for Safety Critical Systems Using Control Barrier Functions," *2019 American Control Conference*, 2019, pp. 4454–4459. https://doi.org/10.23919/ACC.2019.8814657.

[245] Heemels, W., Johansson, K., and Tabuada, P., "An introduction to event-triggered and self-triggered control," *2012 IEEE 51st Conference on Decision and Control*, 2012, pp. 3270–3285. https://doi.org/10.1109/CDC.2012.6425820.

[246] Long, L., and Wang, J., "Safety-critical dynamic event-triggered control of nonlinear systems," *Syst. & Contr. Letters*, Vol. 162, 2022, p. 105176. https://doi.org/10.1016/j.sysconle.2022.105176.

[247] Hespanha, J. P., Liberzon, D., and Teel, A. R., "Lyapunov conditions for input-to-state stability of impulsive systems," *Automatica*, Vol. 44, No. 11, 2008, pp. 2735–2744. https://doi.org/10.1016/j.automatica.2008.03.021.

[248] Azhmyakov, V., Boltyanski, V., and Poznyak, A., "Optimal control of impulsive hybrid systems," *Nonlinear Analysis: Hybrid Systems*, Vol. 2, No. 4, 2008, pp. 1089–1097. https://doi.org/10.1016/j.nahs.2008.09.003.

[249] Wang, H., Zhang, H., Wang, Z., and Chen, Q., "Impulsive control and stability analysis of biped robot based on virtual constraint and adaptive optimization," *Advanced Control for Applications*, Vol. 2, No. 2, 2020. https://doi.org//10.1002/adc2.32.

[250] Li, X., Peng, D., and Cao, J., "Lyapunov Stability for Impulsive Systems via Event-Triggered Impulsive Control," *IEEE Transactions on Automatic Control*, Vol. 65, No. 11, 2020, pp. 4908–4913. https://doi.org/10.1109/TAC.2020.2964558.

[251] Li, X., Zhang, T., and Wu, J., "Input-to-State Stability of Impulsive Systems via Event-Triggered Impulsive Control," *IEEE Transactions on Cybernetics*, Vol. 52, No. 7, 2022, pp. 7187–7195. https://doi.org/10.1109/TCYB.2020.3044003.

[252] Tan, X., Cao, J., and Li, X., "Consensus of Leader-Following Multiagent Systems: A Distributed Event-Triggered Impulsive Control Strategy," *IEEE Transactions on Cybernetics*, Vol. 49, No. 3, 2019, pp. 792–801. https://doi.org/10.1109/TCYB.2017.2786474.

[253] Dong, H., and Akella, M. R., "Autonomous rendezvous and docking of spacecraft under 6-DOF motion constraints," *2017 IEEE 56th Annual Conference on Decision and Control*, 2017, pp. 4527–4532. https://doi.org/10.1109/CDC.2017.8264327.

[254] Alur, D., Rajeev nd Dill, "The theory of timed automata," *Real-Time: Theory in Practice*, Springer Berlin Heidelberg, 1991, pp. 45–73. https://doi.org/10.1007/BFb00319.

[255] Grizzle, J., and Westervelt, E., "Hybrid Zero Dynamics of Planar Bipedal Walking," *Analysis and Design of Nonlinear Control Systems*, Springer Berlin Heidelberg, 2008, pp. 223–237. https://doi.org/10.1007/978-3-540-74358-3_14.

[256] Langson, W., Chryssochoos, I., Raković, S., and Mayne, D., "Robust model predictive control using tubes," *Automatica*, Vol. 40, No. 1, 2004, pp. 125–133. https://doi.org/10.1016/j.automatica.2003.08.009.

[257] Zhang, Y., Li, Y., Tomsovic, K., Djouadi, S. M., and Yue, M., "Review on set-theoretic methods for safety verification and control of power system," *IET Energy Systems Integration*, Vol. 2, No. 3, 2020, pp. 226–234. https://doi.org/10.1049/iet-esi.2019.0133.

[258] Gruber, F., and Althoff, M., "Computing Safe Sets of Linear Sampled-Data Systems," *IEEE Control Systems Letters*, Vol. 5, No. 2, 2021, pp. 385–390. https://doi.org/10.1109/LCSYS.2020.3002476.

[259] Athanasopoulos, N., Bitsoris, G., and Lazar, M., "Construction of invariant polytopic sets with specified complexity," *International Journal of Control*, Vol. 87, No. 8, 2014, pp. 1681–1693. https://doi.org/10.1080/00207179.2014.882518.

[260] Wang, L., Han, D., and Egerstedt, M., "Permissive Barrier Certificates for Safe Stabilization Using Sum-of-squares," *2018 American Control Conference*, 2018, pp. 585–590. https://doi.org/10.23919/ACC.2018.8431617.

[261] Garg, K., Cardenas, A. A., and Sanfelice, R. G., "Sampling-based Computation of Viability Domain to Prevent Safety Violations by Attackers," *2022 IEEE Conference on Control Technology and Applications*, 2022, pp. 720–725. https://doi.org/10.1109/CCTA49430.2022.9966120.

[262] Zeng, J., Zhang, B., Li, Z., and Sreenath, K., "Safety-Critical Control using Optimal-decay Control Barrier Function with Guaranteed Point-wise Feasibility," *2021 American Control Conference*, 2021, pp. 3856–3863. https://doi.org/10.23919/ACC50511.2021.9482626.

[263] Singletary, A., Kolathaya, S., and Ames, A. D., "Safety-Critical Kinematic Control of Robotic Systems," *IEEE Control Systems Letters*, Vol. 6, 2022, pp. 139–144. https://doi.org/10.1109/LCSYS.2021.3050609.

[264] Black, M., Jankovic, M., Sharma, A., and Panagou, D., "Future-Focused Control Barrier Functions for Autonomous Vehicle Control," *2023 American Control Conference*, 2023, pp. 3324–3331. https://doi.org/10.23919/ACC55779.2023.10156163.

[265] Kim, H., Yoon, H., Wan, W., Hovakimyan, N., Sha, L., and Voulgaris, P., "Backup Plan Constrained Model Predictive Control," *2021 IEEE 60th Conference on Decision and Control*, 2021, pp. 289–294. https://doi.org/10.1109/CDC45484.2021.9683388.

[266] Wu, Z., Albalawi, F., Zhang, Z., Zhang, J., Durand, H., and Christofides, P. D., "Control Lyapunov-Barrier function-based model predictive control of nonlinear systems," *Automatica*, Vol. 109, 2019. https://doi.org/10.1016/j.automatica.2019.108508.

[267] Xu, K., Xiao, W., and Cassandras, C. G., "Feasibility Guaranteed Traffic Merging Control Using Control Barrier Functions," *2022 American Control Conference*, 2022, pp. 2309–2314. https://doi.org/10.23919/ACC53348.2022.9867620.

[268] Marvi, Z., and Kiumarsi, B., "Safe Off-policy Reinforcement Learning Using Barrier Functions," *2020 American Control Conference*, 2020, pp. 2176–2181. https://doi.org/10.23919/ACC45564.2020.9147584.