Convex and Distributed Safety Analysis and Design for Control Systems

Han Wang

Linacre College

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

Trinity Term, 2024

This thesis is dedicated to
韩志芳
The best teacher, friend, and mom

我深切地感到
尽管创造的过程无比艰辛
而成功的结果无比荣耀
尽管一切艰辛都是为了成功
但是，人生最大的幸福
也许在于创造的过程
而不是那个结果

——路遥

Acknowledgements

First and foremost, I would like to express my heartfelt gratitude to my supervisors, Prof. Kostas Margellos and Prof. Antonis Papachristodoulou. Over the past four years, their guidance and support have transformed me from someone filled with academic curiosity but little knowledge, into someone now at the beginning of an academic career. Beyond the academic matters, they have also been a tremendous help in many aspects of my personal life. In 2021, when I was in lockdown in Shanghai due to the COVID-19 pandemic, I experienced emotional instability and frequent anxiety about my academic progress. During that time, Antonis and Kostas often took on the role of counsellors, helping to ease my concerns and guiding me through a challenging period. Without their support, I would not have been able to complete this DPhil successfully.

Except for my supervisors, I would also like to thank Professor Claudio De Persis from the University of Groningen, Professor Francesco Bullo from UCSB, and Professor Jianping He from Shanghai Jiao Tong University. Chapters 3 and 4 of this thesis, in particular, have greatly benefited from Claudio's guidance. I was incredibly fortunate to visit Claudio's research group at the University of Groningen and to collaborate with him on several academic papers. Claudio has always been exceptionally patient, offering detailed feedback on my papers. I have learned a lot from him, especially his rigorous academic approach and supportive attitude. I would also like to express my gratitude to my undergraduate supervisors, Jianping and Francesco. Without their recommendations, I would not have had the opportunity to pursue my DPhil at the University of Oxford.

Before my DPhil journey, stories about negative competition inside research groups worried me a lot, but such stories never happen in the supportive Oxford Control Group. I would like to extend a special thank you to Yana Lishkova and Yuwen Chen who are always the first to give a helping hand. I will fondly remember our enjoyable lunches, bars nights, and memorable parties. My thanks also go to Zuxun Xiong, Jingyi Huang, Allan Andre Do Nascimento, Andre Bertolace, Liqun Zhao, Idris Kempf, Manos Alexis, Matthew Newton, Martin Doff-Sotta, and many others with whom I shared countless unforgettable moments in Oxford, Cancun, and Singapore.

My time at Oxford would not have been as memorable without the cherished friends I've made. I would like to thank Keyan Miao, Yufan Wang, Haopeng Xu, Jinyi Zhao, and Ziwei Ye. The time spent with Keyan at the gym, the road trip with Haopeng and Jinyi through the Scottish Highlands, Yufan's late-night musings, and Ziwei's delicious cooking are among my most treasured memories.

I also want to acknowledge my friends outside Oxford, in particular, Qi Cao, Yukun Zhang, Jiaqi Gu, Menglei Zhang, Mingyu Lu, Haoyu Xing, Qiuyu Zhao, Xuda Ding, and Pangkit Fong. They have been my closest friends from primary school through to university, and their friendship is one of the most valuable treasures of my life.

Most importantly, I would like to thank Ruyue Wang and Yi Zhen (Satur) Chong. They are the ones who understand me best, bringing me the greatest joy and helping to lift the most of my negative emotions. The wonderful memories I have with them are so numerous that I could write an entire book to list them all.

Finally, I would also like to send my deepest appreciation to all my family: my father, Xuewen Wang; mother, Zhifang Han; aunt, Zhiping Han; uncle, Mingshan Jiao; grandmother, Aihua Liang; grandfather, Fulong Wang; sister, Yubing Han; brother-in-law, Daocheng Min; niece, Yuhan Min; aunt, Xiulian Qiu; and uncle, Xinyu Wang. Their selfless love and support have given me the power to keep moving forward.

Lastly, I want to thank myself.

Abstract

Our future will increasingly be filled with intelligent autonomous systems such as autonomous cars, robots, and power devices. As these systems are deployed in the real world, especially within human reach, there is an increasing need to provide safety guarantees, such as collision avoidance for moving robots and current/voltage protection for power devices. However, verifying and designing safe control systems is challenging due to the inherent nonconvexity in optimization and poor scalability with the system dimension. This thesis addresses these challenges from different aspects, using advanced convex optimization techniques.

A prevalent technique for designing safe control systems is the use of *control barrier functions* (CBFs). Analogous to control Lyapunov functions (CLFs), which certify stability, CBFs are certificate functions for safety. However, co-designing a candidate CBF and a feedback controller is NP-hard, even for linear systems. This thesis addresses this difficulty by proposing novel convex co-design programs. We parameterize the CBF and feedback controller with certain functional bases to obtain a convex reformulation of the nominal nonconvex problem. The proposed convex program co-designs a CBF and a feedback controller efficiently. We further investigate more complex settings such as systems with nonlinear dynamics, input constraints, mixed-relative degrees, and model uncertainty. Convex conditions are proposed for each setting and we demonstrate great flexibility in being incorporated into the proposed convex program.

Another challenge arises from large-scale multi-agent systems (MASs) which have linearly growing dimensionality of state-space models with the number of agents. We propose a novel distributed control design algorithm for parallel computation with a guaranteed sublinear convergence rate. To facilitate computational implementation, we propose terminating the algorithm before convergence, providing probabilistic results for safety guarantees.

We then consider simultaneously ensuring safety and stability for control systems. This problem is arduous due to the potential conflict between a CBF and a CLF in the state space. We address this difficulty by co-designing these certificate functions that satisfy the relaxed compatibility conditions, and proposing a new control design framework. The designed controller guarantees safety, local stability, and is locally Lipschitz continuous.

Contents

Notation

Sets

$\mathbb{R}$	the set of real numbers
$\mathbb{R}_+$ ($\mathbb{R}_{++}$)	the nonnegative (positive) real numbers
$\mathbb{R}^n$	the set of real n dimensional vectors
$\Sigma[x]$	the set of sum-of-squares polynomials
$\Sigma[x]^{n\times n}$	the set of $n \times n$ dimensional sum-of-squares matrices
$\mathbb{R}[x]$	the set of polynomials
$\mathbb{R}[x]^{n\times m}$	the set of $n \times m$ polynomial matrices
$\mathrm{int}(\mathcal{C})$	the set of interior points of set $\mathcal{C}$
$\mathrm{cl}(\mathcal{C})$	the closure of set C
$\mathcal{C}^c$	the closure of the complement of set $\mathcal{C}$
$\mathcal{E}(x,\delta)$	the ball with radius δ centered at x

Operators

$\deg(p(x))$	the degree of polynomial $p(x)$
$X \succ 0$	matrix X is positive definite
$X \succeq 0$	matrix X is positive semi-definite

Acronyms

LMI	Linear Matrix Inequality
SDP	Semi-definite Program
SOSP	Sum-of-Squares Program
Psatz	Positivestellensatz

CBF . . . . Control Barrier Function

CLF . . . . Control Lyapunov Function

s.t. . . . . Subject to

QP . . . . . Quadratic Program

MAS . . . . Multi-agent System

1 Introduction

1.1 Literature Review and Motivation

Since James Watt's breakthrough improvement of the steam engine, humanity entered the industrial era [1]. In the past centuries, numerous machines powered by steam, electricity, or nuclear energy have been invented and actively applied to daily lives. The central technology for operating these machines is *control theory*. Take the steam engine as an example; James Watt designed a proportional controller to automatically apply more steam to the steam engine when its speed dropped too low and reduce the steam when the speed increased too high [2]. The main objective of control theory is to develop controllers governing the system inputs to robustly drive the system to a desired state. Over the past decades, numerous different methods have been proposed and applied in practice, such as PID [3], frequency domain methods [4, 5], and time domain methods [6–8]. These methods primarily aim at ensuring a level of *stability* under the presence of uncertainty [9–12], time-delay [13–15], and input saturation [16]. Examples of their applications include, but are not limited to, robotics, power systems, autonomous cars, and synthetic biology [17–23]. However, as the applications become increasingly sophisticated

and interact more with the real world, other critical control objectives, such as *safety*, begin to attract attention.

In a word, safety for dynamical systems refers to the property that any trajectory starting from a set of initial states remains within a set of safe states [24, Chapter 4]. For example, in autonomous vehicles, collisions with pedestrians must be avoided, and the velocity must be constrained below a certain threshold. Safety is a critical property for many applications, such as spacecraft rendezvous [25], robot collision avoidance [26], and battery overheating protection [27]. In the literature, systems that strongly demand safety assurance are also termed *safety-critical systems*. Analyzing and designing for safety is not a new topic in the control community, though it is sometimes discussed under different keywords. For example, in viability theory, safety constraints are usually called viability constraints [28]. In model predictive control and reachability theory, safety constraints are typically referred to as state or output constraints. We provide a review of three representative classes of methodologies, including: 1) reachability and viability theory; 2) model predictive control and predictive safety filters; 3) barrier certificates and CBFs, and demonstrate their relationships and differences.

Reachability and Viability Analysis

Reachability analysis aims at synthesizing the set of states that are backward (or forward) reachable for a dynamical system from a given set of initial states [29, 30]. For control systems, the reachable set (tube) and the optimal safe control input can be co-designed by solving a reach-avoid optimal control problem [31, 32]. Depending on whether the objective is to *reach exactly at time T* or *reach within a duration of time T*, the synthesized set is called either the *reachable set* or the *reachable tube*. A similar problem has also been investigated in viability theory [28], where the set of states from which there exists at least one sequence of controls to keep the system within the safety constraints is called a *viability kernel*. The duality between the viability kernel and the reachable tube has been shown in [33], indicating that they do not need to be treated separately in terms of analysis and

synthesis. Intuitively, consider the complementary set of the safe set, termed the unsafe set, as the set to reach. Then the reachable tube is the set from which the system will enter the unsafe set within a duration of time. The viability kernel is then directly the complementary set of the reachable tube.

Reachability analysis provides an efficient way to analyze safety and synthesize a safe controller for a dynamical system. This methodology has been applied to many safety-critical applications, especially in robotics [34–36]. Solving the reachability problem directly involves computing the solution of a Hamilton-Jacobi partial differential equation (HJ-PDE). Algorithms for numerically solving this problem have been proposed through viscosity solutions [37] and level-set methods [38]. However, it is well-known that numerically solving the HJ-PDEs suffers from the curse of dimensionality. As pointed out in [39], even with modern computational techniques such as state decomposition [40] and warm starting [41], computing the safe optimal controller can take more than 200 seconds for the 10-dimensional quadrotor dynamics. This nature makes it challenging for online computation, particularly for systems affected by unpredictable disturbances or when models must be validated or identified using online data.

Model Predictive Control and Predictive Safety Filter

To address the difficulty of solving the reachability optimal control problem and enhance robustness against unpredictable disturbances and model uncertainty, model predictive control (MPC) based methods have been proposed for the design of safe controllers [8, 42, 43]. The safety requirements can be incorporated into the optimization problems as sequential state constraints along the trajectory [44–46]. Compared with the reachability optimal control problem, MPC is usually easier to solve and scales better with the dimension of the state space. With a modern optimization solver such as Mosek [47], MPC problems with 1000 decision variables can be solved in several milliseconds. MPC has demonstrated great power in many real-time safety-critical applications, such as human-robot interaction [48], legged robots [49], surgical robots [50], and autonomous driving [51]. There have been

tremendous extensions to the standard MPC formulation, such as robust MPC [52–54], data-driven MPC [55–57], stochastic MPC [58], and temporal logic MPC [59–61].

Recently, the predictive safety filter has been proposed as a special MPC formulation, which searches for a minimally adapted safe control input from a reference control signal [39]. Unlike generic MPC controllers, predictive safety filters are usually employed as additional safety verification and enhancement modules for other controllers, especially learning-based controllers [62]. Readers are referred to [39, 63] for comprehensive reviews of recent advances in MPC and predictive safety filters. Although much more computationally efficient than reachability analysis, the computational complexity of MPC grows rapidly with the length of the prediction horizon. Unfortunately, the prediction horizon can be very long for ensuring recursive feasibility and stability [64]. Moreover, the MPC open-loop optimization problem is usually nonlinear for nonlinear systems, and nonconvex if the state constraints are nonconvex. Nonconvexity makes online computation even more challenging for real-world applications. This motivates the methodology of control barrier functions (CBFs), which is the key focus of this thesis.

Control Barrier Functions

To further alleviate the online computational burden and efficiently analyze safety, safety certificate functions, namely barrier certificates [65–67] or CBFs [68, 69], have emerged as a promising methodology. Given a dynamical system, a CBF defines a viability kernel, which is a subset of the safe set. The relationship between CBFs and viability/reachability theory can be understood through set invariance [70] and Nagumo's theorem [71]. In this sense, the value function of a reachability optimal control problem is a candidate CBF for the considered dynamical system and the safe set [72, 73]. By enforcing the inner product of the CBF derivative and the vector field of the controlled system to be bounded, safety is rigorously guaranteed at all times. CBFs have been shown to be powerful and scalable in control input design for control-affine systems, as this condition can be encoded

as a linear constraint, even for a nonlinear system and a nonconvex CBF, in a quadratic programming (QP) framework [69]. The safe controller can be designed by solving online QP problems at every state [68, 74]. CBFs have been extensively investigated under a stochastic setting [75, 76], considering robustness [77, 78], high-relative degree [79–81], task specifications [82, 83], and combining with MPC [84, 85].

Although efficient in online computation and flexible with input constraints, nearly all CBF-based methods require a pre-designed CBF [39]. As we will show in Chapter 3, an inappropriate CBF may result in an extremely pathological vector field even for a simple second-order linear system, with an ellipsoidal unsafe set. One way to design a CBF is solving an infinite horizon reach-avoid optimal problem, of which the value function is a CBF [73]. However, as we have discussed before, the computation of a reachability optimal control problem suffers from the curse of dimensionality [86]. Building upon the idea of the dynamic programming principle, reinforcement learning (RL) based methods have been proposed to find approximate solutions [87–90]. Although these methods are usually more computationally efficient in practice, the designed CBF loses formal guarantees for safety.

Efficient convex optimization techniques, especially sum-of-squares programming (SOSP), pave the way for overcoming these limitations. Briefly speaking, SOSP provides sufficient and convex conditions to verify the positivity of real polynomials [91]. SOSP problems can be equivalently transformed into semi-definite programming problems using parsers [92, 93], then solved by modern optimization solvers [94, 95]. SOSP has been used for designing various certificate functions, such as Lyapunov functions [96–98], barrier certificates [65, 67], and density functions [99–101]. Recently, SOSP has been used for CBF design [102–105]. In these works, the CBF and the feedback controller are parameterized by polynomials, then co-designed by solving an SOSP problem. However, the co-design programs introduce cross-products between the CBF and the controller, rendering the SOS constraints bilinear. Solving nonconvex SOSP problems is challenging; existing numerical

methods such as PK iteration [106, 107] and path-following [108] have no guarantees for convergence to a feasible solution without warm-starting. When high-relative degree between the safe set and the system dynamics or input constraints are considered, solving the SOSP problem becomes even more challenging [103]. To overcome these difficulties, this thesis develops advanced approaches to co-design a CBF and a feedback controller under challenging settings.

The aforementioned safety analysis and control synthesis methodologies are mostly based on optimization, making scalability especially important as the system scales up. Take quadratic programming (QP) based methods [68] for example, the computational complexity using interior-point methods is approximately $\mathcal{O}(n^3)$ [109], where n is the dimension of the control input. When a system of m homogeneous agents is considered, the complexity of solving a QP problem is $\mathcal{O}(m^3n^3)$. The cubic growth of complexity makes these methods hard to apply to large-scale multi-agent systems (MASs) [110]. Most of the existing results in this direction involve a centralized approach; however, multi-agent considerations call for distributed solution regimes. To improve computational scalability for MASs, this thesis proposes distributed and data-driven optimization-based approaches, which scale well with the number of agents.

CBF-based distributed algorithms have been proposed in [110–112]. These works propose to directly split the CBF constraints into multiple parts for neighbouring agents to facilitate distributed implementation. Under the assumption that each local optimization problem is feasible, all the CBF constraints are satisfied. However, this assumption is usually much stronger than that of feasibility of the nominal centralized problem. Moreover, optimality of the nominal centralized problem by the distributed controller is not guaranteed. An improved constraint sharing mechanism is developed in [102], where the CBF constraints are dynamically tuned for feasibility, but for single-agent systems. Optimality is further considered in [113], but for multi-agent systems with only one CBF constraint. A dynamical constraint allocation scheme among agents based on a consensus protocol is proposed. In this

thesis, we deal with the problem of guaranteeing feasibility of local problems across iterations while preserving optimality, under multiple CBF safety constraints.

Another problem is ensuring safety while preserving other properties, such as stability. Isolated consideration of either property may result in loss of the other one. For example, consider a car navigating to a target point while avoiding an obstacle. A controller that ensures collision avoidance may always drive the car away from the goal. On the contrary, a stabilizing controller may unavoidably make the car crash into the obstacle. Recently, methods that incorporate a CBF and a control Lyapunov function (CLF) into a light QP framework have been proposed [68, 114]. Unlike MPC, which realizes stability through objective functions, these methods rely on a CLF-induced constraint. As pointed out by [114], the CLF constraint may conflict with the CBF constraint at some points, resulting in undesired equilibrium points or loss of local stability. To guarantee stability and safety simultaneously for a dynamical system, a quadratic programming-based filter has been proposed [69]. Control Lyapunov function (CLF) and CBF constitute separate constraints in such a filter for stability and safety. The filter can accommodate any locally Lipschitz continuous reference signal and subsequently provides the closest certified control signal. For nonlinear control affine systems, CLF and CBF constraints are linear in the input for any given state. Feasibility, however, is not guaranteed: as a trade-off, the CLF constraint is relaxed using a slack variable in the program. Consequently, local stability for the closed-loop system using the designed controller is not guaranteed [115]. Additionally, equilibrium points, other than the origin, exist.

The authors in [116] unify the CLF and CBF into one function called CLBF for simultaneous stability and safety, but only for a bounded control invariant set. Given a CLF and a CBF, a CLBF can be constructed through a linear combination of these functions. In [77], the CLF constraint is modified with additional parameters. By properly designing these parameters, the closed-loop system can be locally stable at the origin. If the reference control signal stabilizes

the system, the filter [69] is shown to guarantee local stability [117]. Under a similar assumption, [118] proposes a new filter that only has a CBF constraint. Sufficient conditions to estimate the region of attraction (ROA) are also provided. To better accommodate the CLF and CBF constraints, [119] proposes to lift either the CLF or the CBF constraint into the objective function. When the CLF constraint is lifted as a penalty term, the closed-loop system is shown to be locally stable in a non-empty region of attraction (ROA) if the penalty parameter is larger than a certain level.

In [69], the CLF and CBF are obtained separately. The potential conflict between these two functions is the main reason for the undesired closed-loop behaviour. The concept of *compatibility* has been identified as a key property to efficiently accommodate the conflict [118, 119]. Compatibility is also related to the control sharing property [102]. In essence, this property necessitates the existence of a controller that satisfies both the CLF and CBF constraints at every state. To design compatible CLF and CBF, a sum-of-squares programming-based synthesis method has been proposed [120]. However, it is important to note that achieving compatibility for the entire state space may be impossible if the complementary set of the control invariant set is bounded [121]. To address these issues, this thesis proposes a new QP framework and numerical methods to mitigate the conflict between the safety and stability constraints defined by a CBF and a CLF that satisfy the relaxed compatibility condition.

An illustration of reachability/viability analysis, MPC, and CBF methodologies in terms of optimality and scalability is shown in Figure 1.1. Reachability/viability analysis demonstrates the highest optimality, as the value function defines the maximal viability kernel, but suffers from the worst scalability due to the curse of dimensionality. The CBF methodology exhibits the best scalability but the lowest optimality, given that its optimization framework is myopic and the CBF typically serves as an approximation of the value function in the reachability optimal control problem. MPC occupies an intermediate position among these methodologies.

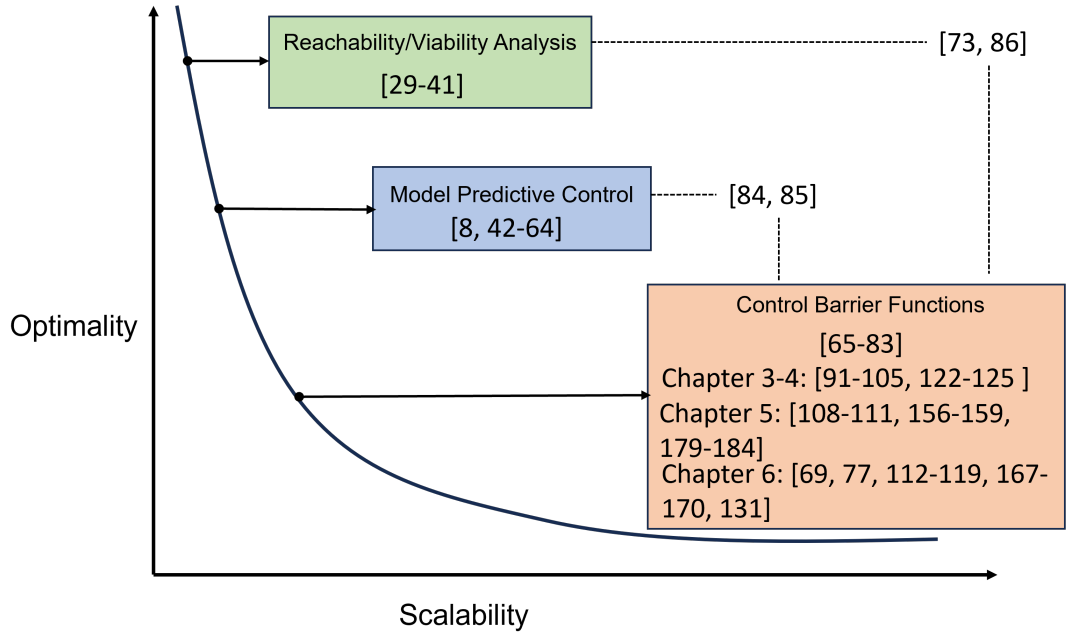


Figure 1.1: Illustration of reachability/viability analysis, MPC and CBFs methodologies

1.2 Contribution to the Literature

As an outline, this thesis focuses on the CBF methodology for safety analysis and safe controller design. Efficient convex optimization based approaches are proposed to overcome the fundamental limitations of the CBF methods. Our contributions to the literature is summarized as follows.

Convex Co-Design of CBFs and Feedback Controllers

We propose an efficient methodology to design a control barrier function and an associated state feedback controller using sum-of-squares programming, for both linear and nonlinear systems. The control barrier function and feedback controller are parameterized by polynomials and co-designed in one unified sum-of-squares program, thus overcoming the need for iterative algorithm [103, 105, 122–124]. Moreover, our formulation is applicable to high and mixed relative degree cases without the need to use backstepping methods.

We also extend the existing literature when considering limits in the system inputs. $\mathcal{L}$-1 norm constrained limitation set is considered in [104, 122, 123, 125]. Specifi-

cally, [104, 122] introduce bilinear constraints in the sum-of-squares programming, [123] proposes a quantifier exchange to drop the dependency on the control input, and [125] proposes re-parameterization for linear systems. In our work, $\mathcal{L}$-1, $\mathcal{L}$-2, and $\mathcal{L}-\infty$ norm constrained limitations are all addressed by means of convex constraints. These input constraints can be appended to the CBF and feedback controller co-design program. Efficacy of our methodology has been demonstrated on various examples, including linear, polynomial, and non-polynomial systems.

One journal paper about the results of linear systems in Chapter 3 is under review [126].

H. Wang, K. Margellos, A. Papachristodoulou, C. De Persis, "Convex Co-design of Control Barrier Functions and Safe Feedback Controllers Under Input Constraints", *arXiv preprint arXiv:2403.11763*, 2024.

The results for nonlinear systems nonconvex design in Chapter 4.2 has been published in [122].

H. Wang, K. Margellos, A. Papachristodoulou, "Safety Verification and Controller Synthesis for Systems with Input Constraints", *IFAC-PapersOnLine*, 2023.

The results for nonlinear systems convex design in Chapter 4.1 is under preparation for submission as a journal paper.

A review on this problem has been published in [127].

H. Wang, K. Margellos, A. Papachristodoulou, "Assessing Safety for Control Systems Using Sum-of-Squares Programming", *Polynomial Optimization, Moments, and Applications (M. Kocvara, B. Mourrain, C. Riener, eds.), Springer-Verlag*, 2023.

Distributed Safety Verification and Safe Controller Design

We provide a distributed algorithm for designing safe controllers for multi-agent systems. Under the assumption of existence of CBFs, a centralized safe control design problem is formulated. Our distributed algorithm parallelizes computation

by decomposing the centralized problem into local problems, while guaranteeing feasibility of every local problem across iterations. The optimal solution returned by our algorithm is guaranteed to be the same as that of the nominal centralized problem, therefore satisfying all the CBF constraints.

In view of practical implementation, and since the convergence guarantees of the proposed algorithm are asymptotic, we propose a truncation mechanism for early termination. This comes at the cost of sacrificing strict guarantees of satisfying the CBF constraints, however, it reduces the communication and computation burden of an asymptotic algorithm, and is accompanied this with a safety verification scheme.

The proposed verification scheme can be applied more generally to verify safety for multi-agent systems. In particular, instead of verifying safety over the whole state-space, which is challenging for multi-agent systems, we propose a scenario-based verification algorithm for a probabilistic quantification of safety by means of satisfying CBF constraints. A sequential sampling algorithm is proposed to sample scenarios efficiently in a distributed fashion. We accompany our solution with a probabilistic safety certificate; to achieve this, we extend the state-of-the-art result [128, Theorem 1] to a multi-agent setting. Both lower and upper bounds on the probability of being unsafe are established, while the safety verification program is also shown to be amenable to parallelized computation.

One journal paper about these results is under review [129].

H. Wang, A. Papachristodoulou, K. Margellos, "Distributed Safe Control Design and Safety Verification for Multi-agent Systems", *arXiv preprint arXiv:2303.12610*, 2023.

The results for safety verification in Chapter 5.3 have been published in [130].

H. Wang, A. Papachristodoulou, K. Margellos, "Distributed Control Design and Safety Verification for Multi-agent Systems", *IEEE Conference on Decision and Control*, 2023.

Safe and Stable Filter Design

We propose a new filter based on the *relaxed compatibility* condition for a CBF and CLF. We demonstrate that our method obtains the desired closed-loop behaviour, i.e. local asymptotic stability and elimination of interior equilibrium points. Our method does not require a stabilizing nominal controller to enhance stability of the designed optimal controller. Moreover, the optimal controller is guaranteed to be locally Lipschitz continuous inside the invariant set without an *a priori* assumption on the linear independence of the CBF- and CLF- induced linear constraints, which is assumed in [131]. Additionally, we provide a design method for a relaxed compatible pair of CLF and CBF for a polynomial dynamical system using sum-of-squares programming. The inherent nonlinearities in the program are addressed through an iterative algorithm. We validate the efficacy of our method by comparative studies against other stat-of-the-art methods, and our method shows superior filter performance.

One journal paper about these results is under review [132].

H. Wang, K. Margellos, A. Papachristodoulou, "Safe and Stable Filter Design Using Relaxed Compatible Control Barrier/Lyapunov Functions".

One conferences paper has been published in [133].

H. Wang, K. Margellos, A. Papachristodoulou, "Relaxed Compatibility Between Control Barrier and Lyapunov Functions", *International Conference on Control*, 2024.

Other Publications

Other work during my DPhil studies which is not included in this thesis also include the following publication:

H. Wang, K. Margellos, A. Papachristodoulou, "A Time-triggered Dimension Reduction Algorithm for the Optimal Task Assignment Problem", *European Journal of Control*, 2022.

Our contributions are summarized in Figure 1.2.

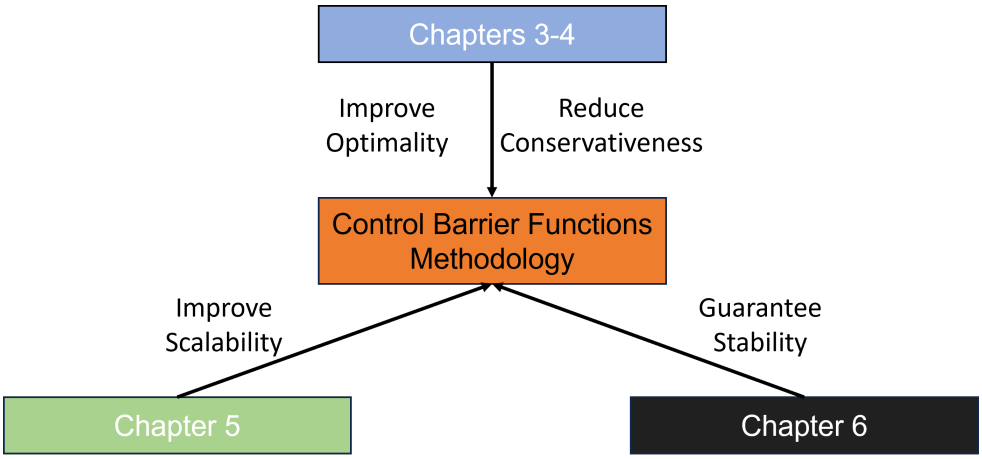


Figure 1.2: Contributions of the thesis

1.3 Outline of the Thesis

Chapter 2

In this chapter, we describe the *safety* problem for control systems, and introduce the control barrier functions (CBFs) methodology, which is the backbone for this thesis. We also provide optimization preliminaries including sum-of-squares programming and scenario optimization.

Chapter 3

In this chapter, we present an overview of the CBF design problem and its relationship with reach-avoid optimal control. We then propose sum-of-squares (SOS) programs to co-design CBFs and state feedback controllers for linear systems, under different conditions of safety specifications. Various input constraints can be incorporated into the SOS programs by adding additional convex constraints. The results are discussed and presented on numerical examples.

Chapter 4

In this chapter, we provide extensions for the results presented in Chapter 3 to nonlinear polynomial and non-polynomial systems. We show how the Psatz and

Taylor expansion can be used for a nonlinear co-design of CBFs and feedback controllers. To further reduce conservatism of convex design, an iterative algorithm that tackles nonconvex design is presented. Efficacy of the proposed methodology is demonstrated on various experiments.

Chapter 5

In this chapter, we propose distributed iterative algorithms for safe control design and safety verification for networked multi-agent systems. These algorithms rely on distributing a control barrier function (CBF) related quadratic programming (QP) problem. The proposed distributed algorithm addresses infeasibility issues of existing schemes through a cooperation mechanism between agents. The resulting control input is guaranteed to be optimal, and satisfies the CBF constraints of all agents. Furthermore, a truncated algorithm is proposed to facilitate computational implementation. The performance of the truncated algorithm is evaluated using a distributed safety verification algorithm. The algorithm quantifies safety for multi-agent systems probabilistically by means of CBFs. Both upper and lower bounds on the probability of safety are obtained using the so called scenario approach. Both the scenario sampling and safety verification procedures are fully distributed. The efficacy of our algorithms is demonstrated by an example on multi-robot collision avoidance.

Chapter 6

In this chapter, we propose a quadratic programming-based filter for safe and stable controller design, via a Control Barrier Function (CBF) and a Control Lyapunov Function (CLF). Our method guarantees safety and local asymptotic stability without the need for an asymptotically stabilizing control law. Feasibility of the proposed program is ensured by a mild regularity condition, termed *relaxed compatibility* between the CLF and CBF. The resulting optimal control law is guaranteed to be locally Lipschitz continuous. We also analyze the closed-loop behaviour by characterizing the equilibrium points, and verifying that there are no equilibrium points in the interior of the control invariant set except at the origin.

For a polynomial system and a semi-algebraic safe set, we provide a sum-of-squares program to synthesize a relaxed compatible pair of CLF and CBF. The proposed method is compared with other methods in the literature using numerical examples, where our method demonstrates superior filter performance and guarantees safety and local stability.

Chapter 7

We summarize the main contributions of the thesis in this chapter and suggest some directions for future research.

2 Background

This chapter provides preliminary information used in subsequent chapters. We first introduce the safety problem and the control barrier functions, then give optimization preliminaries, including sum-of-squares programming and scenario optimization.

2.1 Safety and Control Barrier Functions

Consider a nonlinear system

$$x^\dagger = f(x, u), \tag{2.1}$$

where $x(t) \in \mathbb{R}^n$ is the n-dimensional state, and $u(t) \in \mathcal{U} \subset \mathbb{R}^m$ is the m-dimensional control input, for any time instance t to be specified in the sequel. $f(\cdot, \cdot) : (\mathbb{R}^n, \mathbb{R}^m) \to \mathbb{R}^n$ is a locally Lipschitz function. Here we use $x^\dagger$ to represent the state transition for both continuous and discrete time system. For continuous-time, we consider

$$\dot{x}(t) = f(x(t), u(t)),$$

and for discrete time

$$x(t+1) = f(x(t), u(t)).$$

For such a system, with a slight abuse of notation, $x(t, u, x_0)$ denotes the state at time $t \geq 0$ (for discrete-time systems we also require t to be an integer), with inputs u starting from $x_0 \in \mathcal{I}$, where $\mathcal{I} \subset \mathbb{R}^n$ is the initial set from which the system starts and is assumed to be non-empty. Here we assume that $x(t, u, x_0)$ is unique for any $t \in \mathcal{L}$, where $\mathcal{L} = [0, +\infty)$ for continuous-time systems and $\mathcal{L} = \mathbb{N}_+$ for discrete time ones. To ease notation, in some occurrences we drop the dependence of u on time.

Safety is a system-set property which models whether a dynamical system can stay within a set, $\mathcal{S} \subset \mathbb{R}^n$. Such a set is usually determined by application requirements, such as collision avoidance in robotic applications. We assume that $\mathcal{S}$ is non-empty and compact, and is the super zero-level set of a differentiable function $s(x) : \mathbb{R}^n \to \mathbb{R}$, i.e.

$$\mathcal{S} := \{x | s(x) \geq 0\}. \tag{2.2}$$

Only time-invariant safe sets are considered in this chapter. Naturally, we should have $\mathcal{I} \subseteq \mathcal{S}$, such that system starts from the safe set.

Definition 2.1.1. Consider system (2.1) and the safe set $\mathcal{S}$ and initial set $\mathcal{I}$. The system is *safe* if for any $x_0 \in \mathcal{I}$, $t \in \mathcal{T} := [0, T]$, there exists a locally Lipschitz continuous controller $u(\cdot) \in \mathcal{U}$ such that $x(t, u, x_0) \in \mathcal{S}$. Such a control input is called a *safe control input*.

Given the definition of safety we present here two questions of interest. The first question is to verify safety for a given system, safe and initial sets.

Question 1 (safety verification): consider dynamical system (2.1), control set $\mathcal{U}$, initial set I and safe set $\mathcal{S}$, verify whether (2.1) is safe.

Question 2 (safe control design): consider dynamical system (2.1), control set $\mathcal{U}$, initial set $\mathcal{I}$ and safe set $\mathcal{S}$, design a safe control input.

These questions involve solving analysis and control synthesis problems under input and state constraints; one method is to formulate and solve an optimal control

problem as follows: for any (x, t),

$$\begin{aligned} V(x,t) = \max_{u(\cdot)} \min_{\tau \in [t,T]} \; & s(\xi(\tau)) \\ \text{subject to } & \xi^{\dagger} = f(\xi, u), \\ & u(\tau) \in \mathcal{U}, \quad \text{for all } \tau \in [t,T], \\ & \xi(0) = x. \end{aligned} \tag{2.3}$$

The cost of the optimal control problem (2.3) involves choosing the control input $u(\tau)$ that maximizes the minimum value of $s(\xi)$ over the time interval $\mathcal{T}$. This is because a larger $s(\xi)$ implies a higher level of safety since the safe set is defined over the super-level set of $s(\xi)$. $V(x,t) : \mathbb{R}^n \times \mathbb{R}_+ \to \mathbb{R}$ is the value function of (2.3). Given x, if $V(x,0) \geq 0$, we conclude that the system is safe starting from this state. It is now clear that the set

$$\mathcal{V} := \{x | V(x,0) \geq 0\} \tag{2.4}$$

contains all the initial states x_0 from which (2.1) is safe. This is because for any $x_0 \in \mathcal{V}$ and $t \in \mathcal{T}$, there exists $u \in \mathcal{U}$ such that $s(x(t,u,x_0)) \geq 0$. The following statement based on $\mathcal{V}$ provides an answer to Questions 1&2.

If $\mathcal{I} \subseteq \mathcal{V}$, we have that for any $x_0 \in \mathcal{I}$, $V(x_0, 0) \geq 0$, safety is ensured using the controller $u^*(\cdot)$ obtained from (2.3). Conversely, $\mathcal{V}$ is non-empty if the system is safe, since $\mathcal{I} \subseteq \mathcal{V}$ and $\mathcal{I}$ is non-empty. The optimal control problem (2.3) is well-posed, but is hard to solve in practice as it is equivalent to solving a partial differential equation [29]. Our goal is to construct or approximate the set $\mathcal{V}$ efficiently with alternative methods. We conclude and prove that, if $\mathcal{V}$ is non-empty, then

Proposition 2.1.1. Properties of $\mathcal{V}$

1. $\mathcal{V} \subseteq \mathcal{S}$.
2. If $T < \infty$, then there exists a controller $u(\cdot) \in \mathcal{U}$, such that $x(t,u,x_0) \in \mathcal{S}$ for all $t \in [0,T], x_0 \in \mathcal{V}$.
3. If $T \to \infty$, then there exists a controller $u(\cdot) \in \mathcal{U}$, such that $x(t,u,x_0) \in \mathcal{V}$ for all $t \geq 0, x_0 \in \mathcal{V}$.

These properties are straightforward following the aforementioned discussion. We formalize and prove the last property as it is the backbone of the analysis in the sequel.

Lemma 2.1.1 ([29, Proposition 1])**.** *Consider system* (2.1)*, initial set* $\mathcal{I}$*, and safe set* $\mathcal{S}$*, with* $T \to \infty$*. For any* (x,t)*, let* $V(x,t)$ *be the optimal value function of* (2.3)*, and define* $\mathcal{V}$ *as in* (2.4)*. If* $\mathcal{V}$ *is non-empty, then Property 3 holds.*

Definition 2.1.2. Consider system (2.1) and a set $\mathcal{K}$. A set $\mathcal{B} \subset \mathbb{R}^n$ is called a *T-invariant set* of $\mathcal{K}$ there exists a locally Lipschitz continuous controller $u(\cdot) \in \mathcal{U}$, such that $x(t,u,x_0) \in \mathcal{K}$ for any $t \in [0,T]$, and any $x_0 \in \mathcal{B}$.

Note that $\mathcal{V}$ is the maximal T-invariant set for system (2.1) and safe set $\mathcal{S}$. This is a corollary of Lemma 2.1.1, as one can see that any point exhibiting the invariance property will be within $\mathcal{V}$. As we can see here, if a system is safe, then we can always construct a set $\mathcal{V}$ by solving the optimal control problem (2.3). The relationship between the existence of a T-invariant set $\mathcal{B}$ of $\mathcal{S}$ and safety is shown in the following converse theorem.

Theorem 2.1.1. *Consider system* (2.1)*, initial set* $\mathcal{I}$*, safe set* $\mathcal{S}$ *and time interval* $[0,T]$*. If the system is safe, then there exists a* T*-invariant set* $\mathcal{B}$ *of* $\mathcal{S}$*, such that* $\mathcal{I} \subseteq \mathcal{B} \subseteq \mathcal{S}$*. Conversely, if there exists a* T*-invariant set* $\mathcal{B}$*, such that* $\mathcal{I} \subseteq \mathcal{B} \subseteq \mathcal{S}$*, then the system is safe.*

Proof. We first consider $T < \infty$. For the sufficiency part of the proof suppose the system is safe, and for every $\mathcal{B}$ such that $\mathcal{I} \subseteq \mathcal{B} \subseteq \mathcal{S}$, $\mathcal{B}$ is not a T-invariant set of $\mathcal{S}$. Then there exists at least one $x_0 \in \mathcal{B}$ and $t \in \mathcal{T}$, such that $x(t,u,x_0) \notin \mathcal{S}$. Let $\mathcal{B} = \mathcal{I}$, this contradicts to the assumption that the system is initially safe. For the necessity part we have that for such a $\mathcal{B}$, we have that for any $x_0 \in \mathcal{B}$ and $t \in \mathcal{T}$, $x(t,u,x_0) \in \mathcal{B} \subseteq \mathcal{S}$.

□

With Theorem 2.1.1 in hand, the safety verification problem is equivalent to an existence problem. By constructing a T-invariant set and the corresponding control

input u, we solve the safety problem. Invariance introduced in Definition 2.1.2 is related to a safe set, as it requires that trajectories starting from the invariant set stay in the safe set. However, there is no safe set appearing in the third property. In fact, with $T \to \infty$, every trajectory starting from $\mathcal{V}$ can stay within $\mathcal{V}$ for all time. We pay specific attention to the case $T \to \infty$ since this commonly holds for general nonlinear systems. The set $\mathcal{V}$ obtained by (2.3) is also called a *control invariant set* if it is non-empty.

Definition 2.1.3. A set $\mathcal{B}$ is called a control invariant set for system (2.1) if there exists a locally Lipschitz continuous controller $u(\cdot) \in \mathcal{U}$ such that $x(t, u, x_0) \in \mathcal{B}$ for any $x_0 \in \mathcal{B}$, and any $t \geq 0$.

If $\mathcal{B}$ is a control invariant set sucht that $\mathcal{B} \subseteq \mathcal{S}$, then it is for sure a T-invariant set of $\mathcal{S}$ by definition. Control invariance is an isolated property for a set $\mathcal{B}$ compared with invariant set, which also depends on $\mathcal{S}$. The existence of a control invariant set $\mathcal{B}$ also reveals safety.

Theorem 2.1.2. *Consider system* (2.1)*, initial set* $\mathcal{I}$ *and safe set* $\mathcal{S}$*. If there exists a control invariant set* $\mathcal{B}$ *such that* $\mathcal{I} \subseteq \mathcal{B} \subseteq \mathcal{S}$*, then the system is safe. Conversely, if the system is safe, then there exists a control invariant set* $\mathcal{B}$ *such that* $\mathcal{I} \subseteq \mathcal{B} \subseteq \mathcal{S}$*.*

Proof of Theorem 2.1.4 is analogous to that of Theorem 2.1.1.

In this section we introduce and analyze control invariance for continuous-time systems. We first exploit conditions for control invariance with set representation, and give alternative conditions with function representation.

2.1.1 Control Invariance with Set Representation

Control invariance is closely related to the concept of *tangent cone*, which is defined in a point-wise manner over the set.

Definition 2.1.4. Let $\mathcal{D}$ be a compact set. The tangent cone of $\mathcal{D}$ at x is the set

$$\wp_{\mathcal{D}}(x) = \left\{ z \in \mathbb{R}^n : \liminf_{h \to 0} \frac{\text{dist}(x + hz, \mathcal{D})}{h} = 0, \right\} \tag{2.5}$$

where

$$\text{dist}(x, \wp) = \inf_{y \in \wp} ||x - y||, \tag{2.6}$$

is a distance function and $|| \cdot ||$ denotes the Euclidian norm.

The tangent cone to a set $\mathcal{D}$ at x is shown in Figure 2.1. We only illustrate the case of $x \in \partial\mathcal{D}$ since $\wp_{\mathcal{D}}(x) = \mathbb{R}^n$ if $x \in \text{Int}(\mathcal{D})$, and $\wp_{\mathcal{D}}(x) = \emptyset$ if $x \in \mathcal{D}'$. In the geometric sense, for a convex set $\mathcal{D}$, every vector $f(x, u) \in \wp_{\mathcal{D}}(x)$ points inside $\mathcal{D}$, or at least is tangent to the boundary curve of $\mathcal{D}$ at x. The tangent cone clearly relates to control invariance.

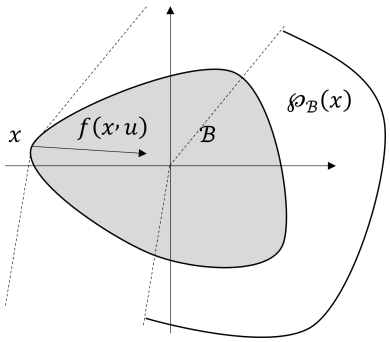


Figure 2.1: Tangent cone for a compact set $\mathcal{B}$ at $x \in \partial\mathcal{B}$.

Theorem 2.1.3 (Nagumo's Theorem [71])**.** *Consider a system $\dot{x} = f(x, u)$. Let $\mathcal{B}$ be a closed and convex set. Then the set $\mathcal{B}$ is a control invariant set for the system if and only if there exists a locally Lipschitz continuous $u(\cdot) \in \mathcal{U}$, such that*

$$f(x, u) \in \wp_{\mathcal{B}}(x) \quad \forall x \in \partial\mathcal{B}$$

For continuous system dynamics, $f(x, u)$ is required to point inside the set $\mathcal{B}$ only for $x \in \partial\mathcal{B}$.

The tangent cone is defined point-wise in x. For special sets such as ellipsoidal control invariant or polyhedral control invariant sets, the tangent cone can be directly calculated. However, there is no explicit form for general compact sets $\mathcal{B}$. Verifying condition (2.5) and designing the control input are still challenging tasks.

2.1.2 Control Invariance with Function Representation

In this section we consider the invariance condition by the set representation of a set $\mathcal{B}$. Similarly to the definition of $\mathcal{S}$, suppose $\mathcal{B}$ is the zero super-level set of a function $b(x) : \mathbb{R}^n \to \mathbb{R}$:

$$\mathcal{B} := \{x | b(x) \geq 0\}.$$

Then the control invariance condition is summarized in the following theorem.

Theorem 2.1.4. *Consider a system $\dot{x} = f(x, u)$ and a closed set $\mathcal{B}$ described by the zero super-level set of a continuously differentiable function $b(x)$. Then $\mathcal{B}$ is a control invariant set for the system if there exits a locally Lipschitz continuous controller $u(\cdot) \in \mathcal{U}$, such that*

$$\frac{\partial b(x)}{\partial x} f(x, u) > 0 \quad \forall x \in \partial\mathcal{B} \tag{2.7}$$

Theorem 2.1.4 certifies control invariance by checking a Lyapunov-like derivative condition for every x on the boundary of the set $\mathcal{B}$. Expanding $b(x)$ at $x(t)$ with respect to t we have

$$b(x(t + \delta t)) = b(x(t)) + \dot{b}(x)|_{x=x(t)}\delta t + o(x(t)),$$

where $o(x(t))$ is a small residual term. If $b(x(t)) = 0$ for $x(t) \in \partial\mathcal{B}$, then $b(x(t + \delta t)) > 0$ if $\dot{b}(x) > 0$. We note here that $\dot{b}(x) = \frac{\partial b(x)}{\partial x} f(x, u)$ is necessary to be strictly positive to bypass the case $o(x(t)) < 0$.

Although condition (2.7) is only sufficient for control invariance and requires the function $b(x)$ to be continuously differentiable, it is easy to be checked numerically. For a given state $x(t) \in \partial\mathcal{B}$ and function $b(x)$, checking (2.7) can be done by solving a feasibility optimisation problem

$$\begin{aligned} &\text{find } u(\cdot) \in \mathcal{U} \\ &\text{subject to } \frac{\partial b(x)}{\partial x} f(x, u) > 0, \forall x \in \partial\mathcal{B}. \end{aligned}$$

Checking (2.7) for any $x \in \partial\mathcal{B}$ is arduous as there are infinite conditions to check. We show how to deal with this using sum-of-squares programming in Chapter 3 for linear systems and Chapter 4 for nonlinear systems.

2.1.3 Control Invariance for Discrete-Time Systems

The control invariance condition for discrete-time systems is slightly different from that for continuous-time systems. To check the control invariance of a given set $\mathcal{B}$, one needs to check the state transition for every $x \in \mathcal{B}$ but not only $x \in \partial\mathcal{B}$. This is because for a discrete-time system $x(t+1) = f(x(t), u(t))$, there may be the case that $f(\partial\mathcal{B}, u) \in \mathcal{B}$, but $f(\mathcal{B}, u) \notin \mathcal{B}$. The control invariance condition for discrete-time systems is formalized in the following theorem, as a natural counterpart of the Nagumo's Theorem 2.1.3.

Theorem 2.1.5. *Consider a system $x(t+1) = f(x(t), u(t))$ and a compact set $\mathcal{B}$ defined by the zero super-level set of a function $b(x)$. Then $\mathcal{B}$ is a control invariant set for the system if and only if there exists $u(\cdot) \in \mathcal{U}$, such that*

$$b(f(x, u)) \geq 0 \quad \forall x \in \mathcal{B} \tag{2.8}$$

We have come up with conditions to check control invariance of $\mathcal{B}$ using a function $b(x)$. If the set $\mathcal{B}$ further satisfies $\mathcal{I} \subseteq \mathcal{B} \subseteq \mathcal{S}$, then the system $x^\dagger = f(x, u)$ is safe, using Theorem 2.1.2. This motivates the concept of *control barrier functions* (CBFs) [65, 68].

Definition 2.1.5. Consider system $x^\dagger = f(x, u)$, an initial set set $\mathcal{I}$, a safe set $\mathcal{S}$, and a continuously differentiable function $b(x) : \mathbb{R}^n \to \mathbb{R}$. $b(x)$ is called a *control barrier function* (CBF) if the set $\mathcal{B} := \{x \in \mathbb{R}^n : b(x) \geq 0\}$ satisfies

1. $\mathcal{I} \subseteq \mathcal{B} \subseteq \mathcal{S}$,
2. $\mathcal{B}$ is control invariant.

2.2 Sum-of-Squares Programming

Consider a polynomial optimisation problem

$$\begin{aligned} &\min_{x \in \mathbb{R}^n} \ f(x) \\ \text{subject to } &g(x) \leq 0, \\ &h(x) = 0, \end{aligned} \tag{2.9}$$

where $f(x) : \mathbb{R}^n \to \mathbb{R}$, $g(x) : \mathbb{R}^n \to \mathbb{R}$, $h(x) : \mathbb{R}^n \to \mathbb{R}$ are polynomial functions. For convex $f(x)$, $g(x)$ and $h(x)$, (2.9) is a convex optimisation problem that can be solved efficiently. If any of these functions is non-convex, we can consider the following problem instead:

$$\begin{aligned} &\max_{\gamma} \ \gamma \\ &\text{subject to } \gamma - f(x) \leq 0, \forall x \in \mathcal{K} := \{x | g(x) \leq 0, h(x) = 0\}. \end{aligned} \tag{2.10}$$

Now the decision variable is γ. The new constraint $\gamma - f(x) \leq 0$ is a nonpositivity constraint for a polynomial, and scales linearly in γ. However, checking positivity of polynomials of any fixed even degree $d \geq 4$ is NP-hard [134, Corollary 2.9]. To deal with this issue, sum-of-squares decomposition is proposed.

Definition 2.2.1. A polynomial $f(x)$ is said to be a sum-of-squares polynomial if there exists polynomials $f_i(x)$, such that

$$f(x) = \sum_i f_i(x)^2. \tag{2.11}$$

We also call (2.11) a sum-of-squares decomposition for $f(x)$. Clearly, if a function $f(x)$ has a sum-of-squares decomposition, then it is non-negative for all $x \in \mathbb{R}^n$. One question here is whether all positive polynomials admit a sum-of-squares decomposition - the answer is no. As an example, the Motzkin polynomial $1 + x^2y^4 + x^4y^2 - 3x^2y^2$ is nonnegative, but has no sum-of-squares decomposition. In the sequel, we use $\mathbb{R}[x]$ to denote the set of real polynomials in x, and $\Sigma[x]$ to denote the set of sum-of-squares polynomials in x.

Computing the sum-of-squares decomposition (2.11) can be efficient as it is equivalent to a positive semidefinite feasibility program.

Lemma 2.2.1. *Consider a polynomial $f(x)$ of degree $2d$ in $x \in \mathbb{R}^n$. Let $z(x)$ be a vector of all monomials of degree less than or equal to d. Then $f(x)$ admits a sum-of-squares decomposition if and only if*

$$f(x) = z(x)^\top Q z(x), Q \succeq 0. \tag{2.12}$$

In Lemma 2.2.1, $z(x)$ is a user-defined monomial basis if d and n are fixed. In the worst case, $z(x)$ has $\begin{pmatrix} n+d \\ d \end{pmatrix}$ components, and Q is a $\begin{pmatrix} n+d \\ d \end{pmatrix} \times \begin{pmatrix} n+d \\ d \end{pmatrix}$ squared matrix. The necessity of Lemma 2.2.1 is natural from the definition of positive semi-definite matrix, considering the monomial $z(x)$ as a vector of new variables z_i. The sufficiency is shown by factorizing $Q = L^\top L$. Then $z(x)^\top Q z(x) = (Lz(x))^\top Lz(x) = ||Lz(x)||_2^2 \geq 0$.

Given $z(x)$, finding Q to decompose $f(x)$ as in (2.12) is a semi-definite program, which can be solved efficiently using interior point methods. Selecting the basis $z(x)$ depends on the structure of $p(x)$ to be decomposed.

Example 2.2.1 ([91, Example 4.1])**.** Consider a quartic polynomial in two variables $p(x_1, x_2) = 2x_1^4 + 2x_1^3x_2 - x_1^2x_2^2 + 5x_2^4$. We want to check whether it is a sum-of-squares polynomial. Define $z(x) = [x_1^2 \quad x_2^2 \quad x_1x_2]^\top$ and consider the following decomposition

$$p(x_1, x_2) = z(x)^\top \underbrace{\begin{bmatrix} 2 & -3 & 1 \\ -3 & 5 & 0 \\ 1 & 0 & 5 \end{bmatrix}}_{Q} z(x)$$

Q is a positive semi-definite matrix since $Q = L^\top L$, and $L = \frac{1}{\sqrt{2}} \begin{bmatrix} 2 & -3 & 1 \\ 0 & 1 & 3 \end{bmatrix}$. Therefore, $p(x_1, x_2)$ has the following SOS decomposition:

$$p(x_1, x_2) = \frac{1}{2}(2x_1^2 - 3x_2^2 + x_1x_2)^2 + \frac{1}{2}(x_2^2 + 3x_1x_2)^2.$$

Definition 2.2.2. $Q(x) : \mathbb{R}^n \to \mathbb{R}^m \times \mathbb{R}^m$ is said to be a sum-of-squares matrix, i.e., $Q(x) \in \Sigma[x]^m$, if there exists a polynomial matrix $T(x)$, such that

$$Q(x) = T(x)^\top T(x).$$

Then if $Q(x) \in \Sigma[x]^m$, we have that $Q(x_0) \succeq 0$ for every $x_0 \in \mathbb{R}^n$.

Lemma 2.2.2. *$Q(x)$ is a sum-of-squares matrix with respect to a monomial basis $Z(x)$, if and only if there exists a symmetric matrix $H \succeq 0$, such that*

$$Q(x) = (Z(x) \otimes I_m)^\top H (Z(x) \otimes I_m).$$

Similarly to sum-of-squares polynomials, one can check whether $Q(x)$ is a sum-of-squares matrix with respect to some monomial basis by solving a semi-definite program.

Going back to problem (2.10), γ should satisfy that the intersected set $\{x|\gamma - f(x) \geq 0\} \bigcap \mathcal{K} \bigcap \{x|\gamma - f(x) = 0\}$ is empty. Here the condition $\{x|\gamma - f(x) > 0\}$ is expressed by $\{x|\gamma - f(x) \geq 0\} \bigcap \{x|\gamma - f(x) = 0\}$. The intersected set has a special structure: it is defined by a series of polynomial equality and inequality constraints.

Definition 2.2.3. A set $\mathcal{X} \subset \mathbb{R}^n$ is semi-algebraic if it can be represented using polynomial equality and inequality constraints. If there are only equality constraints, the set is algebraic.

Three types of polynomials are defined based on a series of polynomials $f_1(x), \ldots, f_m(x)$.

Definition 2.2.4. The *monoid* generated by $f_1(x), \ldots, f_m(x)$ is denoted by

$$\text{monoid}(f_1(x), \ldots, f_m(x)) = \prod_{i=1}^{m} f_i(x)^{k_i}, k_i \in \mathbb{N},$$

where $\mathbb{N}$ is the set of non-negative integers.

Definition 2.2.5. The *ideal* generated by polynomials $f_1(x), \ldots, f_m(x)$ is denoted by

$$\text{ideal}(f_1(x), \ldots, f_m(x)) = \sum_{i=1}^{m} \lambda_i(x) f_i(x),$$

where $\lambda_1(x), \ldots, \lambda_m(x) \in \mathbb{R}[x]$.

Definition 2.2.6. The *cone* generated by polynomials $f_1(x), \ldots, f_m(x)$ is denoted by

$$\text{cone}(f_1(x), \ldots, f_m(x)) = \sum_{i=1}^{r} \sigma_i(x) t_i(x),$$

where $t_i(x), \ldots, t_r(x) \in \text{monoid}(f_1(x), \ldots, f_m(x))$, and $\sigma_1(x), \ldots, \sigma_r(x) \in \Sigma[x]$.

The cone and ideal are closely related to the emptiness of semi-algebraic sets. Specifically, the ideal is related to algebraic sets.

Lemma 2.2.3.

$$-1 \in \text{ideal}(f_1(x), \ldots, f_m(x)) \Leftrightarrow \{x \in \mathbb{R}^n | f_i(x) = 0, \forall i = 1, \ldots, m\} = \emptyset.$$

The cone is related to the sets defined by polynomial inequality constraints.

Lemma 2.2.4.

$$-1 \in \text{cone}(f_1(x), \ldots, f_m(x)) \Leftrightarrow \{x \in \mathbb{R}^n | f_i(x) \geq 0, \forall i = 1, \ldots, m\} = \emptyset.$$

Lemmas 2.2.3 and 2.2.4 are known as *Nullstellensatz*. Based on these results, we have the main result of this section, called the *Positivstellensatz* theorem.

Theorem 2.2.1 ([135, Theorem 4.4.2])**.** *Let $\mathcal{K}$ be a semi-algebraic set, $\mathcal{K} := \{x \in \mathbb{R}^n | f_i(x) \geq 0, g_j(x) = 0, h_k(x) \neq 0, \ \forall i = 1, \ldots, m, j = 1, \ldots, p, k = 1, \ldots, q\}$. Let $f(x) \in \text{cone}(f_1(x), \ldots, f_m(x)), g(x) \in \text{ideal}(g_1(x), \ldots, g_p(x)), h(x) \in \text{monoid}(h_1(x), \ldots, h_q(x))$. We have*

$$f(x) + g(x) + h(x)^2 = 0 \Leftrightarrow \mathcal{K} = \emptyset. \tag{2.13}$$

The *Positivstellensatz* gives a necessary and sufficient condition to test whether a semi-algebraic set is empty or not. When $f_i(x)$, $g_j(x)$, and $h_k(x)$ are given for every $i = 1, \ldots, m, j = 1, \ldots, p, k = 1, \ldots, q$, and the power of $h(x)$ are fixed and given, solutions of Equation (2.13) can be found by solving an SOS feasibility problem

$$\begin{aligned} &\text{find } \lambda_1(x), \ldots, \lambda_p(x) \in \mathbb{R}[x], \sigma_1(x), \ldots, \sigma_p(x) \in \Sigma[x] \\ &\text{subject to } -f(x) - g(x) - h(x)^2 \in \Sigma[x] \end{aligned} \tag{2.14}$$

Here $\lambda_i(x)$'s and $\sigma_j(x)$'s are multipliers introduced in $f(x)$ and $g(x)$, following the definitions of cone and ideal.

One thing worth mentioning here is the choice of these multipliers. If (2.14) is infeasible with certain $\deg(\lambda_i(x))$ and $\deg(\sigma_j(x))$, this does not necessarily imply that $\mathcal{K} = \emptyset$. In this case one can increase the degree of the multipliers and repeat the test (which is of non-decreasing accuracy) but this will result in a larger semidefinite program.

Example 2.2.2 ([136, Course 6])**.** Consider the feasibility problem

$$\mathcal{S} = \{(x, y) \in \mathbb{R}^2 : f(x, y) \geq 0, h(x, y) = 0\}$$

where

$$f(x,y) = x - y^2 + 3, \quad h(x,y) = y + x^2 + 2$$

By Positivestellensatz, $\mathcal{S}$ is empty if and only if there exist polynomials $s_1, s_2 \in \Sigma[x,y]$ and $t \in \mathbb{R}[x,y]$, such that

$$-1 = s_1 + s_2 f + th$$

A certificate is given by

$$s_1 = \frac{1}{3} + 2\left(y + \frac{3}{2}\right)^2 + 6\left(x - \frac{1}{6}\right)^2, s_2 = 2, t = -6.$$

This validates emptiness of $\mathcal{S}$.

The following lemma gives relaxed conditions to test the emptiness of a semi-algebraic set.

Lemma 2.2.5 (S-Procedure)**.** *Suppose $t_i(x) \in \Sigma[x], i \in \mathcal{I}$, then*

$$\begin{aligned} &p(x) - \sum_{i\in\mathcal{I}} t_i(x) q_i(x) \in \Sigma[x] \Rightarrow \\ &p(x) \geq 0 \ \forall x \in \bigcap_{i\in\mathcal{I}} \{x | q_i(x) \geq 0\}. \end{aligned}$$

Suppose $l_i(x) \in \mathbb{R}[x], i \in \mathcal{I}$, then

$$\begin{aligned} &p(x) - \sum_{i\in\mathcal{I}} l_i(x) q_i(x) \in \Sigma[x] \Rightarrow \\ &p(x) \geq 0 \ \forall x \in \bigcap_{i\in\mathcal{I}} \{x | q_i(x) = 0\}. \end{aligned}$$

Compared with *Positivstellensatz*, the S-procedure only gives a sufficient condition for the emptiness of a semi-algebraic set. However, a good feature is that there is no multiplier for $f(x)$. This is especially useful when $f(x)$ is a parameterized function to be constructed. We also highlight here that the S-procedure is sufficient and necessary for quadratic $f(x)$ and $g(x)$. In this case, $t(x)$ degenerates into a positive scalar.

Using *Positivestellensatz* for linear functions results in Farkas Lemma.

Lemma 2.2.6 ([137, Farkas Lemma]). *The set* $\{x \in \mathbb{R}^n | Ax + b \geq 0, Cx + d = 0\}$ *is empty if and only if there exist* $\lambda \geq 0$ *and* μ *such that*

$$\lambda^\top A + \mu^\top C = 0, \lambda^\top b + \mu^\top d = -1.$$

We have now shown the necessary basic results of sum-of-squares decomposition, and how to use it to characterize the emptiness of semi-algebraic sets.

2.3 Scenario Optimization

Robust optimization offers a methodology to immunize decisions against uncertainty. An uncertain optimization problem is formulated as

$$\begin{aligned} &\min_{z \in \mathcal{Z}} c^\top z \\ &\text{subject to } z \in \mathcal{Z}_x, \text{ for all } x \in \mathcal{H}, \end{aligned} \tag{2.15}$$

where $z \in \mathbb{R}^n$ is a decision variable constrained by a set $\mathcal{Z} \subseteq \mathbb{R}^n$, $c \in \mathbb{R}^n$ is a constant vector. The uncertain constraint set $\mathcal{Z}_x$ is parameterized by an uncertain parameter x, which is a random variable defined on a probability space $(\mathcal{H}, \mathcal{F}, \mathbb{P})$. Even in the case where $\mathcal{Z}_x$ is convex for any $x \in \mathcal{H}$, if the uncertain parameters' domain $\mathcal{H}$ is continuous or even unknown, robust optimization problem is usually hard (or even impossible) to perform. The so called scenario approach, on the other hand, proposes to solve the problem over finite empirical records, named *scenarios* ans accompany the resulting solution with probabilistic guarantees on its feasibility properties. The corresponding scenario optimization problem can be formulated as

$$\begin{aligned} &\min_{z \in \mathcal{Z}} c^\top z \\ &\text{subject to } z \in \bigcap_{r=1,\ldots,R} \mathcal{Z}_{x^{(r)}}, \end{aligned} \tag{2.16}$$

where $x^{(r)}$, $r = 1, \ldots, R$ are *scenarios* sampled independently from the set $\mathcal{H}$. If $\mathcal{Z}_x$ is convex for any $x \in \mathcal{H}$, the scenario optimization (2.16) is a convex optimization problem which can be solved efficiently.

Definition 2.3.1 (violation probability). The violation probability of a a given $z \in \mathcal{Z}$ is defined as $V(z) = \mathbb{P}\{x \in \mathcal{H} : z \notin \mathcal{Z}_x\}$.

Clearly, the optimal solution of (2.16) satisfies $z^* \in \bigcap_{r=1,\dots,R} \mathcal{Z}_{x^{(r)}}$, but is not necessarily within $\mathcal{Z}_x$ for an arbitrary new $x \in \mathcal{H}$. i.e., we do not necessarily have $V(z^*) = 0$. In fact, z^* is itself a random variable as it depends on the choice of the scenarios $x^{(r)}$, $r = 1, \dots, R$. To align with our subsequent developments, we will characterize $V(z^*)$ for a slightly more general scenario program; to this end, consider the following scenario optimization problem with relaxed constraints:

$$\begin{aligned} \min_{z \in \mathcal{Z}, \xi^{(r)} \geq 0, r=1,\dots,N} \; & c^\top z + \rho \sum_{r=1}^{N} \xi^{(r)} \\ \text{subject to } & h(z, x^{(r)}) \leq \xi^{(r)}, r = 1, \dots, R, \end{aligned} \tag{2.17}$$

where $x^{(r)}, r = 1, \dots, R$ are independently sampled from $(\mathcal{H}, \mathcal{F}, \mathbb{P})$. Notice that here we consider the explicit characterization of the constraint set $\mathcal{Z}_{x^{(r)}}$ through functions $h(z, x^{(r)})$, $r = 1, \dots, R$.

A constraint $z \in \mathcal{Z}_{x^{(r)}}$ is called a support constraint if its removal (while the other constraints are maintained) changes the solution z^*. We impose the following assumption.

Assumption 2.3.1 ([128, Assumption 2])**.** *Consider problem* (2.17) *and assume that a unique optimal solution* $(z^*, \{\xi^{*,(r)}\}_{r=1}^R$ *exists almost surely with respect to the choice of* $\{x^{(r)}\}_{r=1}^R$*. We further assume that the optimal solution* $(z^*, \{\xi^{*,(r)}\}_{r=1}^R$ *of* (2.17) *coincides almost surely with respect to the choice of the scenarios* $x^{(r)}$*,* $r = 1, \dots, R$ *with the solution that is obtained after eliminating all the constraints that are not of support.*

The violation probability $V(z^*) = \mathbb{P}\{x \in \mathcal{H} : f(z^*, x) > 0\}$ can be then characterized by the following theorem.

Theorem 2.3.1 ([128, Theorem 4])**.** *Consider the optimization problem* (2.17)*. Suppose that its optimal solution* $(z^*, \{\xi^{*,(r)}\}_{r=1}^R$ *satisfies Assumption 2.3.1. Given a confidence parameter* $\beta \in (0, 1)$*, for any* $k = 0, 1, \dots, R-1$ *consider the polynomial equation in the* t *variable*

$$\binom{R}{k} t^{R-k} - \frac{\beta}{2R} \sum_{j=k}^{R-1} \binom{j}{k} t^{j-k}$$

$$-\frac{\beta}{6R}\sum_{j=R+1}^{4R}\left(\begin{array}{c} j \\ k \end{array}\right)t^{j-k}=0, \tag{2.18}$$

and for $k=R$ *consider the polynomial equation*

$$1-\frac{\beta}{6R}\sum_{i=R+1}^{4R}\left(\begin{array}{c} j \\ k \end{array}\right)t^{j-R}=0. \tag{2.19}$$

For any $k=0,\ldots,R-1$, (2.18) *has exactly two solutions in* $[0,+\infty)$, *which we denote with* $\underline{t}(k)$ *and* $\bar{t}(k)$ $(\underline{t}(k))\leq\bar{t}(k)$. *Instead,* (2.19) *has only one solution in* $[0,+\infty)$, *which we denote with* $\bar{t}(N)$, *while we define* $\underline{t}(N)=0$. *Let* $\underline{\epsilon}(k):=\max\{0,1-\bar{t}(k)\}$ *and* $\bar{\epsilon}(k):=1-\underline{t}(k), k=0,1,\ldots,R$. *We then have that*

$$\mathbb{P}^R\{\underline{\epsilon}(s^*)\leq V(z^*)\leq\bar{\epsilon}(s^*)\}\geq 1-\beta, \tag{2.20}$$

where s^* *is the number of* $x^{(r)}$*'s for which* $h(z^*,x^{(r)})\geq 0$.

3

Convex Co-design of Control Barrier Functions and Feedback Controllers for Linear Systems

In this chapter, we address the safety verification and safe control design problems mentioned in Chapter 2. A convex co-design framework for control barrier functions and state feedback controllers is proposed for linear dynamical systems.

3.1 Introduction

Safety is essential for feedback control systems. As a system is steered from an initial set to a target set, safety requires that the trajectory of the system avoids entering an unexpected region, or to remain inside a safe set. On the state space, safety is always formulated by means of constraints imposed on states. Based on these descriptions, two questions are raised: given a dynamical system $\dot{x} = f(x, u)$, a set of initial sets $\mathcal{I}$, and a set of safe states $\mathcal{S}$, (i) verify whether there exists a control input $u(\cdot)$, so that the trajectories starting from $\mathcal{I}$ stay inside $\mathcal{S}$; (ii) design such a control law $u(\cdot)$ that guarantees safety. The Control Barrier Functions (CBF) approach answers these two questions by using a continuously differentiable function that satisfies certain properties [65, 68, 138].

A CBF aims to separate the safe and unsafe regions by its zero super- and sub-level sets; the initial set also belongs to the level set. In addition, there exists a control law, such that the vector field points towards the safe side on its zero sub-level set [65]. This property is also known as *invariance*, characterized by Nagumo's theorem [71]. It is therefore guaranteed that if the system starts from a point inside the zero super-level set, the system can always stay inside. Given a CBF, the controller that guarantees safety can be designed according to the direction requirement of vector field. However, synthesizing a CBF is not a trivial task even for linear systems. In general, even verifying a CBF is an NP-hard problem [104, Proposition 2].

Designing a CBF is even more challenging when the relative degree between the function defines safe set and the system dynamics is high or mixed [103]. For relative degree we mean the number of times we need to differentiate a function whose level set encodes the safe set along the system dynamics until the control explicitly shows [79, 80]. High or mixed relative degree is commonly seen in robotics collision avoidance problems, where the safe set is usually defined over positions for the obstacles, but the control signals are imposed on accelerations.

3.1.1 Motivating Cases

Case 1 (Pathological vector field of CBF-QP)**.** Consider a continuous-time linear system with state matrix $A = \begin{bmatrix} -1 & -1 \\ 0 & -1 \end{bmatrix}$ and input matrix $B = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$. The system is controllable. Let $x = [x_1, x_2]^\top \in \mathbb{R}^2$ denote the states. The unsafe region is defined by $\mathcal{S}^c = \{x | s(x) \leq 0\}$, where $s(x) = x_1^2 + x_2^2 - 1$. Using $s(x)$ as a control barrier function in a quadratic programming framework [68] for controller design, we obtain:

$$\min\ u^\top u,$$

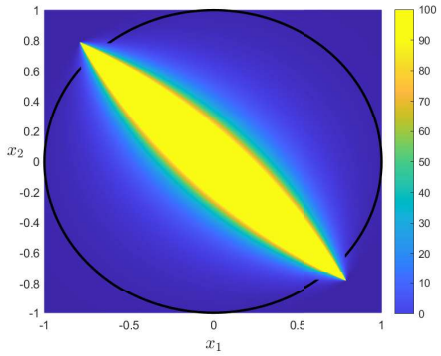
$$\text{subject to } \dot{s}(x) + 10 s(x) \geq 0.$$

Following [139] , the analytical solution is given by

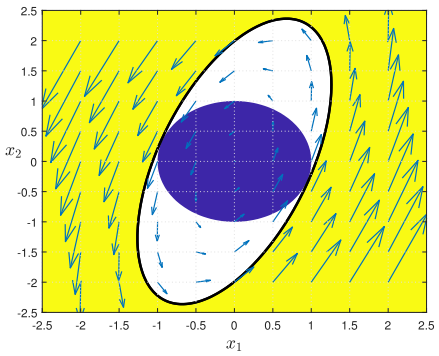
$$u_s(x) = -\frac{\min\{0, f(x)\}}{g(x)},$$

where

$$\begin{aligned} f(x) &= \frac{\partial s(x)}{\partial x} Ax + 10s(x) \\ &= 8x_1^2 - 2x_1x_2 + 8x_2^2 - 10, \\ g(x) &= \frac{\partial s(x)}{\partial x} B = 2x_1 + 2x_2. \end{aligned}$$



(a) Values of $||u_s(x)||_2^2$ for $-1 \leq x_1 \leq 1$, $-1 \leq x_2 \leq 1$. The value of $||u_s(x)||_2^2$ is limited to 100 for visualization. The controller is only locally smooth, and the Lipschitz constant is large in a local region as the value varies a lot with little state changes. Different selection of a class-$\mathcal{K}$ function does not change the result for $x \in \partial\mathcal{B}$, which is the black curve in the figure. Clearly our designed feedback controller $u_b(x) = 1.4164x_1 + 0.59702x_2$ is globally smooth as it is linear.



(b) Comparison of the two control barrier functions in Case 1. The blue round region is the unsafe set $\mathcal{S}^c$. The yellow open region is the control invariant set $\mathcal{B} := \{x|b(x) \geq 0\}$, and $\partial\mathcal{B} := \{x|b(x) = 0\}$ is the black curve. Blue arrows represents the vector field $Ax + Bu_b(x)$, which points inward $\mathcal{B}$ on $\partial\mathcal{B}$.

Figure 3.1: Simulation results for Case 1.

When $g(x)$ tends to zero, and $f(x) < 0$, $u_s(x)$ tends to infinity. As a consequence, the system cannot be safe at some points, especially points on $\partial\mathcal{S}$ with limited control authority. We also show in Figure 3.1a that the Lipschitz constant of $u_s(x)$ is very large. Later on, we will show that, by solving the proposed convex program (3.4) with $||u_b(x)||_2^2 \leq 8$ for x such that $b(x) = 0$, we obtain a new control barrier function $b(x) = 0.88391x_1^2 - 0.50767x_1x_2 + 0.25205x_2^2 - 1$, and a feedback controller $u_b(x) = 1.4164x_1 + 0.59702x_2$. Comparison of the two control barrier functions is shown in Figure 3.1b. It can be seen that the value of $||u_s(x)||_2^2$ is comparably large for small x_2. Meanwhile, our synthesized controller is constrained by $||u_b(x)||_2^2 \leq 8$ for $x \in \partial\mathcal{B}$.

Case 2 (Mixed relative degree)**.** Consider a third-order continuous-time linear system with $\dot{x}_1 = x_2 + x_3$, $\dot{x}_2 = x_1 + u_1$, $\dot{x}_3 = x_1 + u_2$, where $x = [x_1, x_2, x_3]^\top \in \mathbb{R}^3$ is the state, $u = [u_1, u_2]^\top \in \mathbb{R}^2$ is the input. The unsafe region is defined by $\mathcal{S}^c := \{x | s(x) \leq 0\}$, where $s(x) = x_1^2 + x_2^2 - 1$. Let the relative degree be the number of times we need to differentiate $s(x)$ along the dynamics until the control input u appears in the resulting expression. For this case, the relative degree between $s(x)$ and the system is mixed, as the input u_1 appears in the first derivative of $s(x)$, whereas u_2 appears in the second derivative. $s(x)$ can not be directly used as a CBF using high-relative degree (exponential) CBF techniques [79–81]. By solving the convex program (3.4) that we will propose in the sequel, we obtain a control barrier function $b(x) = x_1^2 + x_2^2 - 0.0129x_3^2 - 1$, and a feedback controller $u_1(x) = -2x_1 + 38.9x_2$, $u_2(x) = 76.8x_1 - 0.5x_3$, which guarantees safety for the system. Clearly, the relative degree between $b(x)$ and the system dynamics is one. We highlight here that the backstepping CBF method [140] would require a series of explicit pre-synthesized safe controllers which are, however, not needed for our method, which only requires the solution of a convex program.

The convex synthesis program and extensions for linear systems are presented in Section 3.2. Simulation results are shown in Section 3.3. Section 3.4 concludes the chapter.

3.2 Convex Design for Linear Systems

In this section, we propose convex synthesis programs to construct a CBF and an affine safe feedback controller. In Section 3.2.1, we first consider a *global* design for $\mathcal{B} = \{x \in \mathbb{R}^n : b(x) \geq 0\}$ to be control invariant. For this case, we consider the unsafe set $\mathcal{S}^c$ to be bounded on a subspace of $\mathbb{R}^n$. This is common for robot collision avoidance problems, where the position space is a subspace of the robot state space. The control invariant set $\mathcal{B}$ is constructed *globally* as its projection to the subspace of $\mathcal{S}^c$ is unbounded. For the second case in Section 3.2.2, we construct

a control invariant set $\mathcal{B}^c = \{x \in \mathbb{R}^n : b(x) \leq 0\}$ around a bounded initial set. This control invariant set is called *local* as we will show it is bounded on $\mathbb{R}^n$.

Consider a continuous-time linear system:

$$\dot{x} = Ax + Bu, \tag{3.1}$$

where $x(t) \in \mathbb{R}^n$, $u(t) \in \mathcal{U} \subseteq \mathbb{R}^m$ are the state and control input, and $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times m}$. We assume that the system is stabilizable. Throughout the chapter, the CBF $b(x)$ and feedback controller $u(x)$ are parameterized as follows.

$$b(x) = (x-c)^\top \Omega^{-1}(x-c) - 1, \tag{3.2a}$$

$$u(x) = Y\Omega^{-1}(x-c) + d, \tag{3.2b}$$

where $\Omega \in \mathbb{R}^{n\times n}, Y \in \mathbb{R}^{m\times n}$ are matrices to be designed, $c \in \mathbb{R}^n$ and $d \in \mathbb{R}^m$ are constant vectors that will be clear in the sequel.

3.2.1 Global Design

Consider the safe set $\mathcal{S}$ defined by a union of semi-algebraic sets as:

$$\mathcal{S} := \bigcup_{i=1}^{o} \{x \in \mathbb{R}^n : s_i(\overline{x}) \geq 0\}, \tag{3.3}$$

where $x = [\overline{x}^\top, \underline{x}^\top]^\top$, with $\overline{x} \in \mathbb{R}^{\overline{n}}$, $\underline{x} \in \mathbb{R}^{\underline{n}}$ and $\overline{n} + \underline{n} = n$. If $\underline{n} = 0$, the safe set is defined over all the states.

Assumption 3.2.1. *$\mathcal{S}$ is a semi-algebraic set, and $\mathcal{S}^c$ is bounded on the space $\mathbb{R}^{\overline{n}}$.*

Let $c = [\overline{c}^\top, \underline{c}^\top]^\top \in \mathbb{R}^n$ be a vector of constants such that $\text{rank}([B, Ac]) = \text{rank}(B)$, and consider the following optimization program:

$$\min \ \text{Tr}(\overline{\Omega}) \tag{3.4a}$$

$$\text{subject to } 0 \prec \overline{\Omega} = \overline{\Omega}^\top \in \mathbb{R}^{\overline{n}\times\overline{n}}, 0 \succ \underline{\Omega} = \underline{\Omega}^\top \in \mathbb{R}^{\underline{n}\times\underline{n}}, \tag{3.4b}$$

$$0 \prec R = R^\top \in \mathbb{R}^{\overline{n}\times\overline{n}}, Y \in \mathbb{R}^{m\times n}, \tag{3.4c}$$

$$\sigma_1(\overline{x}), \ldots, \sigma_o(\overline{x}) \in \Sigma[\overline{x}], \epsilon > 0 \tag{3.4d}$$

$$\Omega = \begin{bmatrix} \overline{\Omega} & 0 \\ 0 & \underline{\Omega} \end{bmatrix} \tag{3.4e}$$

$$\Omega A^\top + Y^\top B^\top + A\Omega + BY \succeq 0 \tag{3.4f}$$

$$\begin{bmatrix} R & I_{\overline{n}} \\ I_{\overline{n}} & \overline{\Omega} \end{bmatrix} \succeq 0 \tag{3.4g}$$

$$1 - \overline{x}_c^\top R \overline{x}_c + \sum_{i=1}^{o} \sigma_i(\overline{x}) s_i(\overline{x}) - \epsilon \in \Sigma[\overline{x}], \tag{3.4h}$$

where $\overline{x}_c = \overline{x} - \overline{c}$, $\underline{x}_c = \underline{x} - \underline{c}$, $x_c = x - c$. Notice that this a convex optimization program, where the objective function is linear, and is subject to semi-define constraints The cost function is to minimize the volume of the set $\{\overline{x} \in \mathbb{R}^{\overline{n}} : -(\overline{x} - \bar{c})^\top \overline{\Omega}^{-1} (\overline{x} - \bar{c}) + 1 \geq 0\}$, thus indirectly maximizing the volume of the projection set of $\mathcal{B}$ on the space $\mathbb{R}^{\overline{n}}$. An alternative formulation is $\max \log\det \overline{\Omega}^{-1}$ [6, Section 2.2.4]. However, this is not supported by SeDuMi, which is the solver we are using to solve the semi-definite program. In the following theorem, we give the main result of the chapter, a convex program to synthesize a CBF $b(x)$ and a feedback controller $u(x)$ under Assumption 3.2.1.

Theorem 3.2.1. *Consider Assumption 3.2.1, and let $\mathcal{U} = \mathbb{R}^m$. Assume that a solution to (3.4) exists and is denoted by $\overline{\Omega}, \underline{\Omega}, R, Y, \{\sigma_i(\cdot)\}_{i=1}^o, \epsilon$. Set $u(x) = Y\Omega^{-1}(x - c) + d$ where $d \in \mathbb{R}^m$ is such that $Bd + Ac = 0$. We then have that*

1. *$\mathcal{B} := \{x \in \mathbb{R}^n : (x - c)^\top \Omega^{-1} (x - c) - 1 \geq 0\} \subseteq \mathcal{S}$.*
2. *$\mathcal{B}$ is a control invariant set for $\dot{x} = Ax + Bu(x)$.*

Proof. We first prove that satisfaction of (3.4e), (3.4g) and (3.4h) are sufficient for $\mathcal{B} \subseteq \mathcal{S}$. Given that $R \succ 0$ and $\overline{\Omega} \succ 0$, using Schur complement, (3.4g) is equivalent to $R - \overline{\Omega}^{-1} \succeq 0$. Multiplying the latter condition by $\overline{x}_c^\top = (\overline{x} - \bar{c})^\top$ on the left and by $\overline{x}_c$ on the right, we obtain

$$-\left(\overline{x}_c^\top \overline{\Omega}^{-1} \overline{x}_c - 1\right) + (\overline{x}_c^\top R \overline{x}_c - 1) \geq 0, \forall \overline{x} \in \mathbb{R}^{\overline{n}}.$$

Then we have the following relationship

$$\begin{aligned} &\{\overline{x} \in \mathbb{R}^{\overline{n}} : (\overline{x} - \bar{c})^\top \overline{\Omega}^{-1} (\overline{x} - \bar{c}) - 1 > 0\} \\ &\qquad \subseteq \{\overline{x} \in \mathbb{R}^{\overline{n}} : (\overline{x} - \bar{c})^\top R (\overline{x} - \bar{c}) - 1 > 0\}. \end{aligned}$$

The former set inclusion implies the following relationship for the closures of the associated sets,

$$\{\overline{x} \in \mathbb{R}^{\overline{n}} : (\overline{x} - \overline{c})^\top \overline{\Omega}^{-1} (\overline{x} - \overline{c}) - 1 \geq 0\} \\ \subseteq \{\overline{x} \in \mathbb{R}^{\overline{n}} : (\overline{x} - \overline{c})^\top R (\overline{x} - \overline{c}) - 1 \geq 0\}.$$

The two sets involved are subsets of $\mathbb{R}^{\overline{n}}$. Considering them as the base of cylinder sets in $\mathbb{R}^n$, we obtain

$$\{x \in \mathbb{R}^{n} : (\overline{x} - \overline{c})^\top \overline{\Omega}^{-1} (\overline{x} - \overline{c}) - 1 \geq 0\} \\ \subseteq \{x \in \mathbb{R}^{n} : (\overline{x} - \overline{c})^\top R (\overline{x} - \overline{c}) - 1 \geq 0\}. \tag{3.5}$$

Invoking Lemma 2.2.5 for polynomial functions $s_i(\overline{x})$, $i = 1, \ldots, o$, (3.4h) indicates that

$$-(\overline{x} - \overline{c})^\top R (\overline{x} - \overline{c}) + 1 > 0 \ \forall x : s_1(\overline{x}), \ldots, s_o(\overline{x}) < 0, \tag{3.6}$$

which in turn implies

$$\exists i \in \{1, 2, \ldots, o\} : s_i(\overline{x}) \geq 0, \ \forall x \in \mathbb{R}^n \\ \text{such that } (\overline{x} - \overline{c})^\top R (\overline{x} - \overline{c}) - 1 \geq 0. \tag{3.7}$$

Converting the above relationship in set inclusion form, we have

$$\{x \in \mathbb{R}^{n} : (\overline{x} - \overline{c})^\top R (\overline{x} - \overline{c}) - 1 \geq 0\} \subseteq \mathcal{S}. \tag{3.8}$$

Combining (3.5) with (3.8) implies

$$\{x \in \mathbb{R}^{n} : (\overline{x} - \overline{c})^\top \overline{\Omega}^{-1} (\overline{x} - \overline{c}) - 1 \geq 0\} \subseteq \mathcal{S}. \tag{3.9}$$

Given that $\underline{\Omega} \prec 0$, and using the block representation in (3.4e), we have

$$(x - c)^\top \Omega^{-1} (x - c) - 1 = (\overline{x} - \overline{c})^\top \overline{\Omega}^{-1} (\overline{x} - \overline{c}) \\ + (\underline{x} - \underline{c})^\top \underline{\Omega}^{-1} (\underline{x} - \underline{c}) - 1 \leq (\overline{x} - \overline{c})^\top \overline{\Omega}^{-1} (\overline{x} - \overline{c}) - 1. \tag{3.10}$$

Using (3.10) into the relationship in (3.9), and recalling that $\mathcal{B} = \{x \in \mathbb{R}^n : (x - c)^\top \Omega^{-1} (x - c) - 1 \geq 0\}$, we obtain

$$\mathcal{B} \subseteq \{x \in \mathbb{R}^{n} : (\overline{x} - \overline{c})^\top \overline{\Omega}^{-1} (\overline{x} - \overline{c}) - 1 \geq 0\} \subseteq \mathcal{S}. \tag{3.11}$$

We then prove that $\mathcal{B}$ (the zero super-level set of $b(x)$) is a control invariant set. Since Ω (decomposed as in (3.4e)), is optimal for (3.4), it will have to satisfy (3.4f). We will show that (3.4f) is sufficient for $\mathcal{B}$ to be control invariant, thus establishing the claim. We guarantee control invariance by using an affine state feedback controller as (3.2b), where $K \in \mathbb{R}^{m\times n}$. Transforming the coordinate from x to $x_c = x - c$, and since d is such that $Bd + Ac = 0$, the transformed system dynamics are given by

$$\dot{x}_c = (A + BK)x_c.$$

In the new coordinate $b(x)|_{x=x_c+c} = \tilde{b}(x_c) = x_c^\top \Omega^{-1} x_c - 1$. If $\dot{b}(x) \geq 0$ for any $x \in \mathbb{R}^n$, then $\mathcal{B}$ is invariant. Notice that $\dot{b}(x) \geq 0$ is equivalent to

$$\begin{aligned}\dot{b}(x) = \dot{\tilde{b}}(x_c) = x_c^\top (A^\top \Omega^{-1} + \Omega^{-1} A \\ + K^\top B^\top \Omega^{-1} + \Omega^{-1} BK) x_c \geq 0.\end{aligned} \tag{3.12}$$

Satisfaction of (3.12) for any $x \in \mathbb{R}^n$ is equivalent to

$$A^\top \Omega^{-1} + \Omega^{-1} A + K^\top B^\top \Omega^{-1} + \Omega^{-1} BK \succeq 0. \tag{3.13}$$

Left and right multiplying by Ω on both sides of (3.13), we obtain

$$\Omega A^\top + A\Omega + \Omega K^\top B^\top + BK\Omega \succeq 0.$$

Substituting $K\Omega$ with a new matrix $Y \in \mathbb{R}^{m\times n}$, and noticing that Ω is invertible, we equivalently obtain (3.4f). The latter is thus a sufficient condition for $\mathcal{B}$ to be invariant.

By the proof of the previous part it follows that $u(x) = K(x - c) + d$ renders $\mathcal{B}$ invariant. Moreover, $Y = K\Omega$, which in turn implies that $K = Y\Omega^{-1}$. Therefore, $u(x) = Y\Omega^{-1}(x - c) + d$ guarantees invariance. □

Figure 3.2 visualizes an example of a control invariant set $\mathcal{B}$ and $\mathcal{S}$ on $\mathbb{R}^3$. To gain an intuitive understanding of Theorem 3.2.1, we analytically construct a control invariant set for a simple example.

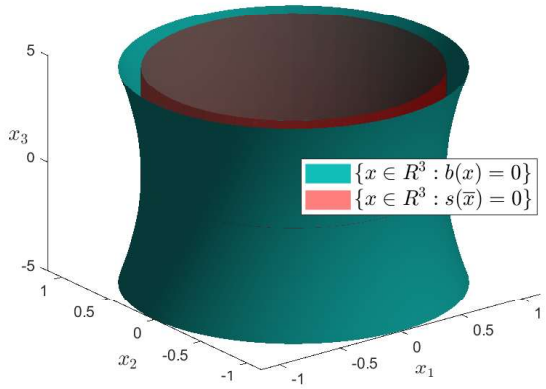


Figure 3.2: Visualization of Case 2 in Section 3.1.1, where the state $\overline{x} = [x_1, x_2] \in \mathbb{R}^2$, $\underline{x} = x_3 \in \mathbb{R}$. The safe set $\mathcal{S} := \{x \in \mathbb{R}^3 : s(\overline{x}) \geq 0\}$ is a cylinder expanded from a set on $\mathbb{R}^2$ to $\mathbb{R}^3$. The region outside of the blue hyperboloid represents the set $\mathcal{B} := \{x \in \mathbb{R}^3 : b(x) \geq 0\}$, which is control invariant from our construction. The safe set $\mathcal{S} := \{x \in \mathbb{R}^3 : s(\overline{x}) \geq 0\}$ is the outside of the inner red cylinder. We can see from the figure that $\mathcal{B} \subseteq \mathcal{S}$.

Example 3.2.1. Consider a car moving along a line

$$\dot{\overline{x}} = \underline{x}$$

$$\dot{\underline{x}} = u$$

where $\overline{x}$ represents the position and $\underline{x} \in \mathbb{R}$ represents the velocity. Let

$$\mathcal{S} = \{x \in \mathbb{R}^2 : s(\overline{x}) := \overline{x}^2 - 1 \geq 0\}.$$

We follow the construction in Theorem 3.2.1. The vector $c \in \mathbb{R}^2$ that satisfies $\text{rank}([B\ Ac]) = \text{rank}(B)$ is any vector such that $\underline{c} = 0$. We also fix $\overline{c} = 0$. Consequently $d = 0$.

Consider the decision variables $\overline{\Omega} \in \mathbb{R}_{>0}$, $\underline{\Omega} \in \mathbb{R}_{<0}$, $R \in \mathbb{R}$, $Y = [Y_1\ Y_2] \in \mathbb{R}^{1\times 2}$, $\sigma(\overline{x}) \in \Sigma[\overline{x}]$ and $\varepsilon > 0$. The constraints in (3.4) can be written as follows

$$\Omega = \begin{bmatrix} \overline{\Omega} & 0 \\ 0 & \underline{\Omega} \end{bmatrix}, \tag{3.14a}$$

$$\begin{bmatrix} 0 & \underline{\Omega} + Y_1 \\ \underline{\Omega} + Y_1 & 2Y_2 \end{bmatrix} \succeq 0, \tag{3.14b}$$

$$\begin{bmatrix} R & 1 \\ 1 & \overline{\Omega} \end{bmatrix} \succeq 0, \tag{3.14c}$$

$$1 - R\overline{x}^2 + \sigma(\overline{x})(\overline{x}^2 - 1) - \epsilon \in \Sigma[\overline{x}]. \tag{3.14d}$$

Looking at the spectrum of the matrix in (3.14b), we conclude that (3.14b) is equivalent to $Y_2 > 0$ and $\underline{\Omega} + Y_1 = 0$. Condition (3.14c) is equivalently expressed as $R - \overline{\Omega}^{-1} \geq 0$. As for $\sigma(\overline{x})$, we set it to be an SOS polynomial of degree 0, hence, $\sigma(\overline{x}) = \overline{\sigma} \geq 0$. Writing the polynomial in (3.14d) as

$$1 - R\overline{x}^2 + \sigma(\overline{x})(\overline{x}^2 - 1) - \epsilon = \begin{bmatrix} 1 \\ \overline{x}^2 \end{bmatrix}^\top \begin{bmatrix} 1 - \overline{\sigma} - \epsilon & 0 \\ 0 & -R + \overline{\sigma} \end{bmatrix} \begin{bmatrix} 1 \\ \overline{x}^2 \end{bmatrix}$$

and bearing in mind Lemma 2.2.5, one realizes that condition (3.14d) is equivalent to $\bar{\sigma} - R \geq 0$ and $1 - \overline{\sigma} - \epsilon \geq 0$. In summary, we have the following conditions

$$\begin{aligned} &Y_2 > 0 \text{ and } Y_1 = -\underline{\Omega} > 0 \\ &\overline{\sigma} \geq R \geq \overline{\Omega}^{-1} > 0 \\ &\epsilon > 0 \text{ and } \overline{\sigma} + \epsilon \leq 1 \end{aligned}$$

We set $Y = [Y_1 \; Y_2] := [-\underline{\Omega} \; Y_2]$, with $\underline{\Omega}, Y_2$ any positive numbers, and $\overline{\Omega} > 1$, $R = \overline{\sigma} = \overline{\Omega}^{-1}$, $\epsilon \leq 1 - \overline{\sigma}$. Note that setting $\overline{\Omega} > 1$ is necessary for having the constraint $\overline{\sigma} + \epsilon \leq 1$ satisfied. To minimize the cost function, we should take $\overline{\Omega}$ as small as possible. We obtain that the function $b(x)$ that defines $\mathcal{B} := \{x \in \mathbb{R}^n : b(x) \geq 0\}$ is

$$b(x) = \overline{\Omega}^{-1}\overline{x}^2 - \underline{\Omega}^{-1}\underline{x}^2 - 1,$$

and the feedback controller is

$$u = Y\Omega^{-1} = [-\underline{\Omega}/\overline{\Omega} \; Y_2/\underline{\Omega}],$$

resulting in the closed-loop dynamics

$$\dot{x} = \begin{bmatrix} 0 & 1 \\ -\underline{\Omega}/\overline{\Omega} & Y_2/\underline{\Omega} \end{bmatrix} x$$

Figure 3.3 shows a sketch of the sets $\mathcal{S}^c$, $\mathcal{B}$. We first observe that $\overline{\Omega} > 1$ established above guarantees that $\mathcal{B} \subset \mathcal{S}$. Second, formulating $\mathcal{B}$ in terms of the entire state vector x results in a set $\mathcal{B}$ which differs from the one a designer could expect, namely, $\mathcal{B} := \{x \in \mathbb{R}^n : |\overline{x}| \geq \bar{b}\}$, with $\bar{b} > 1$.

For any feasible choice of the design parameters, the obtained closed-loop matrix has at least one unstable eigenvalue. To have an understanding of the state response, we compute the spectral representation $e^{(A+BK)t}$ for these values of the design parameters: $\underline{\Omega} = -4$, $\overline{\Omega} = 2$, $Y_2 = 4$. Then the spectral representation is given by

$$e^{(A+BK)t} = \frac{1}{3}\begin{bmatrix} e^{-2t} + 2e^{t} & -e^{-2t} + e^{t} \\ -2e^{-2t} + 2e^{t} & 2e^{-2t} + e^{t} \end{bmatrix}.$$

Hence, if the system starts from the initial condition $x = \begin{bmatrix}\overline{\Omega} & 0\end{bmatrix}^\top = [2\ 0]^\top$, which is on the boundary of $\mathcal{B}$, it will evolve as

$$x(t) = \frac{2}{3}\begin{bmatrix} e^{-2t} + 2e^{t} \\ -2e^{-2t} + 2e^{t} \end{bmatrix}.$$

As a result, both position and velocity diverge exponentially but are certified to stay within $\mathcal{B}$.

The CBF $b(x)$ constructed in the previous example is a function of the whole state $x = [\overline{x}, \underline{x}]^\top$. However, given the definition of $\mathcal{S}^c$, which only constrains the position variable $\overline{x}$, one could alternatively consider a candidate barrier function $\overline{b}(x) := (\overline{x} - \overline{c})^\top \overline{\Omega}^{-1}(\overline{x} - \overline{c}) - 1$ and the corresponding set $\overline{\mathcal{B}} := \{x \in \mathbb{R}^n : \overline{b}(x) \geq 0\}$. We will show below that the set $\overline{\mathcal{B}}$ can not be control invariant using linear feedback $u(x)$. Denote the projection matrix

$$\Pi := [I_{\overline{n}}\ \ 0_{\overline{n}\times\underline{n}}].$$

Use $\overline{x}_c$ instead of $\overline{x} - \overline{c}$, and let $\overline{x}_c = \Pi x_c$. We can derive the following identity

$$\dot{\overline{b}}(x) = x_c^\top(\Pi^\top\overline{\Omega}^{-1}\Pi(A + BK) + (A + BK)^\top\Pi^\top\overline{\Omega}^{-1}\Pi)x_c$$

and express the invariance condition as

$$\Pi^\top\overline{\Omega}^{-1}\Pi(A + BK) + (A + BK)^\top\Pi^\top\overline{\Omega}^{-1}\Pi \succeq 0.$$

We partition $A + BK$ according to the partition $\mathbb{R}^{\overline{n}} \times \mathbb{R}^{\underline{n}}$ to obtain

$$A + BK = \begin{bmatrix} \overline{A}_1 & \overline{A}_2 \\ \underline{A}_1 & \underline{A}_2 \end{bmatrix} + \begin{bmatrix} \overline{B} \\ \underline{B} \end{bmatrix}\begin{bmatrix} \overline{K} & \underline{K} \end{bmatrix}$$

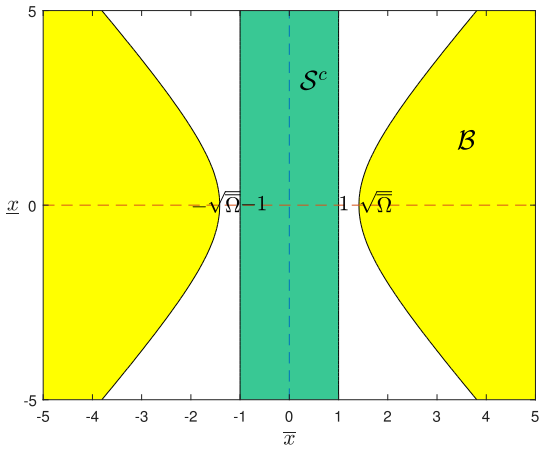


Figure 3.3: Pictorial illustration for Example 3.2.1. The green set $\mathcal{S}^c := \{x \in \mathbb{R}^2 : \overline{x}^2 - 1 \leq 0\}$ is expanded from a segment on $\mathbb{R}^1$. The designed control invariant set $\mathcal{B} := \{x \in \mathbb{R}^2 : \overline{\Omega}^{-1}\overline{x}^2 - \underline{\Omega}\underline{x}^2 - 1 \geq 0\}$ has been filled in yellow. Intuitively, with a large velocity, i.e. larger $|\underline{x}|$, the planar car should stay further away from the obstacle, which can be seen by the gap between $\mathcal{B}$ and $\mathcal{S}^c$ being larger for larger $|\overline{x}|$.

The invariance condition can be expressed as

$$\begin{bmatrix} \overline{\Omega}^{-1}(\overline{A}_1 + \overline{BK}) & \overline{\Omega}^{-1}(\overline{A}_2 + \overline{B}\underline{K}) \\ 0 & 0 \end{bmatrix} + \begin{bmatrix} (\overline{A}_1 + \overline{BK})^\top \overline{\Omega}^{-1} & 0 \\ (\overline{A}_2 + \overline{B}\underline{K})^\top \overline{\Omega}^{-1} & 0 \end{bmatrix} \succeq 0,$$

which leads to a convex condition by multiplying $\begin{bmatrix} \overline{\Omega} & 0 \\ 0 & I_n \end{bmatrix}$ on both sides of the matrices in the inequality. However, the possibility of fulfilling such constraint appears to be related to the possibility of shaping the spectra of $\overline{A}_1 + \overline{BK}$ and $\overline{A}_2 + \overline{B}\underline{K}$, hence, to the controllability of the pairs $(\overline{A}_1, \overline{B})$, $(\overline{A}_2, \overline{B})$. Going back to Example 3.2.1, we have $(\overline{A}_1, \overline{B}) = (0, 0)$, and $(\overline{A}_2, \overline{B}) = (1, 0)$, which shows lack of controllability of both pairs. As a result, the invariance condition above is

$$\begin{bmatrix} 0 & \overline{\Omega}^{-1} \\ \overline{\Omega}^{-1} & 0 \end{bmatrix} \succeq 0$$

which shows that enforcing invariance for the set $\overline{\mathcal{B}}$ via feedback is impossible due to the lack of controllability (the matrix has a positive and a negative eigenvalue).

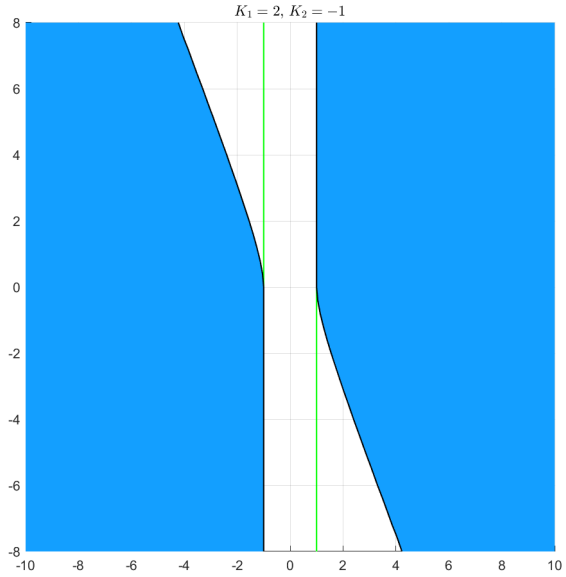


Figure 3.4: Exact invariant set for the planar car using $u = K_1\overline{x} + K_2\underline{x}$, where $K_1 = Y_1\overline{\Omega}^{-1} = 2, K_2 = Y_2\underline{\Omega}^{-1} = -1$. The green vertical lines are the boundary of $\mathcal{S}^c$, the set filled in blue is the exact invariant set.

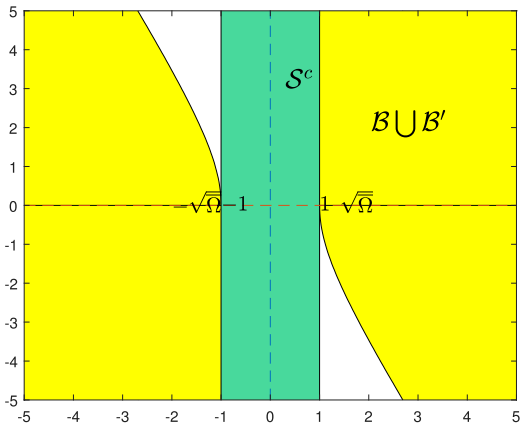


Figure 3.5: The union of control invariant set $\mathcal{B} \bigcup \mathcal{B}'$. $\mathcal{B}$ is computed by solving our program (3.4) which results in $\overline{\Omega} = 1$. Physical considerations for $\mathcal{B}'$ is obtained from the planar car. The union is also control invariant [70], and close to the exact invariant set as in Figure 3.4.

The exact control invariant set for Example 3.2.1 numerically computed by the level-set method toolbox [141] is shown in Figure 3.4. To compute the exact control invariant set, the control feedback u is set in the linear form $u = Y\Omega^{-1}x$, where Y and Ω have the same numerical values as those chosen in Example 3.2.1 ($\underline{\Omega} = -4$, $\overline{\Omega} = 2$, $Y_1 = 4$, $Y_2 = 4$). In comparison, our computed control invariant

set $\mathcal{B}$ determined analytically and depicted in Figure 3.3 is conservative when $\overline{\Omega} \neq 1$. This can be alleviated by minimizing $\mathrm{Tr}(\overline{\Omega})$ as in the program (3.4). Conservative behaviour is also encountered in the first and third quadrants, where the boundary of the exact invariant set coincides with the safe set $\mathcal{S}$. This is natural as the planar car is moving away from the unsafe set $\mathcal{S}^c$, which has been filled in green, in these regions. Our method, however, computes a control invariant set $\mathcal{B}$, that is symmetric with respect to the $\overline{x}$-axis. In practice, one can reduce this conservativeness by taking the union of our computed control invariant set $\mathcal{B}$ with other invariant sets, such as $\mathcal{B}' = \{x \in \mathbb{R}^2 : b'(x) := \underline{x}\overline{x} \geq 0\}$. The new control invariant set is shown in Figure 3.5. The control barrier function corresponds to this union set can be defined by

$$\beta(x) = \max\{b(x), b'(x)\}.$$

Such a kind of CBF has been investigated in [142].

According to Nagumo's Theorem [71], a compact set is invariant for a vector field if and only if the vector field is within the tangent cone for all points on the boundary of the set. For a closed set $\mathcal{B}$, this is equivalent to having $\dot{b}(x) \geq 0$, for any x such that $b(x) = 0$. However in our proposed convex conditions (3.4), we enforce a "strengthened" condition that $\dot{b}(x) \geq 0$, for any $x \in \mathbb{R}^n$. Nevertheless, we show in the following proposition that this does not introduce any conservativeness in the case $\overline{n} = n$.

Proposition 3.2.1. Consider the system (3.1), constant $c \in \mathbb{R}^n$ such that $\mathrm{rank}([B\ Ac]) = \mathrm{rank}(B)$ and a quadratic function $b(x) = (x-c)^\top \Omega^{-1}(x-c) - 1$, with $\Omega \succ 0$. If there exists a feedback controller $u = K(x-c) + d$, with d satisfying $Bd + Ac = 0$, such that $\dot{b}(x) \geq 0$ for any x such that $b(x) = 0$, then $\dot{b}(x) \geq 0$ for any $x \in \mathbb{R}^n$.

Proof. For $x = c$, $\dot{b}(x) = 2(x-c)^\top \Omega^{-1}(A + BK)(x-c) = 0$. On the other hand, observe that for any point $x \neq c \in \mathbb{R}^n$, there exists $y = \frac{1}{\lambda}(x-c) + c$ with

$\lambda = ((x-c)^\top \Omega^{-1}(x-c))^{1/2} > 0$, such that $b(y) = 0$. The function $\dot{b}(x)$ can be rewritten as

$$\dot{b}(x) := 2\lambda^2 \frac{(x-c)^\top}{\lambda} \Omega^{-1}(A+BK) \frac{(x-c)}{\lambda} = \lambda^2 \dot{b}(y)$$

As $b(y) = 0$, we have $\dot{b}(y) \geq 0$ by the proposition' s statement that assumes this is the case for y such that $b(y) = 0$, which implies $\dot{b}(x) \geq 0$, as claimed. □

As a result of Proposition 3.2.1, the synthesized controller $u(x)$ endows robustness as $\dot{b}(x) \geq 0$ for any x such that $b(x) < 0$. If the system starts from an unsafe point x, our synthesized CBF guarantees that there exists a controller that forces the state of the system enters the safe region, if the problem is feasible. This property is especially helpful for unexpected perturbations to the system.

Corollary 3.2.1. *Assume that the projection of $\mathcal{S}^c$ onto $\mathbb{R}^{\overline{n}}$ is a polytope on the space $\mathbb{R}^{\overline{n}}$ with vertices denoted by $v_1, \ldots, v_{o'} \in \mathbb{R}^{\overline{n}}$. Constraint* (3.4h) *can be replaced by linear constraints:*

$$-(v_i - \overline{c})^\top R(v_i - \overline{c}) + 1 \geq 0, i = 1, \ldots, o'. \tag{3.15}$$

Proof. Constraint (3.4h) implies $\mathcal{S}^c \subseteq \mathcal{R} := \{x \in \mathbb{R}^n : -(\overline{x} - \overline{c})^\top R(\overline{x} - \overline{c}) + 1 \geq 0\}$. Denote the projection set of $\mathcal{S}^c$ onto $\mathbb{R}^{\overline{n}}$ by $\overline{\mathcal{S}^c}$, which is a polytope, and the projection set of $\mathcal{R}$ onto $\mathbb{R}^{\overline{n}}$ by $\overline{\mathcal{R}}$, which is an ellipsoid. We then have $\mathcal{S}^c \subseteq \mathcal{R}$ is equivalent to $\overline{\mathcal{S}^c} \subseteq \overline{\mathcal{R}}$, which can be verified by the constraints of all the vertices of $\overline{\mathcal{S}^c}$ be within $\overline{\mathcal{R}}$. We conclude the proof. □

The number of linear constraints (3.15) depends on o'. If the polytopic $\mathcal{I}$ has l facets, then the maximum number of vertices is $\begin{pmatrix} o - \lceil n/2 \rceil \\ \lfloor n/2 \rfloor \end{pmatrix} + \begin{pmatrix} o - \lfloor n/2 \rfloor - 1 \\ \lceil n/2 \rceil - 1 \end{pmatrix}$, which could be quite large. For practical purposes, Corollary 3.2.1 becomes useful if the number of vertices is moderate.

3.2.2 Local Design

In the previous section, we construct a control invariant set $\mathcal{B} := \{x \in \mathbb{R}^n : b(x) \geq 0\}$ globally, it is unbounded on $\mathbb{R}^{\overline{n}}$, and naturally unbounded on $\mathbb{R}^n$. As shown in Example 3.2.1, the closed-loop trajectory diverges using the co-designed linear feedback controller $u(x)$. This is undesired in many applications where boundedness of trajectories is a prerequisite. In this section, we consider constructing a bounded control invariant set around a bounded set of initial conditions $\mathcal{I}$, and inside a intersection of half planes, i.e. the safe set $\mathcal{S}$. The new control invariant set will also be parameterized by a quadratic function. To ease notation, we still use $b(x) = x_c^\top \Omega^{-1} x_c - 1$, but the new control invariant set will be derived by a sub-level set of the function, i.e. $\mathcal{B}^c := \{x \in \mathbb{R}^n : b(x) \leq 0\}$, for boundness. The initial set is defined as an intersection of semi-algebraic sets:

$$\mathcal{I} := \bigcap_{i=1}^{l} \{x \in \mathbb{R}^n : w_i(x) \geq 0\}, \tag{3.16}$$

where $w_1(x), \ldots, w_l(x)$ are all polynomial functions.

Assumption 3.2.2. *$\mathcal{I}$ is a semi-algebraic set, and $\mathcal{I}$ is bounded on the space $\mathbb{R}^n$.*

The safe set is defined by

$$\mathcal{S} := \bigcap_{i=1}^{o} \{x \in \mathbb{R}^n : a_i^\top (x - c) + 1 \geq 0\}, \tag{3.17}$$

where $a_i \in \mathbb{R}^n$, $c \in \mathbb{R}^n$ is a point in the interior of the safe set. The following theorem proposes a convex condition for $b(x) = (x-c)^\top \Omega^{-1}(x-c) - 1$ to be a CBF for $((3.1), \mathcal{I}, \mathcal{S})$, with $\mathcal{B}^c = \{x \in \mathbb{R}^n : b(x) \leq 0\}$ a control invariant set. By $b(x)$ to be a CBF for $((3.1), \mathcal{I}, \mathcal{S})$ we mean that there exists $u(x)$ such that $\frac{\partial b(x)}{\partial x}(Ax + Bu(x)) \leq 0$ for all $x \in \partial\mathcal{B}^c$ and $\mathcal{I} \subseteq \mathcal{B}^c \subseteq \mathcal{S}$ (Figure 3.6). We again use $x_c = x - c$ for notational purpose.

Let $c \in \mathbb{R}^n$ be a constant vector such that $\text{rank}([B, Ac]) = \text{rank}(B)$ as before, and consider the following optimization program.

$$\max \text{Tr}(\Omega) \tag{3.18a}$$

$$\text{subject to } 0 \prec \Omega = \Omega^\top \in \mathbb{R}^n, \tag{3.18b}$$

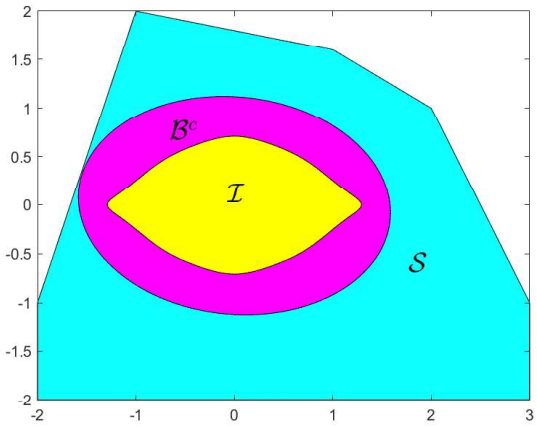


Figure 3.6: Geometric illustration of the the locally constructed CBF $b(x)$ on $\mathbb{R}^2$. The yellow set represents the initial set $\mathcal{I}$, which is bounded on $\mathbb{R}^2$. The blue set represents the safe set $\mathcal{S}$, which is defined by the intersection of half planes on $\mathbb{R}^2$, as in (3.17). The magenta set represents the control invariant set $\mathcal{B}^c := \{x \in \mathbb{R}^2 : b(x) \leq 0\}$, which satisfies $\mathcal{I} \subseteq \mathcal{B}^c \subseteq \mathcal{S}$.

$$0 \prec R = R^\top \in \mathbb{R}^{n\times n}, Y \in \mathbb{R}^{m\times n}, \tag{3.18c}$$

$$\sigma_1(x), \ldots, \sigma_l(x) \in \Sigma[x], \tag{3.18d}$$

$$\Omega A^\top + Y^\top B^\top + A\Omega + BY \preceq 0, \tag{3.18e}$$

$$\begin{bmatrix} R & I \\ I & \Omega \end{bmatrix} \succeq 0, \tag{3.18f}$$

$$-x_c^\top R x_c + 1 - \sum_{i=1}^{l} \sigma_i(x) w_i(x) \in \Sigma[x], \tag{3.18g}$$

$$1 - a_i^\top \Omega a_i \geq 0, i = 1, \ldots, o, \tag{3.18h}$$

where $x_c = x - c$. Similarly to program (3.4) for global design, program (3.18) is a convex optimization program, since the cost function is linear, and is subject to semi-definite and linear constraints. In the following theorem, we show how to synthesize a CBF $b(x)$ and a feedback safe controller $u(x)$ by this convex program under Assumption 3.2.2.

Theorem 3.2.2. *Consider Assumption 3.2.2, and let* $\mathcal{U} = \mathbb{R}^m$. *Assume that a solution to* (3.18) *exists and is denoted by* $\Omega, R, Y, \{\sigma_i(\cdot)\}_{i=1}^l$. *Set* $u(x) = Y\Omega^{-1}(x-c) + d$, *where* $d \in \mathbb{R}^m$ *is such that* $Bd + Ac = 0$. *We then have that*

1. $\mathcal{I} \subseteq \mathcal{B}^c \subseteq \mathcal{S}$, *where* $\mathcal{I}$ *is as in* (3.16), $\mathcal{B}^c = \{x \in \mathbb{R}^n : b(x) \leq 0\}$, $b(x) = (x-c)^\top \Omega^{-1}(x-c) - 1$ *and* $\mathcal{S}$ *is as in* (3.17).

2. $\mathcal{B}^c$ is a control invariant set for $\dot{x} = Ax + Bu(x)$.

Proof. The proof that (3.18e) is sufficient for $\mathcal{B}^c$ to be a control invariant set, and (3.18f) (3.18g) are sufficient for $\mathcal{I} \subseteq \mathcal{B}^c$ is similar to the proof of Theorem 3.2.1. We only prove that (3.18h) is sufficient and necessary for $\mathcal{B}^c \subseteq \mathcal{S}$. Using Farkas' lemma [143], [144, Lemma 6.45] for affine functions $a_i^\top(x-c)+1, i = 1, \ldots, o$, and convex quadratic function $(x-c)^\top \Omega^{-1}(x-c)-1$, we have that $\mathcal{B}^c \subseteq \mathcal{S}$ if and only if for every $i = 1, \ldots, o$, there exists $\lambda_i \geq \frac{1}{2}$ such that

$$2\lambda_i(a_i^\top x_c + 1) - (-x_c^\top \Omega^{-1} x_c + 1) \geq 0, \forall x \in \mathbb{R}^n,$$

which is equivalent to

$$\forall i = 1, \ldots, o, \exists \lambda_i \geq \frac{1}{2}, \text{s.t.} \begin{bmatrix} \Omega^{-1} & \lambda_i a_i \\ \lambda_i a_i^\top & 2\lambda_i - 1 \end{bmatrix} \succeq 0. \tag{3.19}$$

By Schur complement, (3.19) holds if and only if $\Omega \succ 0$, which is true by (3.18b), and if there exists $\lambda_i \geq \frac{1}{2}$ such that $2\lambda_i - 1 - \lambda^2 a_i^\top \Omega a_i \geq 0$. The discriminant of the quadratic polynomial on the left hand side of the inequality is $4 - 4a_i^\top \Omega a_i$. Hence, there exists λ_i such that $2\lambda_i - 1 - \lambda_i^2 a_i^\top \Omega a_i \geq 0$ if and only if $1 - a_i^\top \Omega a_i \geq 0$. Moreover, if $1 - a_i^\top \Omega a_i = 0$, then $\lambda_i = 1 \geq \frac{1}{2}$, and if $1 - a_i^\top \Omega a_i > 0$, then any $\lambda_i \in [1 - \sqrt{1 - a_i^\top \Omega a_i}, 1 + \sqrt{1 - a_i^\top \Omega a_i}]$ satisfies $2\lambda_i - 1 - \lambda_i^2 a_i^\top \Omega a_i \geq 0$. As $1 + \sqrt{1 - a_i^\top \Omega a_i} > \frac{1}{2}$, we have shown that there exists $\lambda_i \geq \frac{1}{2}$ such that $2\lambda_i - 1 - \lambda_i^2 a_i^\top \Omega a_i \geq 0$ if and only if $1 - a_i^\top \Omega a_i \geq 0$, which is (3.18h). Hence, we conclude the proof. □

3.2.3 Input Constraints

In the previous sections for local design we consider the case that $\mathcal{U} = \mathbb{R}^m$. We now extend the local design result to the case that the control authority is limited. Three different types of input constraints are considered: (i) 2-norm bounds, i.e., $\mathcal{U}_1 = \{u \in \mathbb{R}^m : ||u||_2^2 \leq \zeta\}$, where $\zeta > 0$; (ii) ∞-norm bounds, i.e., $\mathcal{U}_2 = \{u \in \mathbb{R}^m : ||u||_\infty \leq \sqrt{\zeta}\}$, where $\zeta > 0$; (iii) polytopic bounds, i.e., $\mathcal{U}_3 = \{u \in \mathbb{R}^m : Hu \leq h\}$, where $H \in \mathbb{R}^{k \times m}$, $h \in \mathbb{R}^k$.

For $\mathcal{U} = \mathcal{U}_1$, consider the following optimization program with decision variables $\Omega, R, Y, \sigma_1(x), \ldots, \sigma_l(x), \mu$:

$$\max \ \mathrm{Tr}(\overline{\Omega}) \tag{3.20a}$$

$$\text{subject to (3.18b)} - \text{(3.18h)}, -d^\top d + \zeta - \varepsilon > \mu > 0, \tag{3.20b}$$

$$\Pi = \begin{bmatrix} \Pi_{11} & \Pi_{12} \\ \Pi_{12}^\top & I_{n+m+1} \end{bmatrix} \succeq 0, \tag{3.20c}$$

where

$$\Pi_{11} = \begin{bmatrix} \Omega & Y^\top d & Y^\top \\ d^\top Y & \mu(-d^\top d + \zeta - \varepsilon) & 0 \\ Y & 0 & \mu \end{bmatrix},$$

$$\Pi_{12} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & \mu & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

and $\varepsilon > 0$ is a small constant. Program (3.20) is a convex program which amends program (3.18) by a new semi-definite constraint (3.20c).

Lemma 3.2.1. *Consider Assumption 3.2.2, and let $\mathcal{U} = \mathcal{U}_1$. Assume that a solution to (3.20) exists and is denoted by $\Omega, R, Y, \{\sigma_i(\cdot)\}_{i=1}^l, \mu$. Set $u(x) = Y\Omega^{-1}(x-c)+d$, where $d \in \mathbb{R}^n$ is such that $Bd + Ac = 0$. We then have that*

1. *$\mathcal{I} \subseteq \mathcal{B}^c \subseteq \mathcal{S}$, where $\mathcal{I}$ is as in (3.16), $\mathcal{B}^c = \{x \in \mathbb{R}^n : b(x) \leq 0\}$, $b(x) = (x-c)^\top \Omega^{-1}(x-c) - 1$ and $\mathcal{S}$ is as in (3.17).*
2. *$\mathcal{B}^c$ is a control invariant set for $\dot{x} = Ax + Bu(x)$ and $u(x) \in \mathcal{U}_1, \forall x \in \mathcal{B}^c$.*

Proof. By Theorem 3.2.2, we have that if $\Omega, R, Y, \{\sigma_i(\cdot)\}_{i=1}^o$ satisfy (3.18b)-(3.18h), then $\mathcal{B}^c$ is a control invariant set, and $\mathcal{I} \subseteq \mathcal{B}^c \subseteq \mathcal{S}$, and $u(x)$ is a safe controller. We only prove that (3.20c) is sufficient for $u(x) \in \mathcal{U}_1$, for all $x \in \mathcal{B}^c$. In condition (3.20c), $\Pi \succeq 0$ is equivalent to

$$\begin{aligned} &\Pi_{11} - \Pi_{12} I_{n+m+1} \Pi_{12}^\top \succeq 0 \\ \iff &\begin{bmatrix} \Omega & Y^\top d & Y^\top \\ d^\top Y & \mu(-d^\top d + \zeta - \varepsilon - \mu) & 0 \\ Y & 0 & \mu \end{bmatrix} \succeq 0 \end{aligned}$$

By Schur complement, if $\mu > 0$, then the last inequality is equivalent to

$$\begin{bmatrix} \Omega - \mu^{-1} Y^\top Y & Y^\top d \\ d^\top Y & \mu(-d^\top d + \zeta - \varepsilon - \mu) \end{bmatrix} \succeq 0. \tag{3.21}$$

Additionally, $-d^\top d + \zeta - \varepsilon - \mu > 0$, then the latter is equivalent to

$$\Omega - \mu^{-1} Y^\top Y - \frac{Y^\top d d^\top Y}{\mu(-d^\top d + \zeta - \varepsilon - \mu)} \succeq 0 \tag{3.22}$$

or

$$\mu\Omega - Y^\top Y - \frac{Y^\top d d^\top Y}{-d^\top d + \zeta - \varepsilon - \mu} \succeq 0. \tag{3.23}$$

The matrix remains positive semidefinite if we left- and right- multiply it by Ω^{-1}, thus we obtain (recall that $K = Y\Omega^{-1}$)

$$\begin{aligned}
&\Omega^{-1}\left(\mu\Omega - Y^\top\left(I + \frac{dd^\top}{-d^\top d + \zeta - \varepsilon - \mu}\right) Y\right)\Omega^{-1} \succeq 0,\\
&\iff \mu\Omega^{-1} - K^\top K - \frac{K^\top d d^\top K}{-dd^\top + \zeta - \varepsilon - \mu} \succeq 0,\\
&\iff \begin{bmatrix} -K^\top K + \mu\Omega^{-1} & -K^\top d \\ -d^\top K & -d^\top d + \zeta - \varepsilon - \mu \end{bmatrix} \succeq 0,\\
&\iff\\
&\begin{bmatrix} x_c \\ 1 \end{bmatrix}^\top \begin{bmatrix} -K^\top K + \mu\Omega^{-1} & -K^\top d \\ -d^\top K & -d^\top d + \zeta - \varepsilon - \mu \end{bmatrix} \begin{bmatrix} x_c \\ 1 \end{bmatrix} \geq 0,
\end{aligned}$$

for any x. Writing the product above explicitly, we obtain for any $x \in \mathbb{R}^n$:

$$\begin{aligned}
&- x_c^\top K^\top K x_c - d^\top K x_c - x_c^\top K^\top d - d^\top d + \zeta - \varepsilon\\
&+\mu\left(x_c^\top \Omega^{-1} x_c - 1\right) \geq 0,
\end{aligned}$$

which is

$$-\left(u(x)^\top u(x) - \zeta + \varepsilon\right) + \mu((x-c)^\top \Omega^{-1}(x-c) - 1) \geq 0.$$

Hence, for any x such that $b(x) = 0$, we have $x_c^\top \Omega^{-1} x_c - 1 \leq 0$, then $u(x)^\top u(x) \leq \zeta - \varepsilon \leq \zeta$. We conclude the proof. □

Condition (3.20c) is an LMI of dimension $2(n + m + 1)$. The dimension of the constraints is twice the equivalent condition $\Pi_{11} - \Pi_{12} I_{n+m+1} \Pi_{12}^\top \succeq 0$, which is however not an LMI due to the term μ^2. One tractable convex relaxation while maintaining a relatively lower dimension is

$$\begin{bmatrix} \Omega & Y^\top d & Y^\top \\ d^\top Y & \frac{(-d^\top d + \zeta - \varepsilon)^2}{2} & 0 \\ Y & 0 & (\frac{-d^\top d + \zeta - \varepsilon}{2}) I_m \end{bmatrix} \succeq 0. \tag{3.24}$$

Here μ takes the value of $\frac{-d^\top d+\zeta-\varepsilon}{2}$, which is the maximizer of $\mu(-d^\top d+\zeta-\varepsilon-\mu)$.

The non-negative tolerance ε is introduced for robustness.

Proposition 3.2.2. Given a CBF $b(x)$, system (3.1), and a control admissible set $\mathcal{U}_1$, for any x such that $b(x)=0$, there exists $\delta(x)>0$, such that for any $x'\in\mathcal{E}(x,\delta(x))$, $u(x')=Y\Omega^{-1}(x'-c)+d\in\mathcal{U}_1$.

Proof. Given that $u(x)$ is a continuous function, $||u(x)||_2^2$ is also a continuous function. Therefore, for any $x\in\partial\mathcal{B}^c$, there exists $\xi(x)>0$, such that for any $y\in\mathcal{E}(x,\xi(x))$, $||u(y)||_2^2-||u(x)||_2^2\leq\frac{\varepsilon}{2}$. From Lemma 3.2.1 we have that $||u(x)||_2^2\leq\zeta-\varepsilon$, thus $||u(y)||_2^2\leq\zeta-\frac{\varepsilon}{2}$. Pick $0<\delta(x)\leq\zeta$, we have that for any $x'\in\mathcal{E}(x,\delta(x))$, $||u(x')||\leq\zeta-\frac{\varepsilon}{2}$. Hence, $u(x')\in\mathcal{U}_1$, and we conclude the proof. □

We then deal with the case that $\mathcal{U}=\mathcal{U}_2$. Consider the following optimization program with decision variables $\Omega,R,Y,\sigma_1(x),\ldots,\sigma_l(x),\mu_1,\ldots,\mu_m$.

$$\min\ \mathrm{Tr}(\overline{\Omega}) \tag{3.25a}$$

$$\text{subject to (3.18b)} - \text{(3.18h)}, -d^\top d+\zeta-\varepsilon>\mu_i>0, \tag{3.25b}$$

$$\Pi^i=\begin{bmatrix}\Pi_{11}^i & \Pi_{12}\\ \Pi_{12}^\top & I_{n+m+1}\end{bmatrix}\succeq 0, i=1,\ldots,m, \tag{3.25c}$$

where

$$\Pi_{11}^i=\begin{bmatrix}\Omega & Y^\top O_i^\top d & Y^\top O_i^\top\\ d^\top O_iY & \mu_i(-d^\top d+\zeta-\varepsilon) & 0\\ O_iY & 0 & \mu_i\end{bmatrix},$$

$$\Pi_{12}=\begin{bmatrix}0&0&0\\0&\mu_i&0\\0&0&0\end{bmatrix},$$

$O_i\in\mathbb{R}^{m\times m}$ is an all-zero matrix, with the i-th diagonal entry is one, $\varepsilon>0$ is a small constant. Program (3.25) is a convex program which amends program (3.18) by a new semi-definite constraint (3.25c).

Lemma 3.2.2. *Consider Assumption 3.2.2, and let $\mathcal{U}=\mathcal{U}_2$. Assume that a solution to (3.25) exists and is denoted by $\Omega,R,Y,\{\sigma_i(\cdot)\}_{i=1}^l,\mu$. Set $u(x)=Y\Omega^{-1}(x-c)+d$, where $d\in\mathbb{R}^n$ is such that $Bd+Ac=0$. We then have that*

1. $\mathcal{I} \subseteq \mathcal{B}^c \subseteq \mathcal{S}$, *where* $\mathcal{I}$ *is as in* (3.16), $\mathcal{B}^c = \{x \in \mathbb{R}^n : b(x) \leq 0\}$, $b(x) = (x-c)^\top \Omega^{-1}(x-c) - 1$ *and* $\mathcal{S}$ *is as in* (3.17).

2. $\mathcal{B}^c$ *is a control invariant set for* $\dot{x} = Ax + Bu(x)$ *and* $u(x) \in \mathcal{U}_2$, $\forall x \in \mathcal{B}^c$.

Proof. Similarly to the proof of Lemma 3.2.1, we prove that (3.25c) is sufficient for $u(x) \in \mathcal{U}_2, \forall x \in \mathcal{B}^c$. (3.25c) is equivalent to

$$\begin{aligned} &- x_c^\top K^\top O_i^\top O_i K x_c - d^\top O_i K x_c - x_c^\top K^\top O_i^\top d - d^\top d + \zeta \\ &+\mu_i \left(x_c^\top \Omega^{-1} x_c - 1 \right) - \varepsilon \geq 0, i = 1, \ldots, m, \end{aligned}$$

which is

$$\begin{aligned} - \left(u_i(x)^\top u_i(x) - \zeta - \varepsilon \right) + \mu_i((x-c)^\top \Omega^{-1}(x-c) - 1) \geq 0, \\ i = 1, \ldots, m. \end{aligned}$$

Then we have $u_i(x)^\top u_i(x) \leq \zeta - \varepsilon \leq \zeta, i = 1, \ldots, m$, for any x such that $b(x) = 0$. Therefore, $||u(x)||_\infty \leq \sqrt{\zeta}$. We conclude the proof. □

We then deal with the case that $\mathcal{U} = \mathcal{U}_3 = \{u \in \mathbb{R}^m : Hu \leq h\}$. Consider the following optimization program

$$\min \ \mathrm{Tr}(\overline{\Omega}) \tag{3.26a}$$

$$\text{subject to (3.18b)} - \text{(3.18h)}, \mu > 0, \tag{3.26b}$$

$$\begin{bmatrix} \Xi_{11}^i & \Xi_{12} \\ \Xi_{12}^\top & I_{n+1} \end{bmatrix} \succeq 0, i = 1, \ldots, k, \tag{3.26c}$$

where

$$\Xi_{11}^i = \begin{bmatrix} \Omega & Y^\top H_i^\top \\ H_i Y & \mu(-2H_i d + 2h_i - \varepsilon), \end{bmatrix} \Xi_{12} = \begin{bmatrix} 0 & 0 \\ 0 & \mu \end{bmatrix},$$

$\varepsilon > 0$ is a small constant. Program (3.26) is a convex program which amends program (3.18) by a new semi-definite constraint (3.26c).

Lemma 3.2.3. *Consider Assumption 3.2.2, and let* $\mathcal{U} = \mathcal{U}_3$. *Assume that a solution to* (3.26) *exists and is denoted by* $\Omega, R, Y, \{\sigma_i(\cdot)\}_{i=1}^l, \mu$. *Set* $u(x) = Y\Omega^{-1}(x-c) + d$, *where* $d \in \mathbb{R}^n$ *is such that* $Bd + Ac = 0$. *We then have that*

1. $\mathcal{I} \subseteq \mathcal{B}^c \subseteq \mathcal{S}$, *where* $\mathcal{I}$ *is as in* (3.16), $\mathcal{B}^c = \{x \in \mathbb{R}^n : b(x) \leq 0\}$, $b(x) = (x-c)^\top \Omega^{-1}(x-c) - 1$ *and* $\mathcal{S}$ *is as in* (3.17).

2. $\mathcal{B}^c$ *is a control invariant set for* $\dot{x} = Ax + Bu(x)$ *and* $u(x) \in \mathcal{U}_3$, $\forall x \in \mathcal{B}^c$.

Proof. Similarly to the proof of Lemma 3.2.1, and 3.2.2, we prove that (3.26c) is sufficient for $u(x) \in \mathcal{U}_3, \forall x \in \mathcal{B}$. (3.26c) is equivalent to

$$\Xi_{11}^i - \Xi_{12} I_{n+1} \Xi_{12}^\top \succeq 0, i = 1, \ldots, k,$$

which implies

$$\Omega - \frac{Y^\top H_i^\top Y H_i}{\mu(-2H_i d + 2h_i - \varepsilon - \mu)} \succeq 0, i = 1, \ldots, k.$$

Therefore, we have

$$\mu\Omega - \frac{Y^\top H_i^\top H_i Y}{-2H_i d + 2h_i - \varepsilon - \mu} \succeq 0.$$

The matrix remains positive semidefinite if we left- and right- multiply it by Ω^{-1}, thus we obtain

$$\begin{aligned} & \mu\Omega^{-1} - \frac{K^\top H_i^\top H_i K}{-2H_i d + 2h_i - \varepsilon - \mu} \succeq 0 \\ \iff & \begin{bmatrix} \mu\Omega^{-1} & -K^\top H_i^\top \\ -H_i K & -2H_i d + 2h_i - \varepsilon - \mu \end{bmatrix} \succeq 0 \\ \iff & -2(H_i(Kx_c + d) - h_i) + \mu(x_c^\top \Omega^{-1} x_c - 1) - \varepsilon \geq 0. \end{aligned}$$

Then we have for every $i = 1, \ldots, k$, $H_i u_i(x) \leq h_i - \varepsilon < h_i$ for any x such that $b(x) = (x - c)^\top \Omega^{-1}(x - c) - 1 \leq 0$, $Hu(x) \leq h$. We conclude the proof. □

Similar to the design in Lemma 3.2.1, ε is also introduced in Lemma 3.2.2 and 3.2.3. As a consequence, a robustness property is imposed on the synthesized CBF as in Proposition 3.2.2.

3.2.4 Additive Perturbation

We now consider robustness to an additive noise. Consider the linear system (3.1) but with a perturbation δ:

$$\dot{x} = Ax + Bu + \delta, \tag{3.27}$$

where $\delta(t) \in \mathbb{R}^n$ is unknown but satisfies $\delta^\top \delta \leq D$ for all $t \in \mathbb{R}_{\geq 0}$ for some known $D > 0$. We modify the synthesis programs (3.4), and (3.18) into *robust versions* as follows.

$$\max \ \mathrm{Tr}(\overline{\Omega}) \tag{3.28a}$$

$$\text{subject to (3.4b)} - \text{(3.4h)}, \mu \geq 0, \lambda \in \mathbb{R}, \tag{3.28b}$$

$$H_1 \succeq 0, \tag{3.28c}$$

where

$$H_1 = \begin{bmatrix} \Omega A^\top + A\Omega + Y^\top B^\top + BY - \lambda\Omega & \Omega & 0 \\ \Omega & \mu I_n & 0 \\ 0 & 0 & -\mu D + \lambda \end{bmatrix}.$$

$$\min \ \mathrm{Tr}(\overline{\Omega}) \tag{3.29a}$$

$$\text{subject to (3.18b)} - \text{(3.18h)}, \mu \geq 0, \lambda \in \mathbb{R}, \tag{3.29b}$$

$$H_2 \succeq 0, \tag{3.29c}$$

where

$$H_2 = \begin{bmatrix} -\Omega A^\top - A\Omega - Y^\top B^\top - BY + \lambda\Omega & \Omega & 0 \\ \Omega & \mu I_n & 0 \\ 0 & 0 & -\mu D + \lambda \end{bmatrix}.$$

Lemma 3.2.4. *Let Assumption 3.2.1 hold. Suppose $\mathcal{U} = \mathbb{R}^m$. Consider system (3.27), safe set $\mathcal{S}$ as in (3.3), constant $c = [\overline{c}^\top, \underline{c}^\top]^\top \in \mathbb{R}^n$, such that rank([B Ac])=rank(B) and δ such that $\delta^\top \delta \leq D$. Let d be a vector satisfying $Bd + Ac = 0$. Assume that a solution to (3.28) exists and is denoted by $\Omega, R, Y, \{\sigma_i(\cdot)\}_{i=1}^o, \mu \geq 0, \lambda \in \mathbb{R}$. Let $u(x) = Y(x)\Omega^{-1}x_c + d$, we then have that*

1. *$\mathcal{B} \subseteq \mathcal{S}$, where $\mathcal{B} = \{x \in \mathbb{R}^n : b(x) \geq 0\}$, $b(x) = x_c^\top \Omega^{-1} x_c - 1$.*
2. *$\mathcal{B}$ is a control invariant set for $\dot{x} = Ax + Bu(x) + \delta$, $\forall \delta \in \{\delta : \delta^\top \delta \leq D\}$.*

Proof. Bearing in mind the proof of Theorem 3.2.1, to prove this statement it is enough to show that $H_1 \succeq 0$ implies $\dot{b}(x) := 2x_c^\top \Omega^{-1}[(A+BK)(x-c)+\delta] \geq 0$ for all $x \in \mathcal{B}$ and for all δ such that $\delta^\top \delta \leq D$. To this end, note that

$$\dot{b}(x) = \begin{bmatrix} x_c \\ \delta \\ 1 \end{bmatrix}^\top \begin{bmatrix} \Omega^{-1}(A+BK) + (A+BK)^\top \Omega^{-1} & \Omega^{-1} & 0 \\ \Omega^{-1} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_c \\ \delta \\ 1 \end{bmatrix}$$

Hence, to enforce $\dot{b}(x) \geq 0$ for all $x \in \mathcal{B}$, *i.e.,* for all x such that

$$\begin{bmatrix} x_c \\ \delta \\ 1 \end{bmatrix} \begin{bmatrix} \Omega^{-1} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} x_c \\ \delta \\ 1 \end{bmatrix} \geq 0$$

and for all δ such that $\delta^\top \delta \leq D$, *i.e.,* for all δ such that

$$\begin{bmatrix} x_c \\ \delta \\ 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & -I_n & 0 \\ 0 & 0 & D \end{bmatrix} \begin{bmatrix} x_c \\ \delta \\ 1 \end{bmatrix} \geq 0.$$

It is sufficient that there exists λ, $\mu \geq 0$ such that

$$\begin{bmatrix} \Omega^{-1}(A+BK) + (A+BK)^\top \Omega^{-1} - \lambda \Omega^{-1} & \Omega^{-1} & 0 \\ \Omega^{-1} & \mu I_n & 0 \\ 0 & 0 & \lambda - \mu D \end{bmatrix} \succeq 0.$$

This condition is equivalent to $H_1 \succeq 0$ provided that $Y = K\Omega$. This ends the proof. □

Lemma 3.2.5. *Let Assumption 3.2.2 hold. Suppose $\mathcal{U} = \mathbb{R}^m$. Consider system* (3.27)*, safe set $\mathcal{S}$ as in* (3.17)*, initial set $\mathcal{I}$ as in* (3.16)*, constant $c = [\overline{c}^\top, \underline{c}^\top]^\top \in \mathbb{R}^n$, such that rank([B Ac])=rank(B) and δ such that $\delta^\top \delta \leq D$. Let d be a vector satisfying $Bd + Ac = 0$. Assume that a solution to* (3.29) *exists and is denoted by $\Omega, R, Y, \{\sigma_i(\cdot)\}_{i=1}^l, \mu \geq 0, \lambda \in \mathbb{R}$. Let $u(x) = Y(x)\Omega^{-1}x_c + d$, we then have that*

1. *$\mathcal{I} \subseteq \mathcal{B}^c \subseteq \mathcal{S}$, where $\mathcal{B}^c = \{x \in \mathbb{R}^n : b(x) \leq 0\}$, $b(x) = x_c^\top \Omega^{-1} x_c - 1$.*
2. *$\mathcal{B}^c$ is a control invariant set for $\dot{x} = Ax + Bu(x) + \delta, \forall \delta^\top \delta \leq D$.*

Proof. The proof is similar to that for Lemma 3.2.4. □

Note here there is a bilinearity term $\lambda\Omega$ in both H_1 and H_2. One tractable way to overcome the bilinearity is to do a line search for the multiplier λ.

3.2.5 Model Uncertainty

Another case that is worth mentioning is model uncertainty. Consider a linear system with an uncertain state matrix and input matrix:

$$\dot{x} = A(\delta)x + B(\delta)u, \tag{3.30}$$

where δ is an uncertain parameter. Uncertainty is commonly encountered in data-driven controller design [145], where the uncertain model is the result of a representation obtained from noise-corrupted data. In fact, under an assumption on sufficiently rich data (cf., [145, Assumption 1]), the matrix $[B(\delta)\ A(\delta)]$ belongs to the set

$$S(\delta) : \overline{\boldsymbol{C}} + S(\delta)\overline{\boldsymbol{B}} + \overline{\boldsymbol{B}}^\top S(\delta)^\top + S(\delta)\overline{\boldsymbol{A}}S(\delta)^\top \preceq 0. \tag{3.31}$$

Here, $\overline{\boldsymbol{A}} \succ 0$ is a symmetric matrix, $\overline{\boldsymbol{C}}$ satisfies $\overline{\boldsymbol{C}} = \overline{\boldsymbol{B}}^\top \overline{\boldsymbol{A}}^{-1} \overline{\boldsymbol{B}}^\top - \delta I$. $\overline{\boldsymbol{A}}$ and $\overline{\boldsymbol{B}}$ are the solution of a semi-definite program [146, Eq. (13)]. Intuitively, the set (3.31) is the over approximation of the union of identified sets. Each of these sets is bounded provided that the measured noise is bounded. We provide the following program to synthesize a control barrier function $b(x)$ and a safe controller for system (3.30), both globally and locally. The global program is given by

$$\min \ \mathrm{Tr}(\overline{\Omega}) \tag{3.32a}$$

$$\text{subject to (3.4b)} - \text{(3.4h)}, \tag{3.32b}$$

$$\begin{bmatrix} \overline{\boldsymbol{C}} & \begin{bmatrix} Y \\ \Omega \end{bmatrix}^\top - \overline{\boldsymbol{B}}^\top \\ \begin{bmatrix} Y \\ \Omega \end{bmatrix} - \overline{\boldsymbol{B}} & \overline{\boldsymbol{A}}^\top \end{bmatrix} \succeq 0. \tag{3.32c}$$

The local program is given by

$$\max \ \mathrm{Tr}(\overline{\Omega}) \tag{3.33a}$$

$$\text{subject to (3.18b)} - \text{(3.18h)}, \tag{3.33b}$$

$$\begin{bmatrix} \overline{\boldsymbol{C}} & -\begin{bmatrix} Y \\ \Omega \end{bmatrix}^\top - \overline{\boldsymbol{B}}^\top \\ -\begin{bmatrix} Y \\ \Omega \end{bmatrix} - \overline{\boldsymbol{B}} & \overline{\boldsymbol{A}}^\top \end{bmatrix} \succeq 0. \tag{3.33c}$$

The two programs are convex, since (3.32c) and (3.33c) are both semi-definite constraints.

Lemma 3.2.6. *Let Assumption 3.2.1 hold, suppose that* $\mathcal{U} = \mathbb{R}^m$. *Consider system* (3.30), *with matrices* $A(\delta)$ *and* $B(\delta)$ *unknown but belonging to the set* (3.31), *safe set* $\mathcal{S}$ *as in* (3.3), *and a constant* $c = [\overline{c}^\top \underline{c}^\top]^\top \in \mathbb{R}^n$, *such that rank([*$B$ Ac*]) = rank(*B*). Let* d *be a vector satisfying* $Bd + Ac = 0$. *Assume that a solution to* (3.32) *exists and is denoted by* $\Omega, R, Y, \{\sigma_i(\cdot)\}_{i=1}^o$. *Let* $u(x) = Y(x)\Omega^{-1}x_c$, *we then have that*

1. $\mathcal{B} \subseteq \mathcal{S}$, *where* $\mathcal{B} = \{x \in \mathbb{R}^n : b(x) \geq 0\}$, $b(x) = x_c^\top \Omega^{-1} x_c - 1$.

2. $\mathcal{B}$ *is a control invariant set for* $\dot{x} = A(\delta)x + B(\delta)u(x)$.

Proof. A straightforward modification of the proof of [145, Theorem 2] shows that (3.32c) implies that $\dot{b}(x) = (x-c)^\top(\Omega^{-1}(A+BK)+(A+BK)^\top\Omega^{-1})(x-c) \geq 0$, for all $x \in \mathbb{R}^n$, and all (A, B) that belong to the set (3.31). As the matrices of system (3.30) belong to the set (3.31), the claim above shows the invariance property of $\mathcal{B}$ for the closed-loop system. The rest of the proof goes like the proof of Theorem 3.2.1. □

Lemma 3.2.7. *Let Assumption 3.2.2 hold, suppose that* $\mathcal{U} = \mathbb{R}^m$. *Consider system* (3.30), *with matrices* $A(\delta)$ *and* $B(\delta)$ *unknown but belonging to the set* (3.31), *safe set* $\mathcal{S}$ *as in* (3.17), *initial set* $\mathcal{I}$ *as in* (3.16), *and a constant* $c = [\overline{c}^\top \underline{c}^\top]^\top \in \mathbb{R}^n$, *such that rank([*$B$ Ac*]) = rank(*B*). Let* d *be a vector satisfying* $Bd + Ac = 0$. *Assume that a solution to* (3.33) *exists and is denoted by* $\Omega, R, Y, \{\sigma_i(\cdot)\}_{i=1}^o$. *Let* $u(x) = Y(x)\Omega^{-1}x_c$, *we then have that*

1. $\mathcal{I} \subseteq \mathcal{B}^c \subseteq \mathcal{S}$, *where* $\mathcal{B}^c = \{x \in \mathbb{R}^n : b(x) \leq 0\}$, $b(x) = x_c^\top \Omega^{-1} x_c - 1$.

2. $\mathcal{B}^c$ *is a control invariant set for* $\dot{x} = A(\delta)x + B(\delta)u(x)$.

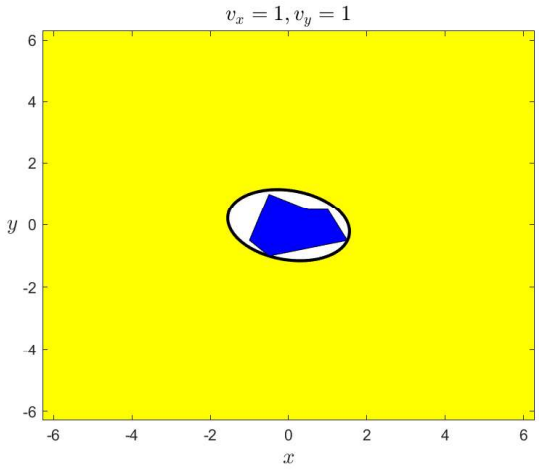


Figure 3.7: The collision space $\mathcal{S}^c$ is filled by dark blue. Velocity is fixed to be $v_x = 1$, $v_y = 1$. The control invariant set $\mathcal{B} := \{x : b(x) \geq 0\}$ is designed by solving the global convex program (3.4), and is filled in yellow.

3.3 Simulation Results

In this section we demonstrate the proposed programs on a linear system with a high relative degree. All the examples are coded using MATLAB R2022a, SOSTOOLS-4.03 [92], and SeDuMi-1.3.7 [95]. In this example, we show how to design CBFs for a linear system with a relative degree. Both the global design and the local design will be conducted. Consider an omni-directional vehicle and a collision avoidance problem. The dynamics of the vehicle are

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{v}_x \\ \dot{v}_y \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ v_x \\ v_y \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} a_x \\ a_y \end{bmatrix}, \tag{3.34}$$

where $[x, y]$ represents the position of the vehicle on the 2-D plane, and $[v_x, v_y]$ represents the corresponding velocity. The vehicle is controlled by tuning the acceleration denoted by $u = [a_x, a_y]$ along the two directions. The position corresponds to $\overline{x}$, while the velocity corresponds to $\underline{x}$ in (3.4). A polytopic obstacle (with five facets) is placed with $\overline{c} = [0, 0]^\top$ be an inside point. Under this configuration, the

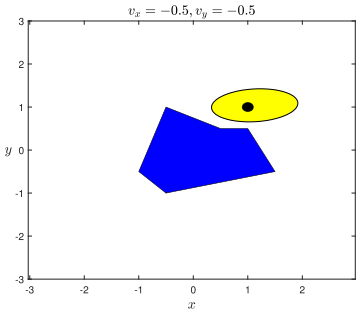
safe set is a semi-algebraic set, which can be formulated as

$$\mathcal{S} := \bigcup_{i=1}^{5} \left\{ \begin{bmatrix} x \\ y \\ v_x \\ v_y \end{bmatrix} \in \mathbb{R}^4 : a_i^\top \left(\begin{bmatrix} x \\ y \end{bmatrix} - \bar{c} \right) + 1 \geq 0 \right\}.$$

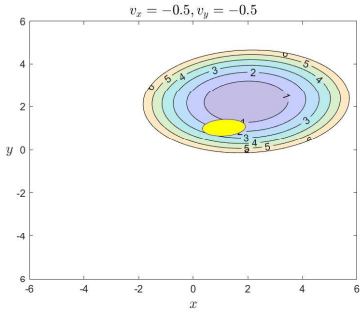
where $a_i \in \mathbb{R}^2$, $i = 1, \ldots, 5$, are known vectors. The collision space $\mathcal{S}^c$ is then a bounded polytope contains $c = [0, 0, 0, 0]^\top$. Given that $\mathcal{S}$ is only defined over $[x, y]$, we consider to design a CBF

$$b(x) = \begin{bmatrix} x \\ y \end{bmatrix}^\top \overline{\Omega}^{-1} \begin{bmatrix} x \\ y \end{bmatrix} - \begin{bmatrix} v_x \\ v_y \end{bmatrix}^\top \underline{\Omega}^{-1} \begin{bmatrix} v_x \\ v_y \end{bmatrix} - 1$$

by solving (3.4). We obtain a control barrier function as $b(x) = 2.4104x^2 - 0.67042xy + 1.3229y^2 - 859.4863v_x^2 - 859.4863v_y^2 - 1$ and a control gain as $K = \begin{bmatrix} 369.6 & 93.6 & -0.5 & 0 \\ 93.6 & 673.4 & 0 & -0.5 \end{bmatrix}$. We visualize the control invariant set $\mathcal{B}$ and the obstacle $\mathcal{S}^c$ on $\mathbb{R}^2$ by fixing $v_x = 1$, $v_y = 1$. The result is shown in Figure 3.7.



(a) Blue region is the obstacle $\mathcal{S}^c$, yellow region is $\mathcal{B}^c$ and black region is the initial set $\mathcal{I}$.



(b) Level sets of $||u(x)||_2^2$. $||u(x)||_2^2 \leq 4$ for any $x \in \mathcal{B}^c$, thus showing that the input constraint is not violated.

Figure 3.8: Convex invariant set $\mathcal{B}^c$ and state feedback controller $u(x)$ designed by solving the local program (3.18). The sets are projected to $\mathbb{R}^2$ by setting $v_x = -0.5, v_y = -0.5$. The yellow region is $\mathcal{B}^c$.

Then we consider a local design. The car is starting from the initial set

$$\mathcal{I} := \left\{ \begin{bmatrix} x \\ y \\ v_x \\ v_y \end{bmatrix} \in \mathbb{R}^4 : (x-1)^2 + (y-1)^2 \leq 0.01 \right.$$

$$\left. (v_x + 0.5)^2 \leq 0.1, (v_y + 0.5)^2 \leq 0.1 \right\}.$$

The acceleration limits are encoded by $a_x^2 + a_y^2 \leq 4$. By solving the local design program (3.18), we obtain $b(x) = 0.66903v_x^2 - 0.44567v_xv_y + 0.28291v_xx - 0.80024v_xy + 1.1024v_y^2 + 0.23651v_yx + 1.4055v_yy + 1.1198x^2 - 0.58818xy + 4.7544y^2 - 1$ and control gain $K = \begin{bmatrix} -1.18 & 0.41 & -0.64 & 0.21 \\ -0.1 & -2.35 & -0.09 & -1 \end{bmatrix}$. The designed control invariant set is $\mathcal{B}^c := \left\{ \begin{bmatrix} x \\ y \\ v_x \\ v_y \end{bmatrix} \in \mathbb{R}^4 : b(x) \leq 0 \right\}$, and level sets of $||u(x)||_2^2$ are visualized in Figure 3.8.

3.4 Conclusion

In this chapter we proposed a method to synthesize a control barrier function and a state feedback controller by solving a single convex program. Our approach considers quadratic control barrier functions and affine state feedback controllers. Different types of control input limits can be handled as additional convex constraints to the synthesis program. We demonstrate the efficacy of our approach on an omni-directional car collision avoidance problem.

However, the results are only applied to linear dynamical systems. We will extend the results to general nonlinear dynamics in the next Chapter.

4

Convex Co-Design of Control Barrier Functions and Feedback Controllers for Nonlinear Systems

In this chapter, we extend the result in Chapter 3 to nonlinear systems. We first show convex co-design methods that appropriately parameterizing a CBF and a feedback controller, and co-optimizes over their parametrization. To further reduce conservatism, we propose an iterative algorithm to solve a nonconvex co-design program that admits more flexible parameterization.

4.1 Convex Design for Nonlinear Systems

In this section we propose extensions of our result to polynomial systems, and more general non-polynomial systems. The nonlinear system considered in this section will be in the input-affine form of

$$\dot{x} = A(x)F(x) + B(x)u, \tag{4.1}$$

where $x \in \mathbb{R}^n$, $u \in \mathbb{R}^m$. $A(x) : \mathbb{R}^n \to \mathbb{R}^{n\times k}$, $B(x) : \mathbb{R}^n \to \mathbb{R}^{n\times m}$ are state and input matrices, respectively. $F(x) : \mathbb{R}^n \to \mathbb{R}^k$ is a vector-valued continuous function, which satisfies $F(0) = 0$. The drift vector field is formulated be a product of

two functions $A(x)$ and $F(x)$, which this gives the designers freedom for different selections. The initial set $\mathcal{I}$ and safe set $\mathcal{S}$ are analogous to Section 3.2 in Chapter 3.

The CBF and feedback controller $u(x)$ are parameterized as

$$\begin{aligned} b(x) =& F(x)^\top \Omega^{-1} F(x) - 1, \\ u(x) =& Y(x)\Omega^{-1} F(x), \end{aligned} \tag{4.2}$$

where $Y(x) \in \mathbb{R}[x]^{m\times n}$ $\Omega \in \mathbb{R}^{n\times n}$ are matrices to be designed. For the special case that $F(x) = x$, the CBF for nonlinear systems is a quadratic function. The nonlinear function $F(x)$ gives higher flexibility in designing a CBF. This is especially important for nonlinear systems.

4.1.1 Nonlinear Polynomial Systems

We first focus on the case that $F(x)$, $A(x)$ and $B(x)$ are all polynomial matrices, i.e. every element of these matrices is a polynomial in x. We assume that $F(0) = 0$, and $c = 0$. Here the selection of c is not as flexible as that for linear systems in Section 3.2 of Chapter 3 This is because transforming the coordinate for nonlinear systems is generally hard. However, as we will show in the example of Section 4.3.1, c can be set to a non-zero constant for a nonlinear platooning system.

As a polynomial function, $F(x)$ is differentiable in every element. The Jacobian matrix $M(x) : \mathbb{R}^n \to \mathbb{R}^{k\times n}$ is determined by

$$M_{ij}(x) = \frac{\partial F_i(x)}{\partial x_j}.$$

The designed controller $u(x)$ takes the form

$$u(x) = K(x)F(x), \tag{4.3}$$

where $K(x) : \mathbb{R}^n \to \mathbb{R}^{m\times k}$ is a matrix valued polynomial function.

Consider global design where the safe set $\mathcal{S}$ follows Assumption 3.2.1 in Chapter 3. For this case, $F(x)$ is designed by

$$F(x) = [\overline{F}(\overline{x})^\top, \underline{F}(\underline{x})]^\top, \tag{4.4}$$

where $\overline{F}(\overline{x}) : \mathbb{R}^{\overline{n}} \to \mathbb{R}^{\overline{k}}$, $\underline{F}(\underline{x}) : \mathbb{R}^{\underline{n}} \to \mathbb{R}^{\underline{k}}$, and $\overline{k} + \underline{k} = k$. The program for global design is presented as follows.

$$\min \operatorname{Tr}(\overline{\Omega}) \tag{4.5a}$$

$$\text{subject to } 0 \prec \overline{\Omega} \in \mathbb{R}^{\overline{k}\times\overline{k}}, 0 \succ \underline{\Omega} \in \mathbb{R}^{\underline{k}\times\underline{k}}, 0 \prec R \in \mathbb{R}^{\overline{k}\times\overline{k}}, \tag{4.5b}$$

$$Y(x) \in \mathbb{R}^{m\times n}, \sigma_1(x), \ldots, \sigma_o(x) \in \Sigma[x], \epsilon > 0, \tag{4.5c}$$

$$\Omega = \begin{bmatrix} \overline{\Omega} & \\ & \underline{\Omega} \end{bmatrix}, \tag{4.5d}$$

$$G(x) = \Omega A(x)^\top M(x)^\top + M(x)B(x)Y(x)+$$

$$M(x)A(x)\Omega + M(x)B(x)Y(x) \in \Sigma[x]^{n\times n}, \tag{4.5e}$$

$$\begin{bmatrix} R & I_{\overline{k}} \\ I_{\overline{k}} & \overline{\Omega} \end{bmatrix} \succeq 0, \tag{4.5f}$$

$$1 - F(x)^\top R F(x) + \sum_{i=1}^{o} \sigma_i(\overline{x}) s_i(\overline{x}) - \epsilon \in \Sigma[\overline{x}]. \tag{4.5g}$$

From Lemma 2.2.5, (4.5e) is a sum-of-squares constraint, which is equivalent to a semi-definite constraint. The other constraints and the cost function are the same as those in program (3.4) in Chapter 3. Therefore, program (4.5) is convex. In the following theorem, we show that a CBF $b(x)$ and a nonlinear feedback controller $u(x)$ can be synthesized with this convex program.

Theorem 4.1.1. *Let Assumption 3.2.1 hold. Suppose $\mathcal{U} = \mathbb{R}^m$. Consider system* (4.1) *and safe set $\mathcal{S}$ as in* (3.3)*. Assume that a solution to program* (4.5) *exists, and is denoted by $\Omega, R, Y(x), \{\sigma_i(\cdot)\}_{i=1}^{o}, \epsilon$. Then*

$$u = Y(x)\Omega^{-1}F(x), \tag{4.6}$$

makes the set $\mathcal{B} := \{x \in \mathbb{R}^n \colon b(x) \geq 0\}$, where $b(x) = F(x)^\top \Omega^{-1} F(x) - 1$, an invariant set for the closed-loop system (4.1)*. Moreover, $\mathcal{B} \subseteq \mathcal{S}$.*

Proof. The proof that (4.5f)-(4.5g) is sufficient for $\mathcal{B} \subseteq \mathcal{S}$ follows similar arguments as in the part 2) of the proof for Theorem 3.2.1 in Chapter 3, we only need to substitute x_c by $F(x)$. We then prove that (4.5e) is sufficient for $\mathcal{B}$ to be an invariant

set for the closed-loop system (4.1), (4.2). The closed-loop system dynamics are given by

$$\dot{x} = (A + BK(x))F(x).$$

Then

$$\begin{aligned}
\dot{b}(x) = \; & 2F(x)^\top \Omega^{-1} M(x)(A(x) + B(x)K(x))F(x) \\
= \; & F(x)^\top \Omega^{-1} M(x)(A(x) + B(x)K(x))F(x) + \\
& F(x)^\top (A(x) + B(x)K(x))^\top M(x)\Omega^{-1} F(x) \\
= \; & F(x)^\top \Omega^{-1}[M(x)(A(x) + B(x)K(x))\Omega + \\
& \Omega(A(x) + B(x)K(x))^\top M(x)]\Omega^{-1} F(x) \\
= \; & F(x)^\top \Omega^{-1}[M(x)(A(x)\Omega + B(x)Y(x)) + \\
& (A(x)\Omega + B(x)Y(x))^\top M(x)]\Omega^{-1} F(x)
\end{aligned}$$

that is, $\dot{b}(x) = (\Omega^{-1}F(x))^\top G(x)(\Omega^{-1}F(x))$. Given that $G(x) \in \Sigma[x]$, using Lemma 2.2.5 we have $(\Omega^{-1}F(x))^\top G(x)(\Omega^{-1}F(x)) \geq 0$ for any polynomial vector $\Omega^{-1}F(x)$. Hence, we conclude the proof. □

We then consider the case for local design, where Assumption 3.2.2 holds. For this case, $F(x)$ is designed to contain both linear and nonlinear components as

$$F(x) = \begin{bmatrix} Q(x) \\ x \end{bmatrix}, \tag{4.7}$$

where $Q(x) : \mathbb{R}^n \to \mathbb{R}^{k-n-1}$ is also a vector-valued polynomial function, $Q(0) = 0$. The local design program is shown as follows.

$$\max \; \mathrm{Tr}(\Omega) \tag{4.8a}$$

$$\text{subject to } 0 \prec \Omega \in \mathbb{R}^{k\times k}, 0 \prec R \in \mathbb{R}^{k\times k}, \tag{4.8b}$$

$$Y(x) \in \mathbb{R}^{m\times k}, \sigma_1(x), \ldots, \sigma_l(x) \in \Sigma[x], \tag{4.8c}$$

$$-G(x) \in \Sigma[x]^{n\times n}, \tag{4.8d}$$

$$\begin{bmatrix} R & I_k \\ I_k & \Omega \end{bmatrix} \succeq 0, \tag{4.8e}$$

$$-F(x)^\top R F(x) + 1 - \sum_{i=1}^{l} \sigma_i(x) w_i(x) \in \Sigma[x], \tag{4.8f}$$

$$1 - \begin{bmatrix} 0 \\ a_i \end{bmatrix}^\top \Omega \begin{bmatrix} 0 \\ a_i \end{bmatrix} \geq 0, i = 1, \ldots, l, \tag{4.8g}$$

Clearly, program (4.8) is also a convex program.

Theorem 4.1.2. *Consider system* (4.1) *where* $F(x)$ *follows* (4.7)*, and let Assumption 3.2.2 hold. Assume that a solution to program* (4.8) *exists, and is denoted by* $\Omega, R, Y(x), \{\sigma_i(\cdot)\}_{i=1}^l, \epsilon$*. Set* $u(x) = Y(x)\Omega^{-1}F(x)$*. We then have that*

1. $\mathcal{I} \subseteq \mathcal{B}^c \subseteq \mathcal{S}$*, where* $\mathcal{B}^c = \{x \in \mathbb{R}^n : b(x) \leq 0\}$*,* $b(x) = F(x)^\top \Omega^{-1} F(x) - 1$*.*
2. $\mathcal{B}^c$ *is a control invariant set for* $\dot{x} = A(x)F(x) + B(x)u(x)$*.*

Proof. The proof for (4.8d) is sufficient for $\mathcal{B}^c$ to be a control invariant set, and (4.8e)-(4.8f) is sufficient for $\mathcal{I} \subseteq \mathcal{B}^c$ is analogous to that for Theorem 3.2.2. We only prove that (4.8g) are sufficient and necessary for $\mathcal{B}^c \subseteq \mathcal{S}$. Given that $\mathcal{S}$ in (3.17) is a convex polytope, $\mathcal{B}^c \subseteq \mathcal{S}$ requires for any $i = 1, \ldots, o$,

$$a_i^\top x + 1 \geq 0 \ \forall x : -F(x)^\top \Omega^{-1} F(x) + 1 \geq 0. \tag{4.9}$$

Using Lemma 2.2.5 with a constant multiplier, we have (4.9) holds if for any $i = 1, \ldots, l$, there exists $\lambda_i \geq \frac{1}{2}$ such that

$$2\lambda_i(a_i^\top x + 1) - (-F(x)^\top \Omega^{-1} F(x) + 1) \geq 0, \forall x \in \mathbb{R}^n.$$

Splitting $F(x)$ to linear parts and nonlinear parts as in (4.7), we obtain:

$$\begin{bmatrix} Q(x) \\ x \\ 1 \end{bmatrix}^\top \Psi_i \begin{bmatrix} Q(x) \\ x \\ 1 \end{bmatrix} \geq 0, i = 1, \ldots, l, \tag{4.10}$$

where

$$\Psi_i = \left[\begin{array}{cc|c} \Omega_{11}{}' & \Omega_{12}{}' & 0 \\ \Omega_{21}{}' & \Omega_{22}{}' & \lambda_i a_i \\ \hline 0 & \lambda_i a_i{}^\top & 2\lambda_i - 1 \end{array}\right].$$

The first block of Ψ_i is Ω^{-1}. (4.10) holds if $\Psi_i \succeq 0$ for any $i = 1, \ldots, l$, which is equivalent to

$$\Omega^{-1} - \begin{bmatrix} 0 \\ a_i \end{bmatrix} \frac{\lambda_i^2}{2\lambda_i - 1} \begin{bmatrix} 0 \\ a_i \end{bmatrix}^\top \succeq 0, i = 1, \ldots, l. \tag{4.11}$$

Given that $\inf_{2\lambda_i - 1 \geq 0} \frac{\lambda_i^2}{2\lambda_i - 1} = 1$, we obtain an equivalent condition for (4.11), given by $1 - \begin{bmatrix} 0 \\ a_i \end{bmatrix}^\top \Omega \begin{bmatrix} 0 \\ a_i \end{bmatrix} \geq 0$, which is (4.8g). Hence, we conclude the proof. □

Similar to the linear system case, an $\mathcal{L}$-norm based input constraint can be encoded as a convex constraint to amend the local design program (4.8)-(4.8g). The amended program for $\mathcal{U} = \mathcal{U}_1$ is shown as follows.

$$\min \ \mathrm{Tr}(\overline{\Omega}) \tag{4.12a}$$

$$\text{subject to } (4.8\text{b}) - (4.8\text{g}), \mu > 0, \tag{4.12b}$$

$$\begin{bmatrix} \Omega & Y(x)^\top \\ Y(x) & \mu I_m \end{bmatrix} \in \Sigma[x]^{(n+m)\times(n+m)}, \tag{4.12c}$$

$$\zeta - \varepsilon - \mu > 0, \tag{4.12d}$$

where $\varepsilon > 0$ is a small constant. This is a convex program since (4.12c) is a sum-of-squares constraint. The result is guaranteed by the following lemma.

Lemma 4.1.1. *Consider system* (4.1) *with* $\mathcal{U} = \mathcal{U}_1$*, where* $F(x)$ *follows* (4.7)*, and let Assumption 3.2.2 in Chapter 3 hold. Assume that a solution to* (4.12) *exists, and is denoted by* $\Omega, R, Y(x), \{\sigma_i(\cdot)\}_{i=1}^l, \mu$*. Set* $u(x) = Y(x)\Omega^{-1}F(x)$*. We then have that*

1. $\mathcal{I} \subseteq \mathcal{B}^c \subseteq \mathcal{S}$*, where* $\mathcal{B}^c = \{x \in \mathbb{R}^n : b(x) \leq 0\}$*,* $b(x) = F(x)^\top \Omega^{-1} F(x) - 1$*.*
2. $\mathcal{B}^c$ *is a control invariant set for* $\dot{x} = A(x)F(x) + B(x)u(x)$*, and* $u(x) \in \mathcal{U}_1$*,* $\forall x \in \mathcal{B}^c$*.*

Proof. From the proof for, we conclude that (4.12b) implies $\mathcal{I} \subseteq \mathcal{B}^c := \{x \in \mathbb{R}^n : b(x) \leq 0\} \subseteq \mathcal{S}$, and $\mathcal{B}^c$ is a control invariant set using $u(x)$. We only prove that (4.12c) and (4.12d) indicate $u(x) \in \mathcal{U}_1, \forall x \in \mathcal{B}^c$. Using Lemma 2.2.5, (4.12c) implies $\begin{bmatrix} \Omega & Y(x)^\top \\ Y(x) & \mu I_m \end{bmatrix} \succeq 0, \forall x \in \mathbb{R}^n$. Given that $\mu > 0$ from (4.12b), using Schur complement, we have

$$\mu\Omega - Y(x)^\top Y(x) \succeq 0, \forall x \in \mathbb{R}^n.$$

Multiplying Ω^{-1} on both sides of $\mu\Omega - Y(x)^\top Y(x)$, and define $K(x) = Y(x)\Omega^{-1}$, we have

$$\mu\Omega^{-1} - K(x)^\top K(x) \succeq 0, \forall x \in \mathbb{R}^n.$$

This induces

$$F(x)^\top [\mu\Omega^{-1} - K(x)^\top K(x)] F(x) \geq 0, \forall x \in \mathbb{R}^n$$

and further implies

$$F(x)^\top[\mu\Omega^{-1} - K(x)^\top K(x)]F(x) + \zeta - \varepsilon - \mu \geq 0, \forall x \in \mathbb{R}^n$$

provided that $\zeta - \varepsilon - \mu > 0$ from (4.12d). Recall that $u(x) = Y(x)\Omega^{-1}F(x) = K(x)F(x)$, we conclude that

$$\zeta - \varepsilon - u(x)^\top u(x) + \mu(F(x)^\top\Omega^{-1}F(x) - 1) \geq 0, \forall x \in \mathbb{R}^n.$$

Then, for any $x \in \mathcal{B}^c$, $F(x)^\top\Omega^{-1}F(x) - 1 \leq 0$, $u(x)^\top u(x) \leq \zeta - \varepsilon \leq \zeta$. Hence, $u(x) \in \mathcal{U}_1, \forall x \in \mathcal{B}^c$. The proof is complete. □

Conditions for the inf-norm, and one-norm constrains are similar to the two-norm constraint, thus are omitted here. The convex conditions for input constraints can be amended to the convex synthesis program.

4.1.2 Non-polynomial Systems

We now consider the case that $H(x) = A(x)F(x)$ and/or $B(x)$ are both non-polynomial functions by which we mean that at least one element of the vector is not a polynomial function in x. The key idea here is to use Taylor's expansion to approximate $H(x)$ and $B(x)$ with polynomial functions. To begin, we make basic assumptions for the functions $H(x)$ and $B(x)$.

Assumption 4.1.1. *$H(x)$ and $B(x)$ are of class C^{r_H} and C^{r_B}, respectively for integers r_H, $r_B \geq 1$. For all $x \in \mathcal{X} \subset \mathbb{R}^n$, where $\mathcal{X}$ is a convex local region centered at the origin. $\partial^\alpha H_i(x)$, $\partial^\beta B_i(x)$ are continuous for $x \in \mathcal{X}$ and for known L_i, $M_{ij} > 0$, where $i = 1, \ldots, n, j = 1, \ldots, m$:*

$$|\partial^\alpha H_i(x) - \partial^\alpha H_i(0)| \leq L_i||x||,$$
$$|\partial^\beta B_{ij}(x) - \partial^\beta B_{ij}(0)| \leq M_{ij}||x||.$$

Using standard Taylor's expansion, functions $H(x)$ and $B(x)$ can be written as

$$H_i(x) = \sum_{|\alpha| \leq r_H} \frac{\partial^\alpha H_i(0)}{\alpha!} x^\alpha + R_{H_i}(x)$$

$$B_{ij}(x) = \sum_{|\alpha| \leq r_B} \frac{\partial^\alpha B_{ij}(0)}{\alpha!} x^\alpha + R_{B_{ij}}(x),$$

where $R_{H_i}(x)$ and $R_{B_{ij}}(x)$ are remainders. With a slight abuse of notation, the non-polynomial system $\dot{x} = H(x) + B(x)u$ can be written in the form of

$$\dot{x} = \tilde{A}(x)\tilde{F}(x) + \tilde{B}(x)u + R_H(x) + R_B(x)u, \tag{4.13}$$

where $\tilde{A}(x)$, $\tilde{F}(x)$, $\tilde{B}(x)$ are polynomial functions. Under Assumption 4.1.1, following [146, Lemma 1], the remainders are bounded by

$$|R_{H_i}(x)| \leq \sup_{x \in \mathcal{X}} \frac{\sqrt{n} L_i}{(r_H + 1)!} ||x||^{r_H + 1},$$
$$|R_{B_{ij}}(x)| \leq \sup_{x \in \mathcal{X}} \frac{\sqrt{n} J_{ij}}{(r_B + 1)!} ||x||^{r_B + 1}.$$

Given that $x \in \mathcal{X}$, $u \in \mathcal{U}$, system (4.13) can be written in the form of a polynomial system with additive bounded noise by

$$\dot{x} = \tilde{A}(x)\tilde{F}(x) + \tilde{B}(x)u + \delta, \tag{4.14}$$

where

$$\delta^\top \delta \leq D =$$
$$\sup_{\substack{x \in \mathcal{X} \\ u \in \mathcal{U}}} \sum_{i=1}^{n} \left(\frac{\sqrt{n} L_i}{(r_H + 1)!} ||x||^{r_H + 1} + \sum_{j=1}^{m} \frac{\sqrt{n} J_{ij}}{(r_B + 1)!} ||x||^{r_B + 1} u_j \right)^2.$$

The results of synthesizing a CBF for a linear system with additive noise in Section 3.2.4 of Chapter 3 can be used here to conduct an analysis and design a method for non-polynomial systems. Similarly define $\tilde{M}(x)$ to be the Jacobian matrix of $\tilde{F}(x)$. The program to synthesize a CBF $b(x)$ and a safe controller $u(x)$ is shown as follows.

$$\min \operatorname{Tr}(\overline{\Omega}) \tag{4.15a}$$
$$\text{subject to } (4.8\text{b}) - (4.8\text{g}), \lambda(x) \in \mathbb{R}, \mu(x) \in \Sigma[x], \tag{4.15b}$$
$$\Lambda \in \Sigma[x]^{2n+1}, \tag{4.15c}$$

where $\Lambda = \begin{bmatrix} \tilde{G}(x) + \lambda(x)\Omega & -\tilde{M}(x)\Omega & 0 \\ -\Omega \tilde{M}(x)^\top & \mu(x) I_n & 0 \\ 0 & 0 & -\mu(x) D + \lambda(x) \end{bmatrix}$. In the program, $A(x)$, $F(x)$, and $B(x)$ are substituted by $\tilde{A}(x)$, $\tilde{F}(x)$, and $\tilde{B}(x)$, respectively in the constraints (4.8b)-(4.8g). $\tilde{G}(x)$ is constructed from $G(x)$ following similar substitution.

Lemma 4.1.2. *Consider system* (4.1) *where either* $H(x) = A(x)F(x)$ *or* $B(x)$ *is a non-polynomial function, let Assumption 3.2.2 hold. Assume that a solution to program* (4.15) *exists, and is denoted by* $\Omega, R, Y(x), \{\sigma_i \cdot\}_{i=1}^{l}, \lambda(x), \mu(x)$. *Set* $u(x) = Y(x)\Omega^{-1}\tilde{F}(x)$. *We then have that*

1. $\mathcal{I} \subseteq \mathcal{B}^c \subseteq \mathcal{S}$, *where* $\mathcal{B}^c = \{x \in \mathbb{R}^n : b(x) \leq 0\}$, $b(x) = F(x)^\top \Omega^{-1} F(x) - 1$.
2. $\mathcal{B}^c$ *is a control invariant set for the non-polynomial system* $\dot{x} = H(x) + B(x)u(x)$.

Note here for non-polynomial systems, we only consider constructing a control barrier function $b(x)$ locally, where the $\mathcal{B}^c = \{x \in \mathbb{R}^n : b(x) \leq 0\}$ is a bounded control invariant set. This is because the remainders of Taylor expansion bounded in a bounded set. Our technique can not be used for globally constructing an unbounded control invariant set $\mathcal{B}^c = \{x \in \mathbb{R}^n : b(x) \leq 0\}$, as the remainders can be unbounded in $\mathcal{B}^c$.

4.2 Iterative Design for Nonlinear Systems

In this section we show how to co-design a CBF and design a safe controller that are parameterized by arbitrary polynomial bases. The system dynamics are polynomial and the safe/initial sets are semi-algebraic. The nonlinear control affine system is given by

$$\dot{x} = f(x) + g(x)u, \tag{4.16}$$

where $f(x) : \mathbb{R}^n \to \mathbb{R}^n$ and $g(x) \in \mathbb{R}^n \to \mathbb{R}^m$ are locally smooth polynomial functions. The control input u is limited by $u \in \mathcal{U} := \{u \in \mathbb{R}^m : Hu + h \geq 0\}$, where $H \in \mathbb{R}^{o \times m}$, $h \in \mathbb{R}^o$. The safe $\mathcal{S}$ and $\mathcal{I}$ are semi-algebraic sets, defined as follows:

$$\mathcal{S} := \{x \in \mathbb{R}^n : s(x) \geq 0\}, \mathcal{I} := \{x \in \mathbb{R}^n : w(x) \geq 0\}, \tag{4.17}$$

where $s(x) : \mathbb{R}^n \to \mathbb{R}$ and $w(x) \in \mathbb{R}^n \to \mathbb{R}$ are both polynomial functions.

To this end, CBF $b(x)$ and feedback controller $u(x)$ are parameterized by

$$b(x) = \sum_{i=1}^{k} p_k \Lambda_k(x), u(x) = \sum_{i=1}^{k} l_i \zeta_i'(x), \tag{4.18}$$

where $p := \{p_1, \ldots, p_k\}$ and $l := \{l_1, \ldots, l_k\}$ denotes a series of parameters which will be decision variables in an optimization problem, and $\Lambda_1(x), \ldots, \Lambda_k(x)$, $\zeta_1(x), \zeta_l(x)$ are given polynomial functions. The co-design program for the CBF and controller is given by

$$\begin{aligned} &\text{find } p, l \\ &\text{subject to } b(x) < 0, \forall x \in \mathcal{S}^c, \\ &\qquad b(x) \geq 0, \forall x \in \mathcal{I}, \\ &\qquad \sup_{u(x)\in\mathcal{U}} \frac{\partial b(x)}{\partial x}(f(x) + g(x)u(x)) \geq 0, \forall x, s.t. b(x) = 0 \end{aligned} \tag{4.19}$$

However, solving (4.19) is computationally hard since it involves infinitely many constraints. Given that $f(x)$, $g(x)$, $u(x)$ are all polynomial functions, $\mathcal{S}$ and $\mathcal{I}$ are both semi-algebraic sets, Positivestellansatz (Psatz) can be leveraged to transform (4.19) into a tractable SOS program. The resulting SOS program is given follows.

$$\text{find } b(x), u(x), \sigma_{\text{safe}}(x), \epsilon_1, \sigma_{\text{init}}(x), \lambda_1(x), \epsilon_2, \lambda_2(x)$$

$$\text{subject to } b(x), u(x), \lambda_1(x), \lambda_2(x) \in \mathbb{R}[x], \sigma_{\text{safe}}(x), \sigma_{\text{init}}(x) \in \Sigma[x], \epsilon_1, \epsilon_2 > 0, \tag{4.20a}$$

$$-b(x) + \sigma_{\text{safe}}(x)s(x) - \epsilon_1 \in \Sigma[x], \tag{4.20b}$$

$$b(x) - \sigma_{\text{init}}(x)w(x) \in \Sigma[x], \tag{4.20c}$$

$$\frac{\partial b(x)}{\partial x}(f(x) + g(x)u(x)) + \lambda_1(x)b(x) - \epsilon_2 \in \Sigma[x], \tag{4.20d}$$

$$-\lambda_2(x)b(x) + Hu(x) + h \in \Sigma[x]^{o\times 1}. \tag{4.20e}$$

Here with a slight abuse of notation, by "find $b(x), u(x), \sigma_{\text{safe}}(x), \epsilon_1, \sigma_{\text{init}}(x), \lambda_1(x), \epsilon_2, \lambda_2(x)$" we mean finding parameters for certain polynomial functional basis that parameterize these functions. The following theorem show that program (4.20) can be used to co-design a CBF $b(x)$ and a feedback controller $u(x)$.

Theorem 4.2.1. *Consider a polynomial nonlinear system* (4.16)*, semi-algebraic safe set* $S = \{x \in \mathbb{R}^n : s(x) \geq 0\}$*, initial set* $I = \{x \in \mathbb{R}^n : w(x) \geq 0\}$*, and*

control admissible set $\mathcal{U} = \{u \in \mathbb{R}^m : Hu + h \geq 0\}$. *Assume a solution to* (4.20) *exists and is denoted by* $b(x), u(x), \sigma_{\text{safe}}(x), \epsilon_1, \sigma_{\text{init}}(x), \lambda_1(x), \epsilon_2, \lambda_2(x)$. *Then,* $b(x)$ *is a CBF and* $\mathcal{B} = \{x \in \mathbb{R}^n : b(x) \geq 0\}$ *is an invariant set with respect to vector field* $f(x) + g(x)u(x)$.

Proof. Condition (4.20b) indicates that for any x, $-B(x) + \sigma_{\text{safe}}s(x) - \epsilon_1 \geq 0$, thus for any x, $-B(x) + \sigma_{\text{safe}}s(x) > 0$. Therefore, for any $x \in \mathcal{S}^c$, we directly have that $\sigma_{\text{safe}}s(x) \leq 0$, and further $b(x) < 0$, i.e., $\mathcal{B} \subseteq \mathcal{S}$ holds. Similarly (4.20c) implies $\mathcal{I} \subseteq \mathcal{B}$ following similar arguments.Using Lemma 2.2.5, constraint (4.20d) implies $\mathcal{B}$ is an invariant set with respect to vector field $f(x)+g(x)u(x)$, because for any x such that $b(x) = 0$, $\frac{\partial b(x)}{\partial x}(f(x) + g(x)u(x)) - \epsilon_2 \geq 0$, and thus $\frac{\partial b(x)}{\partial x}(f(x) + g(x)u(x)) > 0$. Using Lemma 2.2.5 again, constraint (4.20e) indicates that $Hu(x) + h$ is element-wise nonnegative for any $x \in \partial\mathcal{B}$. Therefore, $u(x) \in \mathcal{U}$, for any $x \in \partial\mathcal{B}$. Hence, we conclude the proof. □

The convex design methodology in Section 4.1.1 parameterizes the CBF $b(x)$ and $u(x)$ by certain basis functions as in (4.2) for convexification. Here, these functions can be parameterized by arbitrary polynomial basis, thus more flexible in design. Moreover, $\mathcal{S}$ and $\mathcal{I}$ can be defined by any polynomials functions $s(x)$ and $w(x)$, but not limited to the specific forms in Section 4.1.1. However, program (4.2.1) is *nonconvex* due to the products of decision variables in constraints, i.e., $\sigma_{\text{safe}}(x)s(x)$, $\sigma_{\text{init}}(x)w(x)$, $\frac{\partial b(x)}{\partial x}u(x)$, $\lambda_1(x)b(x)$, and $\lambda_2(x)b(x)$. The nonconvex SOS program cannot be directly transformed into a semi-definite program, and solved by numerical solvers. Here, we propose an iterative algorithm to solve (4.2.1).

1) **Initialization**: We first fix the degree of polynomials $b(x)$, $\sigma_{\text{safe}}(x)$, $\sigma_{\text{init}}(x)$, $\lambda_1(x)$, $\lambda_2(x)$ and $u(x)$. The polynomial/monomial scalar/vector basis for $b(x)$ and $u(x)$ have degree upper bounded by the aforementioned degrees of $b(x)$. Unlike the iterative procedure proposed in [147] which initializes the control law by a scaled LQR controller, we find the initialized feasible control input $u^0(x)$ by solving a

feasibility SOS program.

$$\begin{aligned} &\text{find } k_1, \ldots, k_l, \sigma_{\text{cont}} \\ &\text{s.t. } H(k_0 + \sum_{j=1}^{l} k_j v_j(x)) + h \cdot \sigma_{\text{cont}} \in \Sigma[x]^{o\times 1}. \end{aligned} \tag{4.21}$$

We note here that there is no control barrier function $b(x)$ at this stage of finding the initial feasible control input $u^0(x)$. Therefore, $u^0(x)$ can not be restricted to the domain of $\partial\mathcal{B}$ as that in (4.20e). Other than directly interpreting $H(k_0 + \sum_{j=1}^{l} k_j v_j) + h \in \Sigma[x]^h$, we add an additional positive multiplier σ_{cont} which satisfies $\sigma_{\text{cont}} - \epsilon_3 \in \Sigma[x]$, $\epsilon_3 > 0$ to avoid introducing constant terms in the SOS constraints, as well as improving feasibility. The resulting initial controller $u^0(x)$ is derived by the parameters $k_1, \ldots, k_l$ and the scaled term σ_{cont} from the solution of (4.21)

$$u^0(x) = \frac{1}{\sigma_{\text{cont}}} \cdot (k_0 + \sum_{j=1}^{l} k_j v_j(x)). \tag{4.22}$$

Given initial input $u^0(x)$, the corresponding scaled multiplier σ_{cont}, the initial control barrier function $b^0(x)$ can be found by solving an initial feasibility SOS program as

$$\begin{aligned} &\text{find } b(x), \sigma_{\text{safe}}(x), \sigma_{\text{init}}(x) \\ &\text{s.t. } -b(x) + \sigma_{\text{safe}}(x)s(x) - \epsilon_1 \in \Sigma[x], \\ &\quad b(x) - \sigma_{\text{init}}(x)w(x) \in \Sigma[x], \\ &\quad \sigma_{\text{cont}}(x) \cdot \frac{\partial b(x)}{\partial x}(f(x) + g(x)u^0(x)) - \epsilon_2 \in \Sigma[x], \end{aligned} \tag{4.23}$$

The invariance condition (4.20d) is strengthened to be $\frac{\partial b(x)}{\partial x}(f(x)+g(x)u(x)) - \epsilon_2 \in \Sigma[x]$ for convexity and simplicity of computing. $\sigma_{\text{cont}}(x) \cdot \frac{\partial b(x)}{\partial x}(f(x)+g(x)u^0(x)) - \epsilon_2$ is guaranteed to be a polynomial, since $\sigma_{\text{cont}}(x) \cdot u^0(x)$ is a polynomial.

After obtaining a feasible initial control input $u^0(x)$ and control barrier function $b^0(x)$, the problem of control barrier functions synthesis can be regarded as a barrier certificates synthesis problem with vector field $f(x) + g(x)u^0(x)$. The multipliers $\lambda_1^0(x)$, $\lambda_2^0(x)$ are fixed to be 0 or 1 in initialization for simplicity. The initial control barrier function $b^0(x)$ is used to enlarge the size of the control invariant

set incrementally. The following steps of the algorithm iteratively solve the SOS program to address the bisecting terms $\lambda_1(x)b(x)$ and $\frac{\partial b(x)}{\partial x}(f(x)+g(x)u(x))$ in (4.20d).

2) **Update the control input** $u^k(x)$: At iteration k, given a control barrier function from (4.23) (when $k=1$) or (4.25) (when $k\geq 2$), the controller synthesis is constrained to (4.20e). Fixing $b(x)=b^{k-1}(x)$, a convex programming synthesis procedure for $u^k(x)$ is

$$\begin{aligned}&\text{find } k_0,\ldots,k_l,\lambda_1,\lambda_2\\ &\text{s.t.} -\lambda_2 b^{k-1}(x)+\ H(k_0+\sum_{j=1}^{l}k_j v_j)+h\in\Sigma[x]^{o\times 1},\\ &\frac{\partial b^{k-1}(x)}{\partial x}(f(x)+g(x)u(x))+\lambda_1 b^{k-1}(x)-\epsilon_2\in\Sigma[x],\end{aligned}\tag{4.24}$$

and we have that $u^k(x)=(k_0+\sum_{j=1}^{l}k_j v_j)$. Here we use λ_1 other than λ_1^{k-1} since $b(x)$ has been substituted by $b^k(x)$, thus there is no bilinear term anymore. By limiting the domain of the controller to $\partial\mathcal{B}$, there is no need to have additional multiplier $\sigma_{\text{cont}}(x)$ as that has been used in initial controller design for feasibility.

3) **Synthesize the control barrier function** $b^k(x)$: After obtaining a feasible control input $u^{k-1}(x)$, the synthesis of a control barrier function $b^k(x)$ relies on fixed multipliers $\lambda_1^{k-1}(x)$, $\lambda_2^{k-1}(x)$ to bypass the bilinear terms. Searching for $b^k(x)$ and the remaining multipliers follows the following SOS program

$$\begin{aligned}&\text{find } b(x),\sigma_{\text{safe}}(x),\sigma_{\text{init}}(x),\sigma_{\text{enl}}(x)\\ \text{s.t. }&-b(x)+\sigma_{\text{safe}}(x)s(x)-\epsilon_1\in\Sigma[x],\\ &b(x)-\sigma_{\text{init}}(x)w(x)\in\Sigma[x],\\ &\frac{\partial b(x)}{\partial x}(f(x)+g(x)u^k(x))+\lambda_1^{k-1}(x)b(x)-\epsilon_2\in\Sigma[x],\\ &-\lambda_2^{k-1}b(x)+Hu^k(x)+h\in\Sigma[x],\\ &b(x)-\sigma_{\text{enl}}(x)b^{k-1}(x)\in\Sigma[x],\end{aligned}\tag{4.25}$$

where $\sigma_{\text{enl}}(x)\in\Sigma[x]$. Here the control law $u^{k-1}(x)$ is substituted for the variable $u(x)$, and the multipliers $\lambda_1(x)$ $\lambda_2(x)$ are substituted by $\lambda_1^{k-1}(x)$ and $\lambda_2^{k-1}(x)$, respectively. We introduce additional constraints $b(x)-\sigma_{\text{enl}}(x)b^{k-1}(x)\in\Sigma[x]$

to enlarge the volume of the control invariant set $\mathcal{B}^k$ by enforcing $\mathcal{B}^{k-1} \subseteq \mathcal{B}^k$. A similar technique is also used in [148].

4) **Update the multipliers**: The multiplier $\lambda_1^k(x)$ updates rely on a fixed control barrier function $b^k(x)$ and input $u^k(x)$. Clearly, there is no bilinearity in the control input update procedure (4.24). The multipliers $\lambda_1^k(x)$ and $\lambda_2^k(x)$ are obtained by directly solving it, while there is no need to fix $b(x)$ and re-solve the SOSP.

Remark 4.2.1. For the case where (4.21) or (4.23) is infeasible, there are two ways to ensure feasibility: (i) Increase the degree of the polynomial basis for $b(x)$ and $u(x)$; (ii) Re-solve the problem (4.21) with an alternative objective function for a different initialization. The algorithm terminates upon convergence in two consecutive iterations, i.e. $b^k(x) = b^{k-1}(x)$.

4.3 Numerical Examples

In this section we demonstrate the efficacy of the proposed methodology on numerical examples. In Section 4.3.1 we leverage the convex co-design program (4.8) for a polynomial system example. In Section 4.3.2 we apply the convex co-design program (4.15) to a non-polynomial system example. In Section 4.3.3 we facilitate the nonconvex program (4.20) and the iterative algorithm in Section 4.2 to a polynomial system example.

4.3.1 Vehicles Platooning System

In this example we consider a polynomial system on the example of two cars moving in a platoon formation from [149]. The system as a third-order polynomial model as:

$$\dot{x}_1 = u_1 - \gamma_1 - \beta_1 x_1 - \alpha_1 x_1^2 \tag{4.26a}$$

$$\dot{x}_2 = u_2 - \gamma_2 - \beta_2 x_2 - \alpha_2 x_2^2 \tag{4.26b}$$

$$\dot{x}_3 = x_1 - x_2. \tag{4.26c}$$

γ_1	β_1	α_1	γ_2	β_2	α_2	ζ
$0.005\frac{\text{N}}{\text{kg}}$	$0.1\frac{\text{N s}}{\text{kg m}}$	$0.02\frac{\text{N s}^2}{\text{kg m}^2}$	$0.005\frac{\text{N}}{\text{kg}}$	$0.2\frac{\text{N s}}{\text{kg m}}$	$0.04\frac{\text{N s}^2}{\text{kg m}^2}$	$50\frac{\text{N}^2}{\text{kg}^2}$
τ_h	d_0	d_1	v_M	$\bar{v}$	d	
0.2s	5m	10m	$22.2\frac{\text{m}}{\text{s}}$	$8.5\frac{\text{m}}{s}$	8m	

Table 4.1: Parameters for the two cars platooning simulation in Section 4.3.1

Here, x_1 and x_2 represent the velocity of the preceding and own vehicle, and x_3 represents the relative distance between the two vehicles. The two components of control input, u_1 and u_2 represent the forces normalized by vehicle mass, and γ_k, β_k, α_k for $k = 1, 2$ are the static, rolling and aerodynamic-drag friction coefficients normalized by vehicle mass. The forces are two-norm constrained as $||u||_2^2 \leq \zeta$, for a fixed $\zeta > 0$. Values of employed parameters are shown in Table 4.1.

Safety for this system is defined by the set $\mathcal{S}$ as

$$\begin{aligned}\mathcal{S} := \{x \in \mathbb{R}^3 : d_0 + \tau_h x_2 \leq x_3, x_3 \leq d_1,\\ 0 \leq x_1, x_1 \leq v_M, 0 \leq x_2, x_2 \leq v_M\}.\end{aligned} \tag{4.27}$$

Here $d_0 + \tau_h x_2$ is a relative distance to avoid collision between the two vehicles, d_1 is the distance required to keep the platoon, v_M is the maximum velocity allowed on the road. For this system, we cannot directly leverage the result in Section 4.1.1 to synthesize a CBF, since the origin $[0, 0, 0]$ is not within the safe set (as $d_0 + \tau_h \cdot 0 \leq 0$ is not satisfied). A coordinate transformation is necessary for this case. However, as discussed in Section 4.1.1, a coordinate transformation for general nonlinear systems can be quite hard, which however, is tractable for this case. The initial set is given by

$$\mathcal{I} := \{x \in \mathbb{R}^3 : -(x - c)^\top (x - c) + 0.2 \geq 0\}, \tag{4.28}$$

where $c = [10, 10, 7.5]$. By taking $x_c = x - c$, we obtain the following shifted system dynamics

$$\dot{\tilde{x}}_1 = u_1 - \gamma_1 - (\beta_1 + 2\alpha_1 c_1)\tilde{x}_1 - \alpha_1 \tilde{x}_1^2 - \beta_1 c_1 - \alpha_1 c_1^2 \tag{4.29a}$$

$$\dot{\tilde{x}}_2 = u_2 - \gamma_2 - (\beta_2 + 2\alpha_2 c_2)\tilde{x}_2 - \alpha_2 x_2^2 - \beta_2 c_2 - \alpha_2 c_2^2 \tag{4.29b}$$

$$\dot{\tilde{x}}_3 = \tilde{x}_1 - \tilde{x}_2. \tag{4.29c}$$

The nonlinear terms in the dynamics are $Q(x) = [\tilde{x}_1^2; \tilde{x}_2^2]$. Combining them with the shifted linear terms $\tilde{x} = [\tilde{x}_1; \tilde{x}_2; \tilde{x}_3]$, we have $F(x_c) = [\tilde{x}_1^2; \tilde{x}_2^2; \tilde{x}_1; \tilde{x}_2; \tilde{x}_3]$. The control law can therefore be designed with a feedback term and an open loop term as

$$u(x_c) = KF(x_c) + \theta,$$

where $\theta_1 = \gamma_1 + \beta_1 c_1 + \alpha_1 c_1^2$. Using this control law, the closed loop system becomes $\dot{\tilde{x}} = AF(x_c) + BKF(x_c)$, where

$$A = \begin{bmatrix} -\alpha_1 & 0 & -\beta_1 - 2\alpha_1 c_1 & 0 & 0 \\ 0 & -\alpha_2 & 0 & -\beta_2 - 2\alpha_2 c_2 & 0 \\ 0 & 0 & 1 & -1 & 0 \end{bmatrix},$$
$$B = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}.$$

The safe set for the shifted dynamics is

$$\tilde{\mathcal{S}} := \{x \in \mathbb{R}^3 : d_0 + \tau_h(\tilde{x}_2 + c_1) \leq (\tilde{x}_3 + c_3), \tilde{x}_3 \leq d_1 - c_3,$$
$$0 \leq \tilde{x}_1 + c_1, \tilde{x}_1 \leq v_M - c_1, 0 \leq \tilde{x}_2 + c_2, \tilde{x}_2 \leq v_M - c_2\}.$$

By solving the sum-of-squares program (4.8), we obtain the CBF with the associated feedback controller. The control gain K is

$$K = \begin{bmatrix} 0.016 & -0.007 & -0.9 & 0.871 & -1.053 \\ -0.002 & 0.023 & 0.440 & -0.465 & 1.047 \end{bmatrix}.$$

The control invariant set is visualized in Figure 4.1. The vector field points inwards the control invariant set on the boundary.

Figure 4.2 shows different level set of the control limitation function $||u(x)||_2^2 - \zeta$. It can be inspected that the control invariant set is within the zero sub-level set of $||u(x)||_2^2 - \zeta$, with a certain level of robustness due to ε. The -5-level set of the control limitation function intersects the control invariant set, whereas the 5-level set has no intersection.

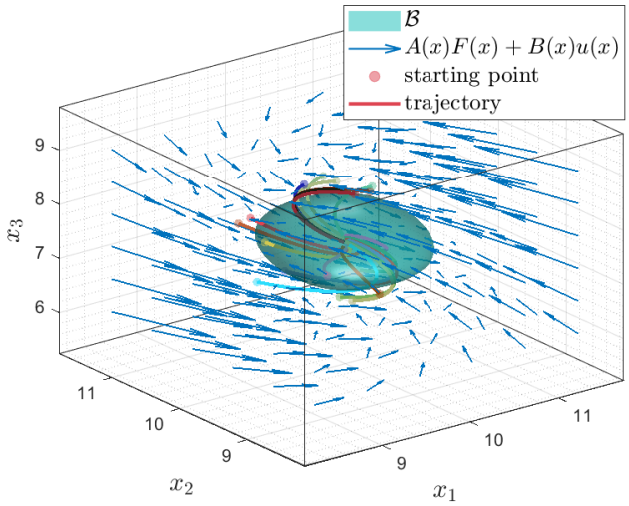


Figure 4.1: Control invariant set for system (4.26) plotted in green. The blue quivers represents the vector field $AF(x_c) + BKF(x_c)$ at each point. We also validate the robustness of the feedback controller by simulating trajectories of the closed loop system. It can be seen all trajectories starting from points outside of the control invariant set converge inside.

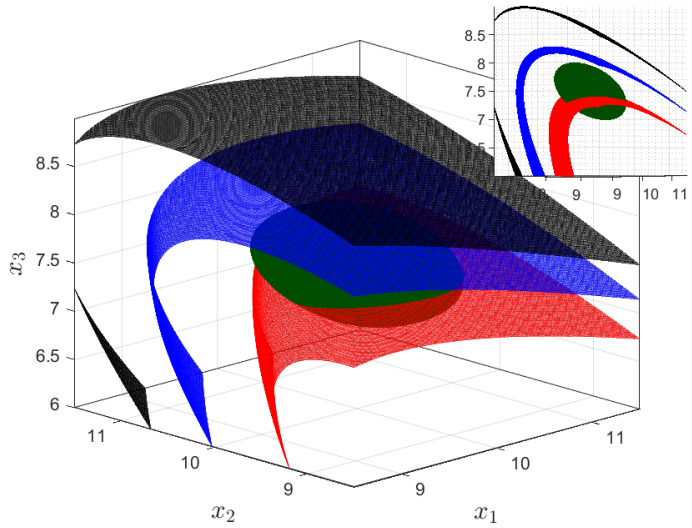


Figure 4.2: The dark blue set is the 0-level set, the black one is the 5-level set, the red one is the -5-level set of of $\zeta - ||u(x)||_2^2$.

4.3.2 Differential Driving Car

We now demonstrate our method on a non-polynomial system as proposed in Section 4.1.2. Consider a differential driving car:

$$\begin{bmatrix} \dot{p}_x \\ \dot{p}_y \\ \dot{v} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} v\sin(\theta) \\ v\cos(\theta) \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ a \\ \omega \end{bmatrix}. \tag{4.30}$$

States p_x and p_y denote the car's position, while v and θ are the linear velocity and angle of the wheels, respectively. The input is the acceleration of the wheels, i.e., a and the angular velocity ω. We consider the case that the car is moving inside a bounded region for safety, where $-1 \leq p_x \leq 1$, $-1 \leq p_y \leq 1$, and keeping $-3 \leq v \leq 3$. The angle θ is bounded by $-\pi \leq \theta \leq \pi$. These state constraints will be treated as the safe set $\mathcal{S}$. The initial set is defined as starting from a initial posture $p_x = 0, p_y = 0, v = 0, \theta = 0$. As the initial set has been degenerated into a single point, the initial condition constraint (4.8f) can be simplified by

$$\left\{-F(x)^\top R F(x) + 1\right\} |_{p_x=0,p_y=0,v=0,\theta=0} \geq 0.$$

The nonlinearity in the dynamics is due to the terms $\sin(\theta)$ and $\cos(\theta)$. Using a Taylor expansion, we obtain polynomial dynamics with noise given by

$$\begin{aligned}
\dot{p}_x &= v\widetilde{\sin}(\theta) + R_{p_x} \\
\dot{p}_y &= v\widetilde{\cos}(\theta) + R_{p_y} \\
\dot{v} &= a, \\
\dot{\theta} &= \omega,
\end{aligned} \tag{4.31}$$

where $\widetilde{\sin}(\theta) = \theta - \theta^3/6 + \theta^5/120 - \theta^7/5040$, $\widetilde{\cos}(\theta) = 1 - \theta^2/2 + \theta^4/24 - \theta^6/720 + \theta^8/40320$, and the remainders are

$$\begin{aligned}
R_{p_x} &= v \sin(\sigma\theta) \left|_{\sigma\in(0,1)} \cdot \frac{\theta^8}{8!}, \right. \\
R_{p_y} &= v \cos(\sigma\theta)|_{\sigma\in(0,1)} \cdot \frac{\theta^9}{9!}.
\end{aligned}$$

From the state constraints, we have $R_{p_x} \leq 0.7060$, $R_{p_y} \leq 0.2464$. The bound of noise is $D = 0.5592$. We use $\tilde{F}(x) = [p_x; p_y; v; \theta]$, and also

$$\tilde{A}(x) = \begin{bmatrix} 0 & 0 & \widetilde{\sin}(\theta) & 0 \\ 0 & 0 & \widetilde{\cos}(\theta) & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, B(x) = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix},$$

and $M(x) = I_4$. Given that $\widetilde{\sin}(\theta)$ and $\widetilde{\cos}(\theta)$ are polynomial functions in θ, with a maximal degree of 8, $Y(x)$ is constructed as a polynomial matrix with every

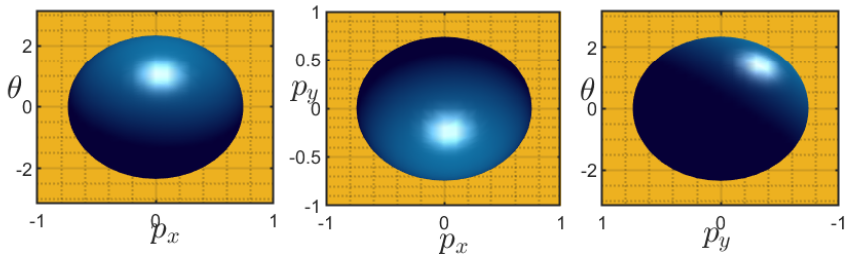


Figure 4.3: The blue ball is the invariant set $\mathcal{B}$, which is inside the safe set $\mathcal{S}$ shown as the yellow box, i.e., $-1 \leq p_x \leq 1$, $-1 \leq p_y \leq 1$, $-\pi \leq \theta \leq \pi$.

element being a monomial of θ, with maximal degree of 8. The other decision variables p_x, p_y and v are not necessary to be included to make $-\Omega A(x)^\top M(x)^\top - Y(x)^\top B(x)^\top M(x)^\top - M(x)A(x)\Omega - M(x)B(x)Y(x)$ a sum-of-squares matrix. Figure 4.3 shows the invariant set $\mathcal{B}$ projected to the $p_x - p_y - \theta$ plane. The synthesized controller is computed for 20 runs, where a different, randomly chosen initial condition is chosen at each run. The resulting trajectories are shown in Figure 4.4. All trajectories stay within the safe set $\mathcal{S}$.

4.3.3 Iterative Design

We first consider a general second order polynomial nonlinear control affine system. This system is defined by

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ x_1 + \frac{1}{3}x_1^3 + x_2 \end{bmatrix} + \begin{bmatrix} x_1^2 + x_2 + 1 \\ x_2^2 + x_1 + 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}, \tag{4.32}$$

where the control input is box constrained, i.e. $u_1 \in [-1.5, 1.5]$, $u_2 \in [-1.5, 1.5]$. The safe set is defined by a disc $S = \{x \in \mathbb{R}^2 : x_1^2 + x_2^2 - 3 \leq 0\}$, and initial set defined by $I = \{x \in \mathbb{R}^2 : (x_1 - 0.4)^2 + (x_2 - 0.4)^2 - 0.16 \leq 0\}$. We leverage the

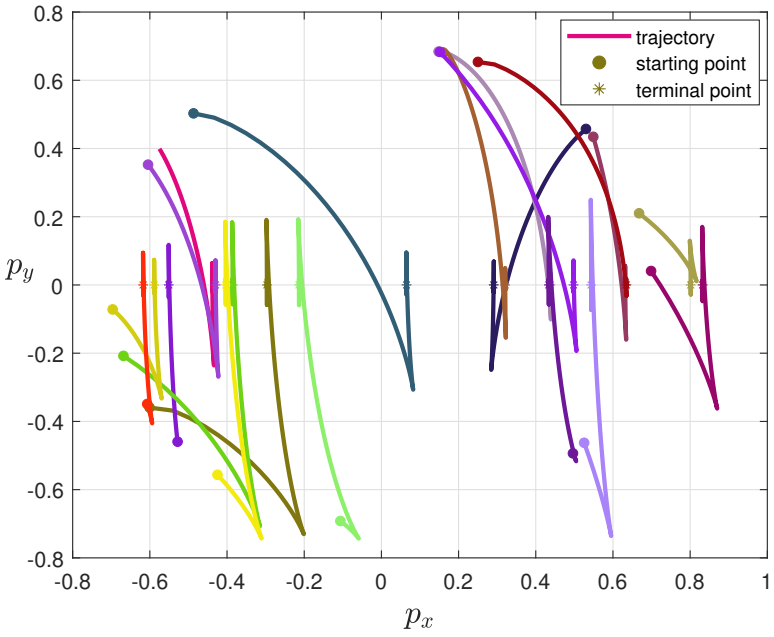
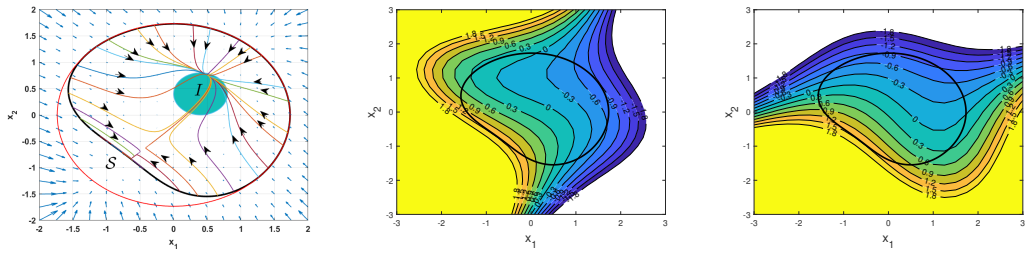


Figure 4.4: Twenties trajectories of the differential driving car in (4.30). Starting point are randomly generated by $-0.8 \leq p_x \leq 0.8$, $-0.8 \leq p_y \leq 0.8$, $-\pi \leq \theta \leq \pi$, $-2 \leq v \leq 2$. All trajectories stay within the safe set $\mathcal{S}$.

control barrier functions synthesis procedures (4.20) to find a polynomial CBF $b(x)$. The control invariant set $\mathcal{B}$ obtained by the CBF design and the values of the safe controllers are shown in Figure 4.5. The vector field, which is represented by the arrows in Figure 4.5a point inside $\mathcal{B}$ on $\partial\mathcal{B}$. The value of the polynomial control law $u_(x)$ is within $[-1.5, 1.5]$ in both coordinates.



(a) Phase portrait for the system (4.32)

(b) Level set of u_1

(c) Level set of u_2

Figure 4.5: The interior of the red disc represents the safe set, the interior of the blue disc represents the initial set from which the trajectories start. The black closed curve encircling the initial set is the control invariant set, defined by the super-zero level set of $b(x)$. The arrows in the figure represent the vector field. The colorful lines are the trajectories starting from $\partial\mathcal{B}$.

4.4 Conclusion

In this chapter we extended the result in Chapter 3 to nonlinear systems, including both polynomial and non-polynomial systems. We proposed a convex design approach for efficient computation, and an iterative design approach to reduce conservativeness. Efficacy of our appraoches is demonstrated on nonlinear system examples.

5

Distributed Safety Verification and Safe Controller Design for Multi-Agent Systems

In this chapter we consider the safety verification and safe control design problem for networked multi-agent systems, which constitutes one of the main challenges mentioned in Chapter 1. Scalable distributed safety verification and safe control design algorithms are proposed using a quadratic programming framework induced by multiple CBFs.

5.1 Introduction

Safety of a dynamical system requires the system state to remain in a safe set for all time. This property is important in many applications such as collision avoidance [150, 151], vehicle platooning [152, 153], vehicle merging control [154], etc. For a single agent system, safety is usually captured by introducing constraints on the state of the agent and the environment. For a multi-agent system, the meaning of safety extends to capture the interactions among agents. In this case, safety is encoded by coupling constraints over the states of a group of agents. For a

networked multi-agent system, where agents cooperate to satisfy safety constraints, we consider designing distributed algorithms to ensure safety for all agents.

Another problem of interest is to validate the proposed control law. For a single agent system, an agent can evaluate the system behaviour to characterize its risk of being unsafe under the employed control input. Similarly, for a multi-agent safety verification problem, cooperation among agents is necessary since safety involves multiple agents. In summary, this chapter focuses on designing a distributed protocol for safe control input design and developing a distributed safety verification algorithm.

5.2 Distributed Safe Control Law

Consider an N-agent system with the dynamics of the i-th agent described by

$$\dot{x}_i = f_i(x_i) + g_i(x_i)u_i, \tag{5.1}$$

where $x_i(t) \in \mathcal{X}_i \subset \mathbb{R}^{n_i}$ denotes its state, $u_i \in \mathcal{U}_i \subseteq \mathbb{R}^{m_i}$ denotes its control input, and $\mathcal{U}_i$ is a convex set. The dynamics $f_i(x_i) : \mathcal{X}_i \to \mathbb{R}^{n_i}$ and $g_i(x_i) : \mathcal{X}_i \to \mathbb{R}^{n_i} \times \mathbb{R}^{m_i}$ are both locally Lipschitz-continuous on a compact set $\mathcal{X}_i \subset \mathbb{R}^{n_i}$, which represents the domain of each agent. Vector $\boldsymbol{x} = [x_1^\top, \ldots, x_N^\top]^\top$ stacks the states of all systems, $\boldsymbol{u} = [u_1^\top, \ldots, u_N^\top]^\top$ stacks the control inputs, while $f(\boldsymbol{x}) = [f_1(x_1)^\top, \ldots, f_N(x_N)^\top]^\top$, $g(\boldsymbol{x}) = \mathrm{diag}(g_1(x_1), \ldots, g_N(x_N))$ stack the dynamics for each agent. The domain and control admissible set for the multi-agent system are then defined by

$$\mathcal{X} := \prod_{i=1}^{N} \mathcal{X}_i, \quad \mathcal{U} := \prod_{i=1}^{N} \mathcal{U}_i,$$

where $\prod$ represents the Cartesian product for the state space of all the agents. Given that all $\mathcal{X}_i$, $i = 1, \ldots, N$, are assumed to be compact, compactness of $\mathcal{X}$ is assured using Tychonoff's theorem [155]. In this way, the system dynamics of the whole multi-agent system can be compactly modelled by $\dot{\boldsymbol{x}} = f(\boldsymbol{x}) + g(\boldsymbol{x})\boldsymbol{u}$.

The networked system is described by an undirected and connected graph $\mathcal{G}$, with nodes set $\mathcal{V} = \{1, \ldots, N\}$, and edges set $\mathcal{E}$ such that $\{i, j\} \in \mathcal{E}$ if agent j

communicates with agent i. Agents are grouped in E sub-networks with specific safety requirement. For each sub-network $\mathcal{G}_e$, $e = 1, \ldots, E$, the set of grouped agents is $\mathcal{V}_e \subseteq \mathcal{V}$. Let $\boldsymbol{x}_e = [x_i^\top]_{i\in\mathcal{V}_e}^\top$ be the stacked states in group e. Each agent i can communicate and cooperate with its neighbour $j \in \mathcal{N}_i$ to stay safe inside group e by ensuring

$$\boldsymbol{x}_e(t) \in \mathcal{S}_e := \{\boldsymbol{x}_e : s_e(\boldsymbol{x}_e) \geq 0\} \ \forall t \geq 0, \tag{5.2}$$

where $s_e(\cdot) \in \mathbb{R}$. Define $\mathcal{S} := \prod_{i=1}^{E} \mathcal{S}_e$. We let $\mathcal{C}_i$ be the set of constraints agent i participates in; then we have $\mathcal{V}_e = \{i | e \in \mathcal{C}_i\}$.

Assumption 5.2.1. *For each $e = 1, \ldots, E$, sub-network $\mathcal{G}_e$ is connected and undirected.*

Connectivity enables communication among agents in every sub-network $\mathcal{G}_e, e \in \{1, \ldots, E\}$. Agents in $\mathcal{G}_e$ are then able to cooperatively design a controller $\boldsymbol{u}_e(\boldsymbol{x})$ for safety, satisfying$s_e(\boldsymbol{x}_e) \geq 0$.

Assumption 5.2.2. *Given sets $\mathcal{X}$ and $\mathcal{S}_e, e = 1, \ldots, E$, we assume there exist control barrier functions $b_e(\cdot)$, such that $\mathcal{B}_e := \{\boldsymbol{x}_e : b_e(\boldsymbol{x}_e) \geq 0\} \subseteq \mathcal{S}_e, e = 1, \ldots, E$. Define $\mathcal{B} := \prod_{e=1}^{E} \mathcal{B}_e$, and $\mathcal{H} := \mathcal{B} \cap \mathcal{X}$. We further assume that* $\mathrm{Int}(\mathcal{H}) \neq \emptyset$.

Assumption 5.2.2 directly then implies that $\mathrm{Int}(\mathcal{S}) \neq \emptyset$ and $\mathrm{Int}(\mathcal{B}) \neq \emptyset$. This is essential for using CBF methods to design safe controllers. However, checking emptiness of these sets is a challenging task. When $s_e(\boldsymbol{x}_e)$, $b_e(\boldsymbol{x}_e)$, $e = 1, \ldots, E$, are polynomial functions, and $\mathcal{X}$ is defined by polynomial functions as well, emptiness can be checked via sum-of-squares programming. We refer the reader to [91] for further details.

Assumption 5.2.3. *Consider the multi-agent system* (5.1) *and CBFs $b_e(\boldsymbol{x}_e)$, $e = 1, \ldots, E$, class-$\mathcal{K}$ functions $\alpha_{ie}(\cdot), i = 1, \ldots, N, e \in \mathcal{C}_i$. For every $\boldsymbol{x} \in \mathcal{B}$, we assume there exists a locally Lipschitz $\boldsymbol{u} = [u_1^\top \in \mathcal{U}_1, \ldots, u_N^\top \in \mathcal{U}_N]^\top \in \mathcal{U}$, such that for any $e \in \{1, \ldots, E\}$:*

$$\sum_{i\in\mathcal{V}_e} \left(\frac{\partial b_e}{\partial x_i}(f_i(x_i) + g_i(x_i)u_i) + \alpha_{ie}(b_e) \right) \geq 0. \tag{5.3}$$

The summation in (5.3) follows from applying the chain rule and considering the partial derivative of $b_e(\boldsymbol{x}_e)$ with respect to the state x_i of every agent $i \in \mathcal{V}_e$.

Assumption 5.2.3 guarantees the existence of one controller $\boldsymbol{u}$ that satisfies all the CBF constraints. This property is also known as control sharing property [102, Definition 2]. CBFs that satisfy Assumption 5.2.2 and 5.2.3 can be designed by sum-of-squares programming [120].

Following [69, Theorem 3], safety constraints can be incorporated in the CBF-QP formulation given by

$$\begin{aligned} J^* = \min_{\boldsymbol{u}\in\mathcal{U}} & \sum_{i=1}^{N} ||u_i - u_i^{\text{des}}(x_i)||_2^2 \\ & \text{s.t.} \sum_{i\in\mathcal{V}_e} \left(\frac{\partial b_e}{\partial x_i}(f_i(x_i) + g_i(x_i)u_i) + \alpha_{ie}(b_e) \right) \geq 0, \\ & \quad \forall e \in \{1, \ldots, E\}, \end{aligned} \tag{5.4}$$

where $\alpha_{ke}(\cdot)$'s are class-$\mathcal{K}$ functions, and hence also $\sum_{k\in\mathcal{V}_e} \alpha_{ke}(\cdot)$ is also a class-$\mathcal{K}$. $u_i^{\text{des}}(x_i)$ is a nominal stabilizing control input.

The CBF constraints in (5.4) are defined on the control inputs for multiple agents. If every agent regards other agents as fixed, (5.4) decomposes to a family of problems, one for each $i = 1, \ldots, N$,

$$\begin{aligned} & \min_{u_i\in\mathcal{U}_i} ||u_i - u_i^{\text{des}}(x_i)||_2^2 \\ & \text{s.t.} \frac{\partial b_e}{\partial x_i}(f_i(x_i) + g_i(x_i)u_i) + \alpha_{ie}(b_e) \geq 0, \forall e \in \mathcal{C}_i. \end{aligned} \tag{5.5}$$

Under Assumptions 5.2.2 and 5.2.3, (5.4) is guaranteed to be feasible, but feasibility of (5.5) is not ensured for every $i \in \{1, \ldots, N\}$. In this work, we propose an improved distributed framework for solving (5.4) with guaranteed feasibility.

Let

$$\begin{aligned} J_i(u_i) &= ||u_i - u_i^{\text{des}}(x_i)||_2^2, \\ h_{ie}(u_i) &= -\left(\frac{\partial b_e}{\partial x_i}(f_i(x_i) + g_i(x_i)u_i) + \alpha_{ie}(b_e) \right). \end{aligned} \tag{5.6}$$

We then have the following *safety* results.

Proposition 5.2.1 ([156, Proposition 1])**.** Consider Assumptions 5.2.2, 5.2.3. Let $\boldsymbol{u}^*_{\text{nom}}(\boldsymbol{x})$ be the optimal solution of (5.4). Suppose $\boldsymbol{u}^*_{\text{nom}}(\boldsymbol{x})$ is locally Lipschitz continuous for every $\boldsymbol{x} \in \mathcal{B}$, then set $\mathcal{B}$ is forward invariant under vector field $f(\boldsymbol{x}) + g(\boldsymbol{x})\boldsymbol{u}^*_{\text{nom}}(\boldsymbol{x})$.

Remark 5.2.1. Local Lipchitz continuity of $\boldsymbol{u}^*_{\text{nom}}(\boldsymbol{x})$ is important for forward invariance of $\mathcal{B}$ under the vector field $f(\boldsymbol{x})+g(\boldsymbol{x})\boldsymbol{u}^*_{\text{nom}}(\boldsymbol{x})$. This can be guaranteed if the CBF constraints are linearly independent, and there are no input constraints. For more general cases, (strong) forward invariance can be guaranteed for a discontinuous vector field, under certain regularity conditions on the different CBFs. Interested readers are referred to [157, 158], and [159, Section 9] for a comprehensive review. As this is tangential to the focus of our work, we will concentrate on the distributed implementation of the QP induced from multi-CBFs (5.4).

Notice that, even not shown explicitly, $h_{ie}(u_i)$ depends on $x_i, i \in \mathcal{V}_e$. We also highlight that (5.4) is parameterized in $\boldsymbol{x}$, which can be thought of as constant as for the optimization problem in (5.4) is concerned. Under Assumptions 5.2.2, 5.2.3, problem (5.4) is always feasible for all $x \in \mathcal{B}$. To begin with our analysis, we propose a relaxed version of (5.4) to guarantee feasibility of the local problems in the proposed distributed algorithm. This will be clarified in the sequel.

$$\begin{aligned} H^* = \min_{\boldsymbol{u}\in\mathcal{U},\boldsymbol{\rho}\geq 0} H(\boldsymbol{u},\boldsymbol{\rho}) := & \\ & \sum_{i=1}^{N}\left\{J_i(u_i) + \sum_{e\in\mathcal{C}_i}(\rho_{ie}^2 + M_i\rho_{ie})\right\} \\ \text{subject to } & \sum_{i\in\mathcal{V}_e} h_{ie}(u_i) \leq \sum_{i\in\mathcal{V}_e}\rho_{ie}, e = 1,\ldots,E. \end{aligned} \tag{5.7}$$

Feasibility of problem (5.7) is clear, as the positive variable $\boldsymbol{\rho}$ relaxes the linear constraints.

In view of an optimality analysis, we further impose the following constraint qualification assumption.

Assumption 5.2.4. *For every $\boldsymbol{x} \in \mathcal{B}$, there exists $\boldsymbol{u}(\boldsymbol{x})\in\mathcal{U}$, such that $\sum_{i\in\mathcal{V}_e} h_{ie} < 0$ for all $e = 1,\ldots,E$.*

Assumption 5.2.4 ensures strong duality for the nominal problem (5.4). As a result, there also exists $\boldsymbol{u}(\boldsymbol{x}) \in \mathcal{U}$ and $\boldsymbol{\rho} = 0$, such that $\sum_{i\in\mathcal{V}_e} h_{ie} < \sum_{i\in\mathcal{V}_e} \rho_{ie}$, for all $e \in \{1,\dots,E\}$. This demonstrates strong duality for the relaxed problem (5.7).

Optimality is analyzed in the following lemma.

Lemma 5.2.1. *Consider Assumptions 5.2.2, 5.2.3 and 5.2.4. Denote the minimizers of* (5.4) *and* (5.7)*, by* $\boldsymbol{u}^*_{\text{nom}}(\boldsymbol{x})$ *and* $(\boldsymbol{u}^*_{\text{rel}}(\boldsymbol{x}), \boldsymbol{\rho}^*)$*, respectively. Let* $\tilde{\boldsymbol{\mu}}^*$ *be an optimal dual variable associated with the CBF constraint in* (5.4)*. If*

$$M_i \geq \tilde{\mu}^*_e, \forall i \in \mathcal{V}_e, \forall e \in \{1,\dots,E\}, \tag{5.8}$$

where $\tilde{\mu}^*_e$ *is the e-th element of* $\tilde{\boldsymbol{\mu}}^*$*, then* $\boldsymbol{u}^*_{\text{rel}}(\boldsymbol{x}) = \boldsymbol{u}^*_{\text{nom}}(\boldsymbol{x})$*,* $\boldsymbol{\rho}^* = 0$*, and* $\tilde{\boldsymbol{\mu}}^*$ *is also an optimal dual solution of* (5.7)*.*

Proof. The dual function of the relaxed problem (5.7) is given by

$$\begin{aligned}
q(\boldsymbol{\mu}) :=& \inf_{\{u_i\in\mathcal{U}_i\}_{i=1}^N, \boldsymbol{\rho}\geq 0} \sum_{i=1}^N \left\{ J_i(u_i) + \sum_{e\in\mathcal{C}_i} (\rho_{ie}^2 + M_i\rho_{ie}) \right\} \\
&+ \sum_{e=1}^E \mu_e \left\{ \sum_{i\in\mathcal{V}_e} h_{ie}(u_i) - \sum_{i\in\mathcal{V}_e} \rho_{ie} \right\} \\
=& \inf_{\{u_i\in\mathcal{U}_i\}_{i=1}^N, \boldsymbol{\rho}\geq 0} \sum_{i=1}^N \left\{ J_i(u_i) + \sum_{e\in\mathcal{C}_i} \mu_e h_{ie}(u_i) \right\} \\
&+ \sum_{e=1}^E \sum_{i\in\mathcal{V}_e} \left\{ \rho_{ie}^2 + (M_i - \mu_e)\rho_{ie} \right\}. \\
=& \, q_{\text{nom}}(\boldsymbol{\mu}) + \inf_{\boldsymbol{\rho}\geq 0} \sum_{e=1}^E \sum_{i\in\mathcal{V}_e} \left\{ \rho_{ie}^2 + (M_i - \mu_e)\rho_{ie} \right\},
\end{aligned} \tag{5.9}$$

where

$$q_{\text{nom}}(\boldsymbol{\mu}) := \inf_{\{u_i\in\mathcal{U}_i\}_{i=1}^N} \sum_{i=1}^N \left\{ J_i(u_i) + \sum_{e\in\mathcal{C}_i} \mu_e h_{ie}(u_i) \right\} \tag{5.10}$$

is the dual function of the nominal problem (5.4). Let $\boldsymbol{\mu}^*$ be the maximizer for $q(\boldsymbol{\mu})$, and $\tilde{\boldsymbol{\mu}}$ be a maximizer of $q_{\text{nom}}(\boldsymbol{\mu})$.

Now consider the value of the second term in $q(\boldsymbol{\mu})$. If $\mu_e \leq M_i$ for some $e \in \{1,\dots,E\}, i \in \mathcal{V}_e$, then

$$\inf_{\rho_{ie}\geq 0} \rho_{ie}^2 + (M_i - \mu_e)\rho_{ie} = 0, \rho^*_{ie} = 0. \tag{5.11}$$

Otherwise if $\mu_e > M_i$, then

$$\inf_{\rho_{ie}\geq 0} \rho_{ie}^2 + (\mu_e - M_i)\rho_{ie} = -\frac{(M_i - \mu_e)^2}{4}, \rho_{ie}^* = \frac{\mu_e - M_i}{2}. \tag{5.12}$$

We first show if (5.8) holds, i.e. $M_i \geq \tilde{\mu}_e^*, \forall i \in \mathcal{V}_e, \forall e \in \{1, \ldots, E\}$, then

$$M_i \geq \mu_e^*, \forall i \in \mathcal{V}_e, \forall e \in \{1, \ldots, E\}, \tag{5.13}$$

where μ_e^* is the e-th element of $\boldsymbol{\mu}^*$, by contradiction. Suppose there exists $e \in \{1, \ldots, E\}, i \in \mathcal{V}_e$ such that $\mu_e^* > M_i$. Then, from (5.9) and (5.12) we have

$$q(\boldsymbol{\mu}^*) < q_{\text{nom}}(\boldsymbol{\mu}^*) \leq q_{\text{nom}}(\tilde{\boldsymbol{\mu}}^*).$$

The second inequality comes from a fact that $\tilde{\boldsymbol{\mu}}^*$ is a maximizer of $q_{\text{nom}}(\boldsymbol{\mu})$. However, from (5.9), (5.11), and (5.13) we have

$$q(\tilde{\boldsymbol{\mu}}^*) = q_{\text{nom}}(\tilde{\boldsymbol{\mu}}^*).$$

We conclude that $q(\tilde{\boldsymbol{\mu}}^*) > q(\boldsymbol{\mu}^*)$, thus reach a contradiction as $\boldsymbol{\mu}^*$ maximizes $q(\boldsymbol{\mu})$. By (5.13) and (5.11) we have $\boldsymbol{\rho}^* = 0$, any $\tilde{\boldsymbol{\mu}}^*$ that maximizes $q_{\text{nom}}(\boldsymbol{\mu})$ also maximizes $q(\boldsymbol{\mu})$.

As a direct result of (5.11) and (5.12), the second part of $q(\boldsymbol{\mu})$, which is $\inf_{\boldsymbol{\rho}\geq 0} \sum_{e=1}^{E} \sum_{i\in\mathcal{V}_e} \{\rho_{ie}^2 + (M_i - \mu_e)\rho_{ie}\}$, is concave and smooth. This is different from [160, Lemma III.2] where the dual function goes to $-\infty$ when $\mu_e > M_i$. Introducing a quadratic term for the relaxation variables enhances convexity of the primal function, hence smoothness of the dual function.

□

Lemma 5.2.1 establishes a lower bound for $M_i, i = 1, \ldots, N$, under which the optimal primal-dual solution of (5.7) coincides with that of (5.4). The lower bound is determined by the optimal dual solution $\tilde{\boldsymbol{\mu}}$ of the unrelaxed problem (5.4). Under Assumption 5.2.4, $\tilde{\boldsymbol{\mu}}^*$ is also bounded following [161, Lemma 1]. In practice, one can select a large enough $M_i, i = 1, \ldots, N$ to satisfy (5.8).

5.2.1 Full Control Law

We now design an algorithm to solve the centralized CBF-QP problem (5.4) in a distributed manner with guaranteed feasibility across iterations; see Algorithm 1. Since $h_{ie}(u_i)$ also depends on x_l for $l \in \mathcal{V}_e \backslash \{i\}$, an additional communication

Algorithm 1 Distributed Safe Control Design Algorithm for agent i at x_i

Initialization Arbitrary $\lambda_{il}^0, \forall l \in \mathcal{N}_i \cap \mathcal{V}_e$, $\forall e \in \mathcal{C}_i$.
Receive x_l for any $l \in \mathcal{N}_i \cap \mathcal{V}_e$,$e \in \mathcal{C}_i$
Send x_i to any $l \in \mathcal{N}_i \cap \mathcal{V}_e$, for $e \in \mathcal{C}_i$.
Output: Optimal control input u_i^*

1: **while** Not reaching convergence **do**
2: **Receive** λ_{il}^k from $\forall l \in \mathcal{N}_i \cap \mathcal{V}_e, \forall e \in \mathcal{C}_i$.
3: **Solve** $((u_i^k, \boldsymbol{\rho}_i^k), \boldsymbol{\mu}_i^k)$ as a primal-dual solution of the following optimization problem

$$\begin{aligned} \min_{u_i, \boldsymbol{\rho}_i} \; & J_i(u_i) + \sum_{e \in \mathcal{C}_i} (\rho_{ie}^2 + M_i \rho_{ie}) \\ \text{s.t. } & u_i \in \mathcal{U}_i, \rho_{ie} \geq 0, \\ & h_{ie}(u_i) + \sum_{l \in \mathcal{N}_i \cap \mathcal{V}_e} (\lambda_{il}^k - \lambda_{li}^k) \leq \rho_{ie}, \forall e \in \mathcal{C}_i. \end{aligned} \tag{5.14}$$

4: **Receive** μ_{le}^k from agent $l \in \mathcal{N}_i \cap \mathcal{V}_e$.
5: **Update** λ_{il} by

$$\lambda_{il}^{k+1} = \lambda_{il}^k - \gamma^k (\mu_{ie}^k - \mu_{le}^k). \tag{5.15}$$

6: **end while**

round at the beginning of the algorithm is designed. For all $i = 1, \ldots, N$, and $e \in \mathcal{C}_i$, agent i is to receive x_l from agent $l \in \mathcal{N}_i \cap \mathcal{V}_e$. Within a finite number of communication rounds, agent i can gather all the other agents' states in sub-networks $e \in \mathcal{C}_i$. Then, for any $e \in \mathcal{C}_i$, functions $h_{ie}(u_i)$ can be constructed as in (5.6).

There are two main computation and two communication steps in the algorithm. At the first computation step (Step 3), agent i solves the optimization problem (5.14) to obtain the optimal primal-dual solution $((u_i^k, \boldsymbol{\rho}_i^k), \boldsymbol{\mu}_i^k)$, where $\boldsymbol{\rho}_i$ includes relaxation variables denoted by ρ_{ie} (penalized in the cost by M_i), and $\boldsymbol{\mu}_i$ includes the dual variables μ_{ie}, for all $e \in \mathcal{C}_i$. In practice, μ_{ie} corresponds to the constraints

allocated to agent i, i.e. $h_{ie}(x_i)+\sum_{l\in\mathcal{N}_i\cap\mathcal{V}_e}(\lambda_{il}^k-\lambda_{li}^k)\leq\rho_{ie}$. Moreover, the constraints in the distributed problem (5.14) are relaxed by an additional non-negative relaxation variable ρ_{ie}. This guarantees the feasibility of the local optimization problem. However, this does not necessarily imply satisfaction of the CBF constraints in (5.4) by using $\boldsymbol{u}^{k+1}$.

The first computation step uses auxiliary variables λ_{il}^k and λ_{li}^k. The difference $\lambda_{il}^k-\lambda_{li}^k$ constitutes estimates of the neighbouring terms $h_{le}(u_l)$. λ_{il}^0 is initialized arbitrarily. As we will show in Theorem 5.2.1, the initialization will not influence convergence to the optimizer. Among all these variables, λ_{le}^k for $l\in\mathcal{N}_i\cap\mathcal{V}_e$ are updated and stored by neighbours. They are available to agent i via communication in Step 2. The second computation step is to update the local auxiliary variables (5). Part of the dual variables used in the update are received from the neighbours at Step 4. Here the update is a gradient-like procedure, with stepsize $\gamma^k>0$.

Remark 5.2.2. Algorithm 1 capitalizes on the primal-decomposition algorithm in [160, Algorithm RSDD], however, with several key extensions. First, the relaxation penalty in the cost includes a new quadratic term. This renders the cost function strongly convex, allowing for superior convergence properties and ensuring uniqueness of the minimizer across iterations. Moreover, for every agent $i\in\{1,\ldots,N\}$, each CBF constraint $e\in\mathcal{C}_i$ is relaxed by an individual relaxation variable ρ_{ie}. On the contrary, [160, Algorithm RSDD] uses one relaxation variable for all the constraints. Multiple relaxation variables enable stricter satisfaction of CBF constraints across iterations. This is especially important when a particular $\rho_{ie_1}^k$ is significantly larger than the other ones $\rho_{ie_2}^k, e_2\in\mathcal{C}_i\backslash e_1$. It should also be noted that Algorithm 1 is applicable to the case where $\mathcal{G}$ is divided into several sub-networks $\mathcal{G}_e, e\in\{1,\ldots,E\}$, while [160, Algorithm RSDD] only deals with a single network. This becomes of importance for multi-agent applications where safety constraints are typically defined on several sub-networks.

Among different types of distributed optimization algorithms, primal-decomposition methods, firstly proposed by [160, Algorithm RSDD] is selected here for its ability

to guarantee almost-safety across iterations. This is realized by allocating the auxiliary variables $\boldsymbol{\lambda}$, while balancing the safety requirement to every agent. We say "almost" here since additional relaxation variables are introduced in every local optimization problem for feasibility. In applications that require high control frequency, the algorithm may stop before reaching convergence. When the relaxation variables $\boldsymbol{\rho}^k = 0$ for a given $k > 0$, then for any $e \in \{1, \ldots, E\}$ we have that

$$\sum_{i \in \mathcal{V}_e} h_{ie}(u_i^k) = \sum_{i \in \mathcal{V}_e} \underbrace{\left\{ h_{ie}(u_i^k) + \sum_{l \in \mathcal{N}_i \cap \mathcal{V}_e} (\lambda_{il}^k - \lambda_{li}^k) \right\}}_{\leq 0} \leq 0,$$

which implies that the CBF constraints are satisfied with $\boldsymbol{u}^k$. The next theorem gives the convergence result.

Theorem 5.2.1. *Consider Assumptions 5.2.1, 5.2.2, 5.2.3, 5.2.4, and let $M_i \geq \tilde{\mu}_e$ for every $i = 1, \ldots, N$, $e \in \mathcal{C}_i$. For every agent $i = 1, \ldots, N$, and any bounded $\boldsymbol{\lambda}^0$,*

(a) if $\mathcal{U}_i \subset \mathbb{R}^{m_i}$. Choose the sequence $\{\gamma^k\}_{k \geq 0}$, with each $\gamma^k \geq 0$, and $\sum_{k=0}^{\infty} \gamma^k = \infty$, $\sum_{k=0}^{\infty} (\gamma^k)^2 < \infty$. Then we have $\lim_{k \to \infty} H(\boldsymbol{u}^k, \boldsymbol{\rho}^k) - J^* \to 0$, and $\boldsymbol{u}^k$ converges to the primal optimal solution of (5.4).

(b) if $\mathcal{U}_i = \mathbb{R}^{m_i}$, and for every $e \in \{1, \ldots, E\}$ $\sum_{i \in \mathcal{V}_e} h_{ie}(u_i)$ are linearly independent in $\boldsymbol{u}$. Let the step size $\gamma^k = \gamma > 0$ be a small constant. $H(\boldsymbol{u}^k, \boldsymbol{\rho}^k)$ converges to the optimal cost J^* in (5.4) *sublinearly, i.e. $H(\boldsymbol{u}^k, \boldsymbol{\rho}^k) - J^* \leq \frac{2||\boldsymbol{\lambda^0} - \boldsymbol{\lambda^*}||_2^2}{\gamma k}$, and $\boldsymbol{u}^k$ converges to the primal optimal solution of* (5.4).

Proof. We begin with (a). Under Assumption 5.2.4, strongly duality holds for the primal problem (5.4) and the dual problem (5.9). With a slight abuse of notation, we define

$$q_i(\boldsymbol{\mu}_i) := \inf_{\{u_i \in \mathcal{U}_i\}, \boldsymbol{\rho} \geq 0} \left\{ J_i(u_i) + \sum_{e \in \mathcal{C}_i} (\rho_{ie}^2 + M_i \rho_{ie}) + \sum_{e \in \mathcal{C}_i} \mu_{ie}(h_{ie}(u_i) - \rho_{ie}) \right\}. \tag{5.16}$$

By Assumption 5.2.1, we have $\mathcal{G}_e$ is undirected and connected for every $e \in \{1,\dots,E\}$. Therefore, suppose $\mu_{ie} = \mu_{je}$, $\forall e \in \{1,\dots,E\}, i \in \mathcal{V}_e$, $j \in \mathcal{N}_i \cap \mathcal{V}_e$, then we can deduce that

$$\mu_{ie} = \mu_{je}, \forall i,j \in \mathcal{V}_e. \tag{5.17}$$

Recalling that $i \in \{1,\dots,N\}$ is the numbering of agent, $e \in \mathcal{C}_i$ is the group of CBF constraint that includes agent i, $\mathcal{N}_i$ is the set of neighbouring agents for agent i, and $\mathcal{V}_e$ is the set of agents in group e. $\mathcal{N}_i \cap \mathcal{V}_e \neq \emptyset$ due to Assumption 5.2.1. The new variable μ_{ie} and μ_{le} can be regarded as local copies of μ_e by agent i and agent l, which is associated with the e-th CBF constraint for a group of agents. Using the decomposed dual function (5.16) and the new constraint (5.17), we come up with an equivalent decomposed dual problem

$$\begin{aligned} &\max_{\boldsymbol{\mu}_i \geq 0} \sum_{i=1}^{N} q_i(\boldsymbol{\mu_i}) \\ &\text{subject to } \mu_{ie} = \mu_{le}, \forall i \in \{1,\dots,N\}, e \in \mathcal{C}_i, l \in \mathcal{N}_i \cap \mathcal{V}_e, \end{aligned} \tag{5.18}$$

If $\mathcal{V}_e = \{1,\dots,N\}, \forall e \in \{1,\dots,E\}$, (5.18) is a generic dual decomposition problem [162, Section 3.1.3].

Consider the dual function of (5.18)

$$d(\boldsymbol{\lambda}) := \sum_{i=1}^{N} \sup_{\boldsymbol{\mu}_i \geq 0} \left(q_i(\boldsymbol{\mu}_i) + \sum_{e \in \mathcal{C}_i} \sum_{l \in \mathcal{N}_i \cap \mathcal{V}_e} \lambda_{il}^\top (\mu_{ie} - \mu_{le}) \right), \tag{5.19}$$

where λ_{il} is a free dual variable for the constraint $\mu_{ie} = \mu_{le}$ in (5.18). Recalling that the network $\mathcal{G}$ is undirected, for each $(i,l) \in \mathcal{E}$ we also have $(l,i) \in \mathcal{E}$. This indicates that in (5.19), we have both $\lambda_{il}^\top(\mu_{ie} - \mu_{le})$ and $\lambda_{li}^\top(\mu_{le} - \mu_{ie})$ for every given $i \in \{1,\dots,N\}, e \in \mathcal{C}_i, l \in \mathcal{N}_i \cap \mathcal{V}_e$. By gathering the terms involving $\boldsymbol{\mu}_i$ together, such as $\lambda_{il}^\top \mu_{ie}$ and $-\lambda_{li}^\top \mu_{ie}$, and doing some algebraic calculations, we obtain

$$d(\boldsymbol{\lambda}) = \sum_{i=1}^{N} \sup_{\boldsymbol{\mu}_i \geq 0} \left(q_i(\boldsymbol{\mu}_i) + \sum_{e \in \mathcal{C}_i} \mu_{ie}^\top \sum_{l \in \mathcal{N}_i \cap \mathcal{V}_e} (\lambda_{il} - \lambda_{li}) \right) \tag{5.20}$$

As (5.19) is traversing every $i \in \{1,\dots,N\}$, μ_{le} in (5.19) is also contained in (5.20), for $l \in \mathcal{V}_e$. A procedure similar to (5.19) and (5.20) has been proposed in [160, Section III.B] but only for one network $\mathcal{G}$. Our formulation generalizes these results to constraints defined on multiple sub-networks $\mathcal{G}_e$, for $e \in \{1,\dots,E\}$.

The dual problem of (5.18) is then given by

$$d^* = \min_{\boldsymbol{\lambda}} d(\boldsymbol{\lambda}). \tag{5.21}$$

Strong duality holds between problem (5.18) and (5.21) since (5.18) is an linear equality constrained concave problem. Therefore, solving problem (5.21) leads to the optimal solution of problem (5.18). Solving problem (5.21) has advantages in terms of distributed computation. This can be seen by applying the gradient descent method to solve (5.21). From (5.19), for every $i \in \{1, \ldots, N\}$, $e \in \mathcal{C}_i$, and $l \in \mathcal{N}_i \cap \mathcal{V}_e$, the gradient $\nabla d(\lambda_{il})$ is given by

$$\nabla d(\lambda_{il}) = \mu_{ie} - \mu_{le}. \tag{5.22}$$

At iteration k, each agent i performs two steps:

- (i) for every $e \in \mathcal{C}_i$, $l \in \mathcal{N}_i \cap \mathcal{V}_e$, calculate the gradient $\nabla d(\lambda_{il}^k)$: receive λ_{li}^k, $l \in \mathcal{N}_i \cap \mathcal{V}_e$, and compute μ_{ie} by solving

$$\max_{\boldsymbol{\mu}_i \geq 0} \left(q_i(\boldsymbol{\mu}_i) + \sum_{e \in \mathcal{C}_i} \mu_{ie}^\top \sum_{l \in \mathcal{N}_i \cap \mathcal{V}_e} (\lambda_{il}^k - \lambda_{li}^k) \right). \tag{5.23}$$

- (ii) use gradient descent: for every $e \in \mathcal{C}_e, l \in \mathcal{N}_i \cap \mathcal{V}_e$, receive μ_{le}^k and update λ_{il} by (5.22):

$$\lambda_{il}^{k+1} = \lambda_{il}^k - \gamma^k (\mu_{ie}^k - \mu_{le}^k). \tag{5.24}$$

(5.24) is Step 5 of Algorithm 1. We then show that solving (5.23) is equivalent to solving (5.14) at Step 3. For every $i \in \{1, \ldots, N\}$, dualizing the CBF constraints in (5.14) by $\boldsymbol{\mu}_i \geq 0$ yields a dual problem

$$\begin{aligned}
&\max_{\boldsymbol{\mu}_i \geq 0} \inf_{\{u_i \in \mathcal{U}_i\}, \boldsymbol{\rho} \geq 0} \Bigg\{ J_i(u_i) + \sum_{e \in \mathcal{C}_i} (\rho_{ie}^2 + M_i \rho_{ie}) \\
&+ \sum_{e \in \mathcal{C}_i} \mu_{ie}(h_{ie}(u_i) - \rho_{ie}) \Bigg\} + \sum_{e \in \mathcal{C}_i} \mu_{ie}^\top \sum_{l \in \mathcal{N}_i \cap \mathcal{V}_e} (\lambda_{il}^k - \lambda_{li}^k) \\
&\overset{(5.16)}{=} \max_{\boldsymbol{\mu}_i \geq 0} \left(q_i(\boldsymbol{\mu}_i) + \sum_{e \in \mathcal{C}_i} \mu_{ie}^\top \sum_{l \in \mathcal{N}_i \cap \mathcal{V}_e} (\lambda_{il}^k - \lambda_{li}^k) \right),
\end{aligned} \tag{5.25}$$

which is (5.23). Therefore, Steps 2-5 in Algorithm 1 involve performing gradient descent to solve problem (5.21) in a distributed manner.

Diminishing step-size is used here as [160]. Specifically, (5.23) is the dual problem of (5.14). Strong duality holds for large enough $\boldsymbol{\rho}$ as the relaxed CBF constraints hold strictly. Updating (5.24) is the same as (5.15) for every agent across iterations. Given that $d(\boldsymbol{\lambda})$ is convex, gradient descent guarantees that $d(\boldsymbol{\lambda}^k)$ convergence to the optimal value $d^* = J^*$ since strong duality holds between (5.7) and (5.18), as well as (5.18) and (5.21). Moreover, the relaxed problem (5.7) is strongly (hence also strictly) convex, which indicates uniqueness of the optimal solution $(\boldsymbol{u}^*_{\text{rel}}, \boldsymbol{\rho}^*)$. Using Lemma 5.2.1, we obtain $\boldsymbol{u}^*_{\text{rel}} = \boldsymbol{u}^*_{\text{nom}}$, which is the optimal solution of (5.4).

We then prove (b). First we prove that $q(\boldsymbol{\mu})$ in (5.9) is a concave quadratic function. When every $\mathcal{U}_i = \mathbb{R}^{m_i}$ and the CBF constraints are linearly independent, the relaxed CBF-QP (5.7) is a linearly constrained strongly convex quadratic problem. Following the example [163, Section 5.2.4, Eq. 5.28][1], $q_{\text{nom}}(\boldsymbol{\mu})$ in (5.10) is a strongly concave quadratic function. Together with (5.9), (5.11) and (5.12), we conclude that $q(\boldsymbol{\mu})$ is a strongly concave and smooth function. From duality between strong concavity (convexity) and smoothness [164, Theorem 6], $d(\boldsymbol{\lambda})$ is a smooth and necessarily convex function. Using constant step size

$$0 < \gamma < \frac{1}{2L}, \tag{5.26}$$

where L is Lipschitz constant of $\nabla d(\boldsymbol{\lambda})$, in a gradient descent method to minimize a smooth and convex function $d(\boldsymbol{\lambda})$, the generated iterates converge sublinearly as

$$\begin{aligned} d(\boldsymbol{\lambda}^k) - J^* &\leq \frac{2(d(\boldsymbol{\lambda}^0) - J^*)||\boldsymbol{\lambda^0} - \boldsymbol{\lambda^*}||_2^2}{2||\boldsymbol{\lambda}^0 - \boldsymbol{\lambda}^*||_2^2 + k\gamma(2 - L\gamma)(d(\boldsymbol{\lambda}^0) - J^*)} \\ &\leq \frac{2(d(\boldsymbol{\lambda}^0) - J^*)||\boldsymbol{\lambda^0} - \boldsymbol{\lambda^*}||_2^2}{k\gamma(d(\boldsymbol{\lambda}^0) - J^*)} \leq \frac{2||\boldsymbol{\lambda}^0 - \boldsymbol{\lambda}^*||_2^2}{k\gamma}. \end{aligned} \tag{5.27}$$

The first inequality is proved by [165, Theorem 2.1.14], the second one comes from eliminating the term $||\boldsymbol{\lambda}^0 - \boldsymbol{\lambda}^*||_2^2$ from the denominator, and considering $2 - L\gamma \geq 1$ from (5.26).

[1]The example demonstrates that the dual function of a convex quadratically constrained quadratic programming problem is a concave quadratic function. Our problem is as a special case where the quadratic terms are zero in the constraints.

Recalling the expression of $d(\boldsymbol{\lambda})$ from (5.16), and the duality result from (5.25), we have

$$
\begin{aligned}
& d(\boldsymbol{\lambda}^k) = \\
& \sum_{i=1}^{N} \inf_{u_i, \boldsymbol{\rho}_i \geq 0} \left(\sup_{\boldsymbol{\mu}_i \geq 0} \left(J_i(u_i) + \sum_{e \in \mathcal{C}_i} \left(\rho_{ie}^2 + M_i \rho_{ie} \right) \right) \right. \\
& \left. + \sum_{e \in \mathcal{C}_i} \mu_{ie}(h_{ie}(u_i) - \rho_{ie}) + \sum_{e \in \mathcal{C}_i} \mu_{ie}^\top \sum_{l \in \mathcal{N}_i \cap \mathcal{V}_e} (\lambda_{il}^k - \lambda_{li}^k) \right) \\
& = \sum_{i=1}^{N} \left(J_i(u_i^k + \sum_{e \in \mathcal{C}_i} ((\rho_{ie}^k)^2 + M_i \rho_{ie}^k) \right) \\
& = \sum_{i=1}^{N} ||u_i^k - u^{\text{des}}||^2 + \rho_{\text{sum}}^k = H(\boldsymbol{u}^k, \boldsymbol{\rho}^k). \qquad (5.28)
\end{aligned}
$$

Hence, by (5.27) and (5.28), we conclude that $H(\boldsymbol{u}^k, \boldsymbol{\rho}^k) - J^* < \frac{2||\boldsymbol{\lambda}^0 - \boldsymbol{\lambda}^*||_2^2}{\gamma k}$.

□

Given that (5.4) is guaranteed to be feasible under Assumption 5.2.3, the optimal controller designed by Algorithm 1 is guaranteed to satisfy all the CBF constraints. However, this does not necessarily hold for $\boldsymbol{u}^k(\boldsymbol{x})$ with arbitrary k, if $\boldsymbol{\rho}^k(\boldsymbol{x}) \neq 0$. However, terminating the algorithm early, and considering $\boldsymbol{u}^k(\boldsymbol{x})$ at the time of termination has many benefits in terms of reducing computation and communication complexity. This motivates the analysis of a truncated algorithm as presented in the next section.

5.2.2 Truncated Control Law

Algorithm 1 can be implemented in a distributed fashion with ensured safety and optimality properties, however, it may not be suitable for control tasks that require high control frequency, i.e. multi-robot system control, as its theoretical properties are established in an asymptotic manner. This motivates the use of a *truncated algorithm*, Algorithm 2, where the algorithm terminates after a finite number of iterations, denoted by η.

Algorithm 2 Truncated Distributed Safe Control Design Algorithm for agent i

Initialization Predefined $\lambda_{il}^0, \forall l \in \mathcal{N}_i \cap \mathcal{V}_e$, $\forall e \in \mathcal{C}_i$, truncated parameter $\eta \in \mathbb{N}$
Receive x_l for any $l \in \mathcal{N}_i \cap \mathcal{V}_e$,$e \in \mathcal{C}_i$
Send x_i to any $l \in \mathcal{N}_i \cap \mathcal{V}_e, e \in \mathcal{C}_i$
Output: Optimal control input u_i^*

1: **while** $k \leq \eta$ **do**
2: steps 2, 3, 4 in Algorithm 1
3: step 5 in Algorithm 1
4: **end while**

Algorithm 2 is computationally more efficient compared to Algorithm 1, at the cost of potentially violating the control barrier function constraints. The violations are reflected in the non-zero relation variables $\boldsymbol{\rho}^\eta(\boldsymbol{x})$. In general, it is challenging to provide an explicit bound for η, under which $\boldsymbol{\rho}^\eta(\boldsymbol{x}) = 0$, as the distributed algorithm converges asymptotically as per Theorem 5.2.1. Moreover, $\boldsymbol{\rho}^\eta(\boldsymbol{x})$ depends on the state $\boldsymbol{x} \in \mathcal{H} := \mathcal{X} \cap \mathcal{B}$, which parameterizes the optimization problem (5.7). To quantify safety of the multi-agent system (5.1) with $\boldsymbol{u}(\boldsymbol{x}) = \boldsymbol{u}^\eta(\boldsymbol{x})$, we study the problem of safety verification by means of CBFs. This is established in the following section.

5.3 Distributed Safety Verification

In this section we show how to verify safety for a multi-agent system for any $\boldsymbol{x} \in \mathcal{H}$, using the truncated controller $\boldsymbol{u}^\eta(\boldsymbol{x})$ designed by Algorithm 2. The verification is conducted by checking the *risk* of becoming unsafe along the current trajectories by means of CBFs. We would like to measure the violations of the CBF constraints for the multi-agent system (5.1), under the control law $\boldsymbol{u}^\eta(\boldsymbol{x})$. However, this problem becomes challenging as $\text{Int}(\mathcal{H}) \neq \emptyset$, and one would need to verify a safety property for an uncountable number of points. Instead of verifying this for any $\boldsymbol{x} \in \mathcal{H}$, we propose to verify over a *finite* number of scenarios, i.e. samples of $\boldsymbol{x}$, from $\mathcal{H}$. Notice that the multi-agent system under consideration (see (5.1)) is deterministic; however, we draw scenarios as a discrete approximation of $\mathcal{H}$. The scenario approach [128] provides then the theoretical foundation for quantifying the probability that the solution that satisfies our safety property for a finite number of

scenarios, satisfies this property when it comes to yet another realization of $\boldsymbol{x} \in \mathcal{H}$. Such a generalization property is in turn probabilistic, with a probability measure implicitly defined using the mechanism employed to draw scenarios (see Section 5.3.2).

We note here the analysis conducted in this section can be applied to, but not limited to the controller designed using Algorithm 2. The only requirement for the verified controller $\boldsymbol{u}(\boldsymbol{x})$ is locally Lipschitz continuous, which is necessary for the solution of the multi-agent system to be unique. We also highlight that in this section a CBF is only regarded as a verification criterion but not necessarily as a control design principle.

5.3.1 Scenario Based Safety Verification

Consider an N-agent system (5.1) and a safe invariant set $\mathcal{B}$. Our objective is to verify whether all the CBF constraints are satisfied for the multi-agent system (5.1) using $\boldsymbol{u}(\boldsymbol{x})$, for any $\boldsymbol{x} \in \mathcal{H}$. A new set $\mathcal{Z}_{\boldsymbol{x}}$ is introduced to represent the satisfaction of all the CBF constraints.

$$\mathcal{Z}_{\boldsymbol{x}} := \left\{ \boldsymbol{z} : \sum_{i \in \mathcal{V}_e} h_{ie}(u_i(\boldsymbol{x})) \leq \sum_{i \in \mathcal{V}_e} z_{ie}, \forall e \in \{1, \ldots, E\} \right\}. \tag{5.29}$$

Then, if $0 \in \mathcal{Z}_{\boldsymbol{x}}, \forall \boldsymbol{x} \in \mathcal{H}$, we conclude that all CBF constraints are satisfied using $\boldsymbol{u}(\boldsymbol{x})$, for any $\boldsymbol{x} \in \mathcal{H}$. With a slight abuse of notation, we define $\mathcal{Z}^i_{\boldsymbol{x}}$ as

$$\mathcal{Z}^i_{\boldsymbol{x}} := \left\{ \boldsymbol{z} : \sum_{k \in \mathcal{V}_e} h_{ke}(u_k(\boldsymbol{x})) \leq \sum_{k \in \mathcal{V}_e} z_{ke}, \forall e \in \mathcal{C}_i \right\} \tag{5.30}$$

to represent the satisfaction of CBF constraints that involve agent i, for every $i \in \{1, \ldots, N\}$. If $0 \in \mathcal{Z}^i_{\boldsymbol{x}}$, $\forall \boldsymbol{x} \in \mathcal{H}$, the CBF constraints that involve agent i are satisfied using $\boldsymbol{u}(\boldsymbol{x})$. Conversely, if $0 \notin \mathcal{Z}^i_{\boldsymbol{x}}$, at least one CBF constraint that involves agent i is violated, for some $\boldsymbol{x} \in \mathcal{H}$. Therefore, $\mathcal{Z}_{\boldsymbol{x}}$ can be expressed as

$$\mathcal{Z}_{\boldsymbol{x}} = \bigcap_{i=1}^{N} \mathcal{Z}^i_{\boldsymbol{x}}. \tag{5.31}$$

We propose a scenario-based safety verification program as follows.

$$\begin{aligned}\min_{\boldsymbol{z}\geq 0,\boldsymbol{\zeta}\geq 0}\ & \sum_{i=1}^{N}\sum_{e\in\mathcal{C}_i}\left(z_{ie}^2+M\sum_{r=1}^{R}\zeta_{ie}^{(r)}\right)\\ \text{s.t.}\ & \sum_{i\in\mathcal{V}_e}h_{ie}(u_i(\boldsymbol{x}^{(r)}))\leq\sum_{i\in\mathcal{V}_e}(z_{ie}+\zeta_{ie}^{(r)}),\\ & e=1,\ldots,E, r=1,\ldots,R,\end{aligned}\tag{5.32}$$

where scenarios $\boldsymbol{x}^{(r)}\in\mathcal{H}$ for any $r=1,\ldots,R$ are extracted according to some probability distribution to be clarified in the sequel. Throughout the section $\bar{X}=\{\boldsymbol{x}^{(1)},\ldots,\boldsymbol{x}^{(R)}\}$ denotes the set of scenarios, where $\boldsymbol{x}^{(r)}=[(x_1^{(r)})^\top,\ldots,(x_N^{(r)})^\top]^\top\in\mathbb{R}^{\sum_{i=1}^{N}n_i}$, for $r=1,\ldots,R$, and R is the number of scenarios. Relaxation variables $\boldsymbol{\zeta}$ are introduced, while $M>0$ is a penalty coefficient. Let $(\boldsymbol{z}^*(\boldsymbol{x}),\boldsymbol{\zeta}^*(\boldsymbol{x}))$ denote the optimal solution of (5.32). In the sequel, we drop the dependency of $\boldsymbol{x}$ for simplicity.

Program (5.32) is a data-driven QP, where all the constraints are linear based on the samples. If for any scenario $\boldsymbol{x}^{(r)}, r=1,\ldots,R$, and the corresponding control input $\boldsymbol{u}(\boldsymbol{x})$, all the CBF constraints are satisfied, then $\boldsymbol{\zeta}^*=0$. Conversely, $\boldsymbol{\zeta}^*\neq 0$ represents a CBF constraint violation, and indicates the risk of being unsafe by means of CBF, up to level $\boldsymbol{z}^*$. Following Definition 2.3.1, the violation probability for (5.32) is defined by

$$V(\boldsymbol{z}):=\mathbb{P}\left\{\boldsymbol{x}\in\mathcal{H}:\boldsymbol{z}\notin\mathcal{Z}_{\boldsymbol{x}}\right\}.\tag{5.33}$$

Then, $V(\boldsymbol{z}^*)=\mathbb{P}\left\{\boldsymbol{x}\in\mathcal{H}:\boldsymbol{z}^*\notin\mathcal{Z}_{\boldsymbol{x}}\right\}$ represents the probability of at least one CBF constraint is violated up to $\boldsymbol{z}^*$, for any $\boldsymbol{x}\in\mathcal{H}$. Our goal is to *distributedly* characterize the violation probability $V(\boldsymbol{z}^*)$ using *finite* scenarios, i.e. samples of $\boldsymbol{x}$ from $\mathcal{H}$.

5.3.2 Sampling the Scenarios

The scenarios are sampled independently from the set $\mathcal{H}$. For sampling we define a probability density $\pi(\boldsymbol{x})$ associated with set $\mathcal{H}$ that satisfies $\int_{\mathcal{H}}\pi(\boldsymbol{x})d\boldsymbol{x}=1$. One

typical choice of $\pi(\boldsymbol{x})$ is to set it according to the density of the uniform distribution, i.e., $\pi(x) = \pi^{\text{uni}}(\boldsymbol{x}) = \frac{1}{\int_{\mathcal{H}} d\boldsymbol{x}}$.

The existence of $\pi^{\text{uni}}(\boldsymbol{x})$ is assured as $\mathcal{H}$ is a non-empty and compact set, due to Assumption 5.2.2. Then, $\boldsymbol{x}$ can be sampled R times independently from the distribution $\pi^{\text{uni}}(\boldsymbol{x})$. Note that the choice of the probability distribution does not affect the probabilistic results established in the sequel due to the distribution-free nature of scenario approach [128, Section 3.1]. Although the uniform distribution here is well-defined, the set $\mathcal{H}$ is defined implicitly as an intersection of multiple sets. Sampling a point from the proposed uniform distribution is rather arduous in practice, and every agent may not have access to $\mathcal{H}$. Here, we provide a sequential algorithm to sample scenarios $\boldsymbol{x}^{(r)}$, $r = 1, \ldots, R$.

Algorithm 3 Scenarios Sampling Algorithm

Initialization Set $\mathcal{H} = \mathcal{B} \cap \mathcal{X}$, failed times $F = 0$.
Output: Scenario $\boldsymbol{x}^{(r)}$.
1: Sample $x_1^{(r)}$ from $\pi_1(\boldsymbol{x})$.
2: **for** $i = 2, \ldots, N$ **do**
3: Construct a set $\mathcal{H}_i = \bigcap_{e \in \mathcal{C}_i} \mathcal{H}_{ie}$ following (5.34).
4: **if** $\mathcal{H}_i = \emptyset$ **then**
5: $F \leftarrow F + 1$.
6: go to $i = i - F$ ($i = 1$ is step 1).
7: **end if**
8: Sample $x_i^{(r)}$ from distribution $\pi_i = \frac{1}{\int_{\mathcal{X}_i} dx}$.
9: **while** $x_i^{(r)} \notin \mathcal{H}_i$ **do**
10: Sample $x_i^{(r)}$ from distribution π_i.
11: **end while**
12: **end for**

The algorithm constructs the densities from which samples are extracted sequentially for each agent. We first define the sets from which samples are extracted for agent i with part of the states of agents in the same sub-network $\mathcal{G}_e$ fixed.

$$\mathcal{H}_{ie} = \begin{cases} \mathcal{X}_i, \text{ if } \exists l \in \mathcal{V}_e, \text{such that } l > i \\ \{x_i \in \mathcal{X}_i | b_e(x_i, \{x_l^{(r)}\}) \geq 0)\}, \text{ otherwise} \end{cases} \tag{5.34}$$

Let $\mathcal{H}_i := \bigcap_{e \in \mathcal{C}_i} \mathcal{H}_{ie}$. The parameters in (5.34) can all be collected by local communication, since only states of agents in the same sub-network are required.

At Step 1, the first scenario $x_1^{(r)}$ associated with agent 1 is sampled from distribution $\pi_1 = \frac{1}{\int_{\mathcal{X}} dx}$, since now there are no other agents involved to restrict the set for agent 1. Then, the sampling-construction procedures repeat sequentially from agent 2 to agent N. For $i = 2, \ldots, N$, before sampling the scenario $x_i^{(r)}$, we first check whether $\mathcal{H}_i$ is empty (Step 4). If $\mathcal{H}_i = \emptyset$ (Step 5), then go back to the sampling-construction of agent $i - F$, $F \neq 1$ is to avoid a deadlock on step i. The deadlock happens when for given scenarios $x_1^{(r)}, \ldots, x_{i-2}^{(r)}$, the set $\mathcal{H}_{i-1}$ is such that for any $x_{i-1}^{(r)} \in \mathcal{H}_{i-1}$, $\mathcal{H}_i = \emptyset$. It is guaranteed that $F \leq i - 1$ for $i \geq 2$, since $\mathcal{H}_1 = \mathcal{X}_1 \neq \emptyset$. After finding feasible scenarios $x_1^{(r)}, \ldots, x_{i-1}^{(r)}$, we sample the scenario $x_i^{(r)}$ for the ith agent from the uniform distribution π_i (Step 8). The sampled scenario is then checked at Step 9. If $x_i^{(r)} \notin \mathcal{H}_i$, it will be sampled again following π_1. The loop will terminate in finite time since $\text{Int}(\mathcal{H}_i \cap \mathcal{X}) \neq \emptyset \ \forall i \in \{1, \ldots, N\}$.

Proposition 5.3.1. Consider Assumptions 5.2.1, 5.2.2, and assume scenarios $\boldsymbol{x}^{(r)}$, $r = 1, \ldots, R$ are sampled using Algorithm 3. We then have that $\boldsymbol{x}^{(r)} \in \mathcal{H}$, for all $r = 1, \ldots, R$. Moreover, all scenarios are independently and identically sampled.

Proof. The feasibility result holds directly from the definition of every set $\mathcal{H}_i$ in (5.34) that $x_i^{(r)}$ is sampled from. As a result, we have $b_{ie}(x_i^{(r)}, \{x_k^{(r)}\}) \geq 0$ for any $i = 1, \ldots, N$, $e \in \mathcal{C}_i$, and $k \in \mathcal{V}_e \backslash i$. Therefore, $\boldsymbol{x}^{(r)} \in \mathcal{H}$. Moreover, for all $r = 1, \ldots, R$, $\boldsymbol{x}^{(r)}$ are independent since $x_1^{(r)}$, $r = 1, \ldots, R$ are independently sampled from the distribution π_1.

At Step 6, when $F = i - 1$, it returns Step 1 to resample $x_1^{(r)}$. This happens when there exists $e \in \mathcal{C}_2$, and b_e is defined only on Agent 1 and 2, such that $\mathcal{H}_{2e} = \{x_2 \in \mathcal{X}_2 | b_e(x_2, x_1^{(r)}) \geq 0\} = \emptyset$. $x_1^{(r)}$ will then be resampled from the distribution π_1 to make $\mathcal{H}_{2e} \neq \emptyset$. Therefore, the actual distribution $\tilde{\pi}_1$ from which $x_1^{(r)}$ is sampled is defined on a set $\tilde{\mathcal{X}}_1 \in \mathcal{X}_1$, which satisfies

$$\{x_2 \in \mathcal{X}_2 | b_e(x_2, x_1^*) \geq 0\} \neq \emptyset, \forall x_1^* \in \tilde{\mathcal{X}}_1. \tag{5.35}$$

It trivially holds that $\text{Int}(\tilde{\mathcal{X}}_1) \neq \emptyset$ since $\text{Int}(\mathcal{H}) \neq \emptyset$, from Assumption 5.2.2. $\tilde{\pi}_1$ can be different from π_1, but is identical for every $r = 1, \ldots, R$. Similarly,

the resampling mechanism implicitly defines distributions $\tilde{\pi}_2, \ldots, \tilde{\pi}_N$ that may be different from $\pi_2, \ldots, \pi_N$. But these distributions are identical for scenarios $\boldsymbol{x}^{(r)}, r = 1, \ldots, R$. □

We note here that the elements in $\boldsymbol{x}^{(r)}$ are correlated, but this will not influence the independence results in Proposition 5.3.1 since we seek independence across r.

5.3.3 Distributed Safety Verification

After sampling scenarios $\boldsymbol{x}^{(r)}$, $r = 1, \ldots, R$ using Algorithm 3, we are at the stage of solving the safety verification program (5.32).

Letting the local cost function $J_i(\boldsymbol{z}_i, \boldsymbol{\zeta}_i)$, and constraint function $\hat{h}_{ie}(\boldsymbol{z}_i, \boldsymbol{\zeta}_i)$ be

$$\begin{aligned} J_i(\boldsymbol{z}_i, \boldsymbol{\zeta}_i) &= \sum_{e \in \mathcal{C}_i} \left(z_{ie}^2 + \sum_{r=1}^{R} \zeta_{ie}^{(r)} \right), \\ \hat{h}_{ie}^{(r)}(\boldsymbol{z}_i, \boldsymbol{\zeta}_i) &= h_{ie}(u_i(\boldsymbol{x}^{(r)})) - z_{ie} - \zeta_{ie}^{(r)}, r = 1, \ldots, R, \end{aligned} \tag{5.36}$$

Algorithm 1 can be applied to solve the distributed scenario optimization problem (5.32). The relaxation variables in Algorithm 1 are unnecessary, since every optimization sub-problem across iteration is solvable. We then have the following theorem as the main result on distributed probabilistic safety. The following theorem constitutes the multi-agent counterpart of Theorem 2.3.1. Using the density functions constructed in Algorithm 3 and considering Assumption 5.2.2, there will be no repeated scenarios for $r = 1, \ldots, R$. Therefore, eliminating all the constraints that are not of support for (5.32) will not change the optimal solution $\boldsymbol{z}^*$, and hence due to Assumption 5.2.2, the non-degeneracy requirement of Assumption 2.3.1 is satisfied

Theorem 5.3.1. *Let Assumptions 5.2.1 and 5.2.2 hold. Consider the optimization problem*

(5.32)*, let* $(\boldsymbol{z}^*, \{\boldsymbol{\zeta}^{*,(r)}\}_{r=1}^{R})$ *be the optimal solution. Choose* $\beta_i \in (0, 1), i = 1, \ldots, N$*, and set* $\beta = \sum_{i=1}^{N} \beta_i$*. For* $i = 1, \ldots, N$*, and* $0 \leq k_i \leq R-1$*, consider the polynomial*

equation in the t_i variable

$$\begin{aligned}&\binom{R}{k_i} t_i^{R-k_i} - \frac{\beta_i}{2R} \sum_{j=k_i}^{R-1} \binom{j}{k_i} t_i^{j-k_i} \\ &- \frac{\beta_i}{6R} \sum_{j=R+1}^{4R} \binom{j}{k_i} t_i^{j-k_i} = 0,\end{aligned} \tag{5.37}$$

while for $k_i = R$ consider the polynomial equation

$$1 - \frac{\beta}{6N} \sum_{j=R+1}^{4R} \binom{j}{k_i} t_i^{j-R} = 0. \tag{5.38}$$

For every $i = 1, \dots, N$ and any $k_i = 0, \dots, R-1$, Equation (5.37) has exactly two solutions in $[0, +\infty)$ denoted by $\underline{t}_i(k_i)$ and $\bar{t}_i(k_i)$, where $\underline{t}_i(k_i) \leq \bar{t}_i(k_i)$. Instead, Equation (5.38) has only one solution in $[0, +\infty)$, which we denote with $\bar{t}(R)$, while we define $\underline{t}(R) = 0$. Let $\underline{\epsilon}_i(k_i) := \max\{0, 1 - \bar{t}_i(k_i)\}$, $\bar{\epsilon}_i(k_i) := 1 - \underline{t}_i(k_i)$, and $\underline{\epsilon}(s^*) = \sum_{i=1}^N \underline{\epsilon}_i(s_i^*)$, $\bar{\epsilon}(s^*) = \min\{\sum_{i=1}^N \bar{\epsilon}_i(s_i^*), 1\}$. We then have that

$$\mathbb{P}^R \left\{ \frac{\underline{\epsilon}(s^*)}{N} \leq V(\boldsymbol{z}^*) \leq \bar{\epsilon}(s^*) \right\} \geq 1 - \beta, \tag{5.39}$$

where s_i^* is the number of $\boldsymbol{x}^{(r)}$'s for which there exists $e \in \mathcal{C}_i$, such that $\sum_{k \in \mathcal{V}_e} h_{ke}(u_k(\boldsymbol{x}^{(r)})) \geq \sum_{k \in \mathcal{V}_e} z_{ke}^*$. Recalling Equation (5.33), the violation probability $V(\boldsymbol{z}^*)$ is defined by $V(\boldsymbol{z}^*) = \mathbb{P}\{\boldsymbol{x} \in \mathcal{H} : \boldsymbol{z}^* \notin \mathcal{Z}_{\boldsymbol{x}}\}$.

Proof. We have that

$$\begin{aligned}&\mathbb{P}^R \left\{ \frac{\sum_{i=1}^N \underline{\epsilon}_i(s_i^*)}{N} \leq \mathbb{P}\left\{\boldsymbol{x} \in \mathcal{H} : \boldsymbol{z}^* \notin \mathcal{Z}_{\boldsymbol{x}}\right\} \leq \sum_{i=1}^N \bar{\epsilon}_i(s_i^*) \right\} \\ &= \mathbb{P}^R \left\{ \frac{1}{N} \sum_{i=1}^N \underline{\epsilon}_i(s_i^*) \leq \mathbb{P}\left\{ \boldsymbol{x} \in \mathcal{H} : \right.\right. \\ &\left.\left. \exists i \in \{1, \dots, N\},\, \boldsymbol{z}^* \notin \mathcal{Z}_{\boldsymbol{x}}^i \right\} \leq \sum_{i=1}^N \bar{\epsilon}_i(s_i^*) \right\} \\ &= \mathbb{P}^R \left\{ \frac{1}{N} \sum_{i=1}^N \underline{\epsilon}_i(s_i^*) \leq \mathbb{P}\left\{ \boldsymbol{x} \in \mathcal{H} : \bigcup_{i=1}^N \left\{ \boldsymbol{z}^* \notin \mathcal{Z}_{\boldsymbol{x}}^i \right\} \right\} \right. \\ &\left. \bigcap \mathbb{P}\left\{ \bigcup_{i=1}^N \left\{ \boldsymbol{x} \in \mathcal{H} : \boldsymbol{z}^* \notin \mathcal{Z}_{\boldsymbol{x}}^i \right\} \right\} \leq \sum_{i=1}^N \bar{\epsilon}_i(s_i^*) \right\}\end{aligned} \tag{5.40}$$

The second equation comes from the fact that $\boldsymbol{z}^* \in \mathcal{Z}_{\boldsymbol{x}}$ is equivalent to $\boldsymbol{z}^* \in \mathcal{Z}_{\boldsymbol{x}}^i$ $\forall i \{1, \dots, N\}$. The second equation changes $\exists i \in \{1, \dots, N\}, \boldsymbol{z}^* \notin \mathcal{Z}_{\boldsymbol{x}}^i$ into

$\bigcup_{i=1}^{N}\{\boldsymbol{z}^* \notin \mathcal{Z}_{\boldsymbol{x}}^i\}$. Similar tricks have been used in [166, Equation 15] to derive an upper bound for the inner probability. Here we extend the results to both upper and lower bounds, using Theorem 2.3.1. We separately deal with the two bounds on the probability. For the upper bound we have

$$\begin{aligned}&\mathbb{P}^R\left\{\mathbb{P}\left\{\bigcup_{i=1}^{N}\left\{\boldsymbol{x}\in\mathcal{H}:\boldsymbol{z}^*\notin\mathcal{Z}_{\boldsymbol{x}}^i\right\}\right\}\leq\sum_{i=1}^{N}\bar{\epsilon}_i(s_i^*)\right\}\\ \geq&\mathbb{P}^R\left\{\sum_{i=1}^{N}\mathbb{P}\left\{\boldsymbol{x}\in\mathcal{H}:\boldsymbol{z}^*\notin\mathcal{Z}_{\boldsymbol{x}}^i\right\}\leq\sum_{i=1}^{N}\bar{\epsilon}_i(s_i^*)\right\}.\end{aligned}$$

The equality is achieved when for any $i\neq j$, $\boldsymbol{z}^*\notin\mathcal{Z}_{\boldsymbol{x}}^i$ and $\boldsymbol{z}^*\notin\mathcal{Z}_{\boldsymbol{x}}^j$ are mutually exclusive. For the lower bound we have

$$\begin{aligned}&\mathbb{P}^R\left\{\frac{1}{N}\sum_{i=1}^{N}\underline{\epsilon}_i(s_i^*)\leq\mathbb{P}\left\{\boldsymbol{x}\in\mathcal{H}:\bigcup_{i=1}^{N}\left\{\boldsymbol{z}^*\notin\mathcal{Z}_{\boldsymbol{x}}^i\right\}\right\}\right\}\\ \geq&\mathbb{P}^R\left\{N\cdot\frac{1}{N}\sum_{i=1}^{N}\underline{\epsilon}_i(s_i^*)\leq\sum_{i=1}^{N}\mathbb{P}\left\{\boldsymbol{x}\in\mathcal{H}:\boldsymbol{z}^*\notin\mathcal{Z}_{\boldsymbol{x}}^i\right\}\right\}.\end{aligned}$$

The equality is achieved if for any $i\neq j$, $\boldsymbol{z}^*\neq\mathcal{Z}_{\boldsymbol{x}}^i\Leftrightarrow\boldsymbol{z}^*\neq\mathcal{Z}_{\boldsymbol{x}}^j$ and $\underline{\epsilon}_i(s_i^*)=\underline{\epsilon}_j(s_j^*)$. The right-hand side of (5.40) can be then lower-bounded by

$$\begin{aligned}&\mathbb{P}^R\left\{N\cdot\frac{1}{N}\sum_{i=1}^{N}\underline{\epsilon}_i(s_i^*)\leq\sum_{i=1}^{N}\mathbb{P}\left\{\boldsymbol{x}\in\mathcal{H}:\boldsymbol{z}^*\notin\mathcal{Z}_{\boldsymbol{x}}^i\right\}\right.\\ &\left.\bigcap\sum_{i=1}^{N}\mathbb{P}\left\{\boldsymbol{x}\in\mathcal{H}:\boldsymbol{z}^*\notin\mathcal{Z}_{\boldsymbol{x}}^i\right\}\leq\sum_{i=1}^{N}\bar{\epsilon}_i(s_i^*)\right\}\\ &\geq\mathbb{P}^R\left\{\bigcap_{i=1}^{N}\left\{\underline{\epsilon}_i(s_i^*)\leq\mathbb{P}\left\{\boldsymbol{x}\in\mathcal{H}:\boldsymbol{z}^*\notin\mathcal{Z}_{\boldsymbol{x}}^i\right\}\leq\bar{\epsilon}_i(s_i^*)\right\}\right\}\\ &\geq 1-\sum_{i=1}^{N}\mathbb{P}^R\left\{\bar{\epsilon}_i(s_i^*)<\mathbb{P}\left\{\boldsymbol{x}\in\mathcal{H}:\boldsymbol{z}^*\notin\mathcal{Z}_{\boldsymbol{x}}^i\right\}\right.\\ &\left.\bigcup\mathbb{P}\left\{\boldsymbol{x}\in\mathcal{H}:\boldsymbol{z}^*\notin\mathcal{Z}_{\boldsymbol{x}}^i\right\}<\underline{\epsilon}_i(s_i^*)\right\}.\end{aligned}\tag{5.41}$$

By applying Theorem 2.3.1 to every agent $i\in\{1,\ldots,N\}$, in the sense that it holds only for the the CBF constraints that involve agent i, we have that for any $i\in\{1,\ldots,N\}$

$$\begin{aligned}&\mathbb{P}^R\left\{\boldsymbol{x}\in\mathcal{H}:\underline{\epsilon}_i(s_i^*)\leq\mathbb{P}\left\{\boldsymbol{x}\in\mathcal{H}:\boldsymbol{z}^*\notin\mathcal{Z}_{\boldsymbol{x}}^i\right\}\leq\bar{\epsilon}_i(s_i^*)\right\}\\ &\geq 1-\beta_i\\ \Rightarrow&\sum_{i=1}^{N}\mathbb{P}^R\left\{\bar{\epsilon}_i(s_i^*)<\mathbb{P}\left\{\boldsymbol{x}\in\mathcal{H}:\boldsymbol{z}^*\notin\mathcal{Z}_{\boldsymbol{x}}^i\right\}\right.\end{aligned}$$

$$\bigcup \mathbb{P}\left\{\boldsymbol{x} \in \mathcal{H} : \boldsymbol{z}^* \notin \mathcal{Z}_{\boldsymbol{x}}^i\right\} < \underline{\epsilon}_i(s_i^*)\Big\} < \sum_{i=1}^{N} \beta_i. \tag{5.42}$$

Here s_i^* is the number of $\boldsymbol{x}^{(r)}$'s for which there exists $e \in \mathcal{C}_i$, such that $\sum_{k\in\mathcal{V}_e} h_{ke}(u_k(\boldsymbol{x}^{(r)})) \geq \sum_{k\in\mathcal{V}_e} z_{ke}^*$. For a specific r, this means that agent i recognizes that at least one CBF constraint is violated up to level $\sum_{i\in\mathcal{V}_e} z_{ie}^*$, over this scenario $\boldsymbol{x}^{(r)}$. After solving the scenario program (5.32) and communicating with the neighbouring agents in $\mathcal{G}_e$ in a distributed manner, every individual agent is able to compute $\bar{\epsilon}_i^*(s_i^*), \underline{\epsilon}_i^*(s_i^*)$ by (5.37), (5.38). Since $\frac{\underline{\epsilon}(s^*)}{N} < \underline{\epsilon}(s^*) < \bar{\epsilon}(s^*)$, substituting (5.42) into (5.40) with $i = 1, \ldots, N$ we obtain

$$\mathbb{P}^R\left\{\frac{\underline{\epsilon}(s^*)}{N} \leq \mathbb{P}\left\{\boldsymbol{x} \in \mathcal{H} : \boldsymbol{z}^* \notin \mathcal{Z}_{\boldsymbol{x}}\right\} \leq \bar{\epsilon}(s^*)\right\} \geq 1 - \beta. \tag{5.43}$$

□

Theorem 5.3.1 constitutes a generalization of [128, Theorem 2] to a multi-agent setting. It also extends [166] by determining the lower bound $\frac{\underline{\epsilon}(s^*)}{N}$. Theorem 5.3.1 states that with confidence $1 - \beta$, the system tends to be unsafe by means of the CBFs with probability within the interval $[\frac{\underline{\epsilon}(s^*)}{N}, \bar{\epsilon}(s^*)]$.

5.4 Simulation Results

The distributed safe control input design and safety verification algorithms are numerically validated on a multi-robot positions swapping problem. To facilitate comparison, we adopt a similar setup as in [110].

5.4.1 Multi-Robot Position Swapping

Robots are assigned different initial positions and are required to navigate towards target locations. In a distributed framework, robots are equipped with sensing and communication modules for collision detection and information sharing. A group of ten robots, indexed by $i = 1, \ldots, 10$ are considered, with double integrator dynamics

$$\begin{bmatrix}\dot{\boldsymbol{p}}_i \\ \dot{\boldsymbol{v}}_i\end{bmatrix} = \begin{bmatrix}0 & I_{2\times 2} \\ 0 & 0\end{bmatrix}\begin{bmatrix}\boldsymbol{p}_i \\ \boldsymbol{v}_i\end{bmatrix} + \begin{bmatrix}0 \\ I_{2\times 2}\end{bmatrix}\boldsymbol{a}_i, \tag{5.44}$$

where $\boldsymbol{p}_i \in \mathbb{R}^2$, $\boldsymbol{v}_i \in \mathbb{R}^2$ represent positions and velocities, and $\boldsymbol{a}_i \in \mathbb{R}^2$ is the control input, representing accelerations. The acceleration is limited as $||\boldsymbol{a}_i||_\infty \leq a_i^{\max}$. $a_i^{\max}$ will be cleared in the sequel. Each robot is regarded as a disk centered at $\boldsymbol{p}_i$ with radius $D_i \in \mathbb{R}_+$. The safety certificate $s_{ij}(\boldsymbol{p}, \boldsymbol{v})$ for collision avoidance between robot i and j is defined by

$$s_{ij}(\boldsymbol{p}, \boldsymbol{v}) = ||\Delta \boldsymbol{p}_{ij}||_2^2 - D_{ij}, \tag{5.45}$$

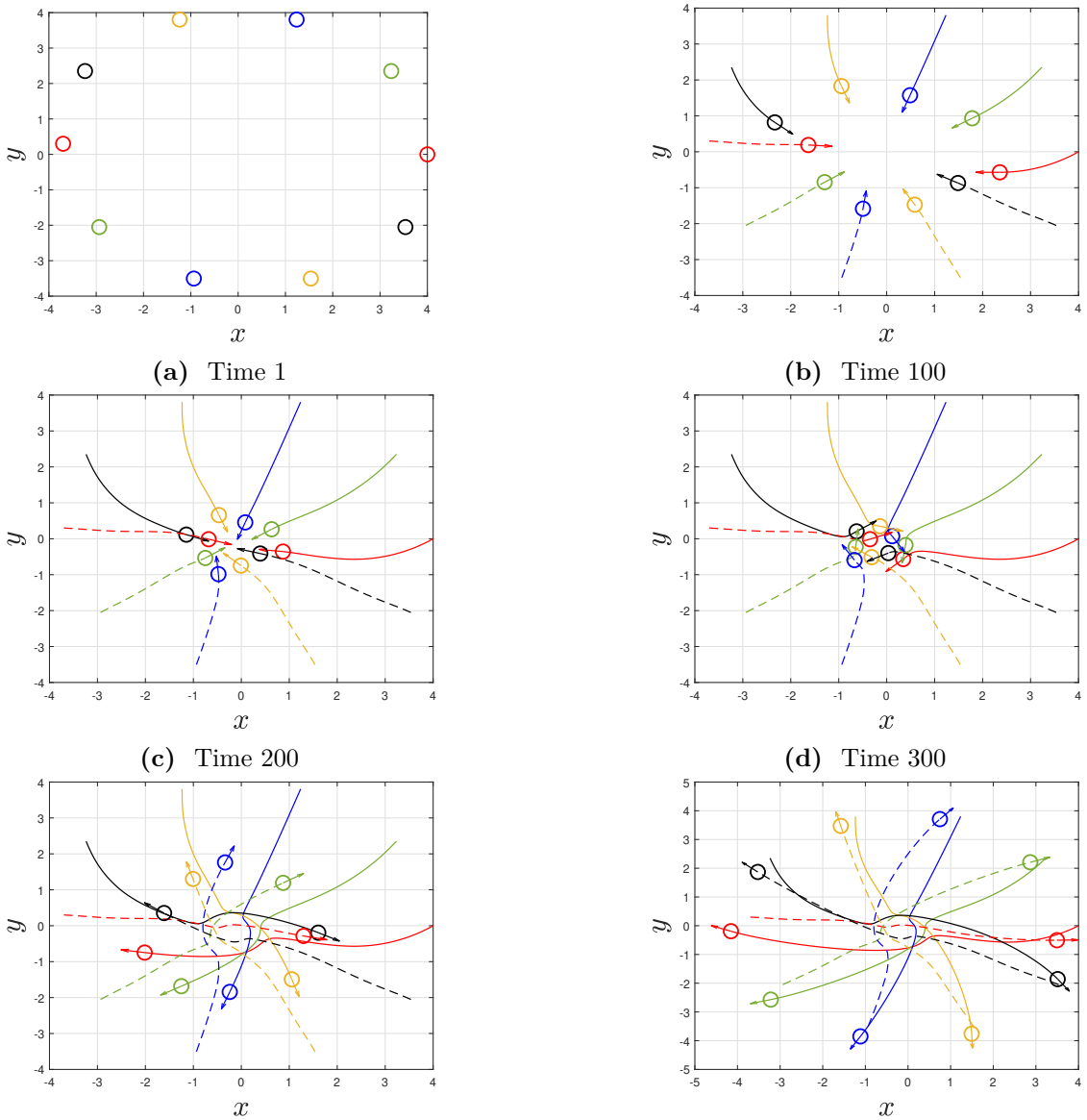
where $\Delta \boldsymbol{p}_{ij} = \boldsymbol{p}_i - \boldsymbol{p}_j$, $D_{ij} = D_i + D_j$. Note here that the system is heterogeneous as different robots have different mobility. Following [110], the control barrier function for invariance certificates is then defined pair-wisely, as

$$\begin{aligned} b_{ij}(\boldsymbol{p}, \boldsymbol{v}) = &\sqrt{2(a_i^{\max} + a_j^{\max})(||\Delta \boldsymbol{p}_{ij}||_2^2 - D_{ij})} \\ &+ \frac{\Delta \boldsymbol{p}_{ij}^\top}{||\Delta \boldsymbol{p}_{ij}||_2^2} \Delta \boldsymbol{v}_{ij}, \end{aligned} \tag{5.46}$$

where $\Delta \boldsymbol{v}_{ij} = \boldsymbol{v}_i - \boldsymbol{v}_j$. The function $b_{ij}(\boldsymbol{p}, \boldsymbol{v})$ is guaranteed to be a CBF since when $b_{ij}(\boldsymbol{p}, \boldsymbol{v}) > 0$, collision can be avoided with maximum braking acceleration $\boldsymbol{a}_i^{\max} + \boldsymbol{a}_j^{\max}$ applied to robots i and j. For $i = 1, \ldots, 5$, $\boldsymbol{a}_i^{\max} = 1$, while for $i = 6, \ldots, 10$, $\boldsymbol{a}_i^{\max} = 10$. Note that although $b_{ij}(\boldsymbol{p}, \boldsymbol{v})$ is guaranteed to be a CBF for safety certificate $s_{ij}(\boldsymbol{p}, \boldsymbol{v})$, the corresponding invariant set $\mathcal{B} = \prod_{\{i,j\} \in \mathcal{E}} \mathcal{B}_{ij}$ is possibly empty. Intuitively, this is since robots cannot utilize maximum braking force to avoid collision with multiple other robots simultaneously. This problem is beyond the scope of this chapter, and we still adopt the CBF as in (5.46).

5.4.2 Distributed Control: Asymptotic Algorithm

The distributed safe control design procedure of Algorithm 1 that exhibits asymptotic convergence and optimality guarantees is implemented for robots to swap positions with the opposite robots while avoiding collision. The resulting simulation results are shown in Figure 5.1.



(a) Time 1 **(b)** Time 100

(c) Time 200 **(d)** Time 300

(e) Time 400 **(f)** Time 500

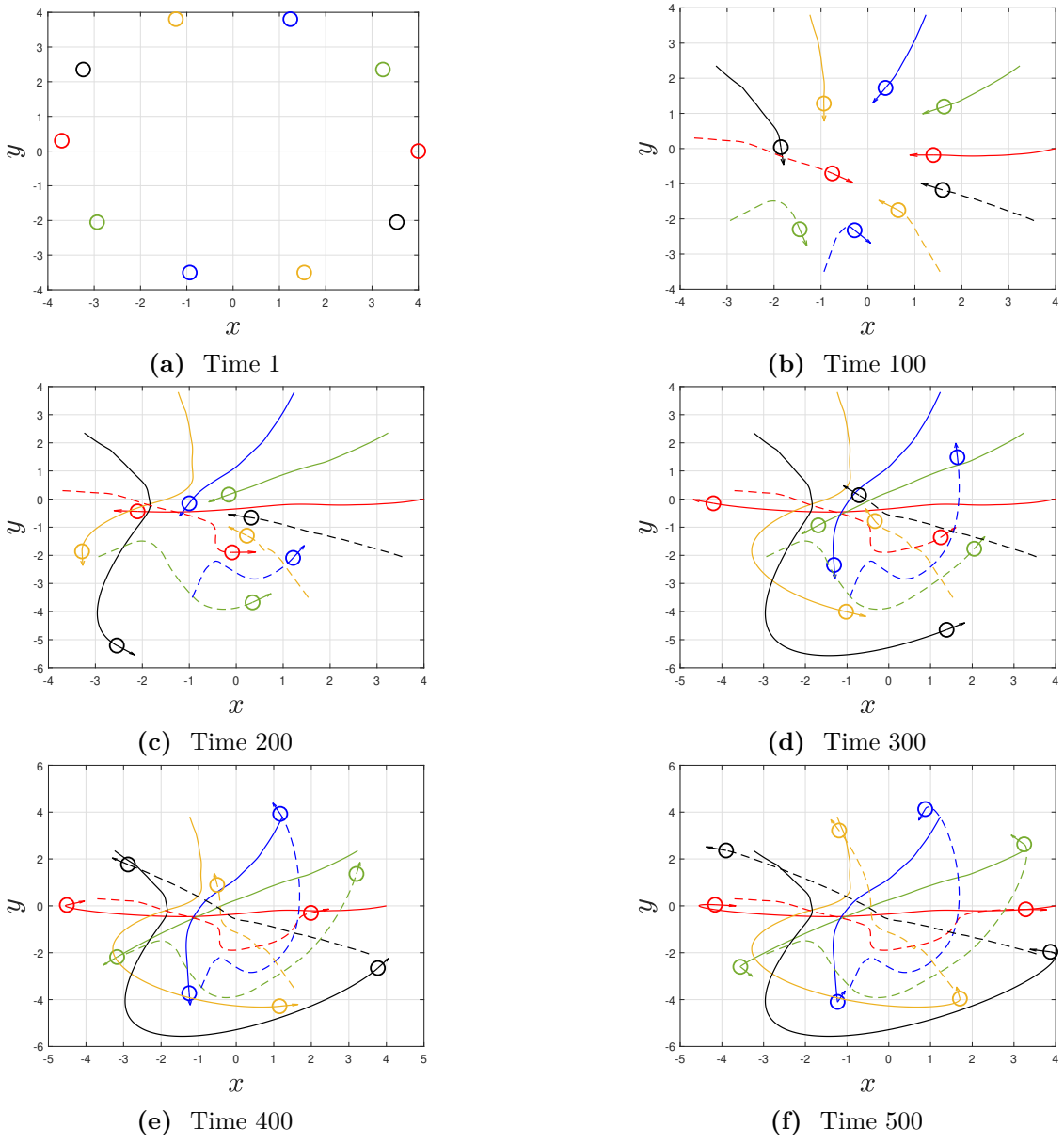
Figure 5.1: Trajectory of ten robots swapping positions according to Algorithm 1. Robots with the same color are swapping positions, and avoiding collision with the others.

5.4.3 Distributed Control: Truncated Algorithm

The truncated Algorithm 2 is then implemented for the same setting, the truncation parameter $\eta = 30$.

The resulting swapping trajectories are shown in Figure 5.2. Define

$$\rho_{\text{sum}}^k = \sum_{i=1}^{N} \sum_{e \in \mathcal{C}_i} \left((\rho_{ie}^k)^2 + M_i \rho_{ie}^k \right). \tag{5.47}$$



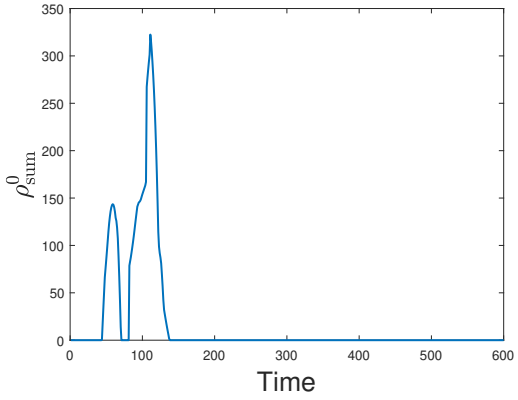
(a) Time 1 **(b)** Time 100

(c) Time 200 **(d)** Time 300

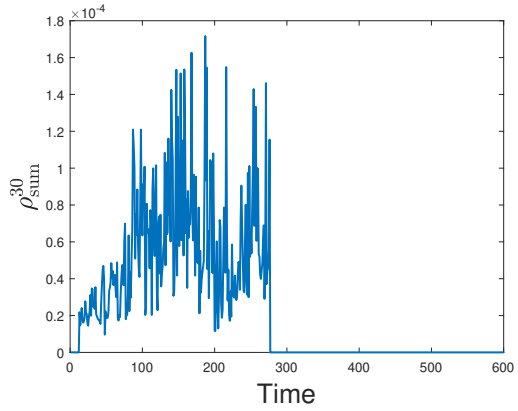
(e) Time 400 **(f)** Time 500

Figure 5.2: Trajectory of ten robots swapping positions while avoiding collision by means of Algorithm 2, with $\eta = 30$.

The evolution of the relaxation parameters $\rho^0_{\text{sum}}(\boldsymbol{x})$ and $\rho^{30}_{\text{sum}}(\boldsymbol{x})$ at each time step along the trajectory is shown in Figures 5.3a and 5.3b. It can be seen that ρ^{30}_{sum} is close to zero at every time step, even ρ^0_{sum} is relatively large at some time steps. This empirically demonstrates the safety guarantees performance of the proposed distributed algorithm. From our experience, η could be much smaller for a practical implementation.



(a) $\rho^0_{\text{sum}}(\boldsymbol{x})$ along the trajectory



(b) $\rho^{30}_{\text{sum}}(\boldsymbol{x})$ along the trajectory

Figure 5.3: Evolution of the relaxation parameters $\rho^0_{\text{sum}}(\boldsymbol{x})$ and $\rho^{30}_{\text{sum}}(\boldsymbol{x})$ evaluated at the state trajectory, across algorithm iterations.

5.4.4 Distributed Safety Verification

Safety verification is performed for a four-robot system, within the working space $\mathcal{X}$, defined as $\{p : ||p|| \leq p^{\max} = 6\} \times \{v : ||v|| \leq v^{\max} = 1\}$. Each robot is using Algorithm 2 to safely move towards the origin. We sample 200 scenarios via Algorithm 3. Theorem 5.3.1 yields then that with confidence at least 0.9, $\mathbb{P}\{\boldsymbol{x} \in \mathcal{H} : \boldsymbol{z}^* = 0 \notin \mathcal{Z}_{\boldsymbol{x}}\} \in [0, 0.146]$. We repeat this procedure 300 times, each time using 300 scenarios, and construct the empirical cumulative distribution function of $\mathbb{P}\{\boldsymbol{x} \in \mathcal{H} : 0 \notin \mathcal{Z}_{\boldsymbol{x}}\}$. This is shown in Figure 5.4; it can be observed that the empirical probability that $\mathbb{P}\{\boldsymbol{x} \in \mathcal{H} : 0 \notin \mathcal{Z}_{\boldsymbol{x}}\} \in [0, 0.146] \approx 1$, thus satisfying the theoretical confidence lower bound of 0.9.

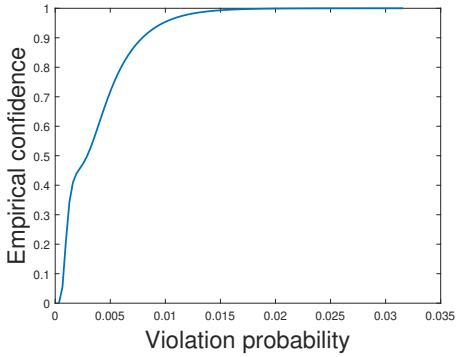


Figure 5.4: Cumulative distribution function for safety violation.

5.5 Conclusion

In this chapter we presented distributed safe control design and safety verification algorithms for multi-agent systems. The proposed control algorithms introduce auxiliary and relaxation variables to allow feasibility across iterations. We guaranteed convergence to an optimal solution and established a sublinear convergence rate under certain conditions. We also addressed the problem of distributed safety verification for given control inputs. A scenario-based verification program was formulated and can be solved locally by each agent. The scenarios are sampled independently by a sequential algorithm. The distributed scenario program characterizes the probability of being unsafe, with both lower and upper bounds being determined. Simulation on a multi-robot swapping position problem demonstrated the efficacy of our result. Current work concentrates in accounting for communication delays and model uncertainty in real systems.

6

Safe and Stable Filter Design Using a Relaxed Compatibitlity Control Barrier – Lyapunov Condition

In Chapters 3 - 5, we focus on the problem of synthesizing safe controllers. This chapter extensively proposes a control design approach for ensuring both safety and local stability. Results are presented to show how this approach guarantees local stability and safety, as well as eliminates undesired equilibrium points.

6.1 Introduction

Safety and stability are essential properties for modern automation applications [167]. In control theory, one popular methodology to verify and enforce these properties is to use *certificate functions*. For a nonlinear autonomous system, stability can be verified by constructing a Lyapunov function [168]. When a control input is considered, the concept of a Control Lyapunov Function (CLF) has been introduced[169]. With a Control Lyapunov Function in hand, a stabilizing controller can be designed using Sontag's universal formula [170]. On the other hand, safety verification can be achieved by defining a barrier function over the state-space [65]. The zero-level set of a barrier function is forward invariant and

discriminates the safe and unsafe regions. Analogous to CLFs, Control Barrier Functions (CBFs) have been introduced to guide controller design for safety [69, 138].

Table 6.1: Comparisons on closed-loop behaviours of our filter and the other CLF-CBF based filters in the literature. By $\mathcal{O}_l$ we mean the l-sublevel set of a CLF, $\mathcal{O}_{\eta \leq l}$ denotes the η-sublevel set of the CLF, where $0 \leq n \leq l$. It holds that $\mathcal{O}_{\eta \leq l} \subseteq \mathcal{O}_l$. Note that in [131] compatibility between CBF and CLF is strict (in a sense specified formally in the sequel), however, linear independence of the associated constraints is imposed as an assumption.

Paper	Local Stability	Interior Equilibria	ROA	Stabilizing Controller	Compatibility	CBF/CLF Design
This Chapter	✓	Do not Exist	$\mathcal{O}_l$	Not Required	Relaxed	✓
[69]	–	Exist	–	Not Required	Not Required	–
[77]	✓	Exist	–	Not Required	Not Required	–
[117]	✓	Do not Exist	$\mathcal{O}_l$	Required	Not Required	–
[118]	✓	Do not Exist	$\mathcal{O}_{\eta \leq l}$	Required	Strict	–
[119]	✓	Do not Exist	$\mathcal{O}_{\eta \leq l}$	Not Required	Strict	–
[131]	✓	Do not Exist	$\mathcal{O}_l$	Not Required	Strict	✓

In this chapter, we propose a new filter based on the *relaxed compatibility* condition for a CBF and CLF. We demonstrate that our method obtains the desired closed-loop behaviour, i.e. local asymptotic stability and elimination of interior equilibrium points. Our method does not require a stabilizing nominal controller to enhance stability of the designed optimal controller. Moreover, the optimal controller is guaranteed to be locally Lipschitz continuous inside the invariant set without an *a priori* assumption on the linear independence of the CBF- and CLF-induced linear constraints, which is assumed in [131]. Additionally, we provide a design method for a relaxed compatible pair of CLF and CBF for a polynomial dynamical system using sum-of-squares programming. The inherent nonlinearities in the program are addressed through an iterative algorithm. We validate the efficacy of our method by comparative studies against other stat-of-the-art methods, and our method shows superior filter performance.

A classification of the most closely related results in the literature and a qualitative comparison of their closed-loop behaviour with respect to the proposed approach is provided in Table 6.1. Under mild conditions, i.e. with relaxed compatibility and without pre-stabilizing controller, our method guarantees local stability, eliminates all the interior equilibrium points, and results in a large ROA $\mathcal{O}_l$, where l denotes

the l-superlevel set of a CLF. Numerical comparisons of filter performance in terms of optimality and CBF/CLF design with other works are illustrated by numerical studies in Section 6.4.

Our filter is proposed and analyzed in Section 6.2. The CBF/CLF design program is presented in Section 6.3. In Section 6.4, we conduct comparative numerical studies, while Section 6.5 provides some concluding remarks.

6.2 Safety and Stability Filter

6.2.1 Problem statement and motivating example

Consider a nonlinear control-affine system

$$\dot{x} = f(x) + g(x)u, \tag{6.1}$$

where $x(t) \in \mathcal{X} \subset \mathbb{R}^n$ denotes the state of the system and $u(t) \in \mathbb{R}^m$ denotes the input. Here $f(x) : \mathcal{X} \to \mathbb{R}^n$, $g(x) : \mathcal{X} \to \mathbb{R}^m$ are locally Lipschitz continuous functions in $\mathcal{X}$, and $f(0) = 0$. Our goal is to design a locally Lispchitz continuous state feedback controller $u(\cdot) : \mathcal{X} \to \mathbb{R}^m$ such that the solution of the closed-loop system $\dot{x} = f(x) + g(x)u(x)$ that starts from $x(0) = x_0$, stays within a *safe set* $\mathcal{S}$ for every t that belongs to the time-domain over which solutions are defined. If such a controller $u(\cdot)$ exists, we say that the system is *safe*.

Definition 6.2.1. A locally positive definite and differentiable function $V(\cdot) : \mathcal{X} \to \mathbb{R}$ is called a *Control Lyapunov Function (CLF)* if there exist a positive definite function $\gamma(\cdot) : \mathbb{R}^n \to \mathbb{R}$, and a locally Lispchitz continuous state-feedback controller with $u(x) : \mathcal{X} \to \mathbb{R}^m$, such that

$$\mathcal{L}_f V(x) + \mathcal{L}_g V(x)u(x) + \gamma(x) \leq 0, \forall x \in \mathcal{X}, \tag{6.2}$$

where $\mathcal{L}_f V(x) = \frac{\partial V(x)}{\partial x} f(x)$, $\mathcal{L}_g V(x) = \frac{\partial V(x)}{\partial x} g(x)$.

The definition below is known as a *compatibility* condition [118–120], and captures the cases under which the CLF and CBF conditions are both feasible.

Definition 6.2.2 (Compatibility[120, Definition 1])**.** Consider system (6.1), CLF $V(x)$, CBF $b(x)$, extended class-$\mathcal{K}$ function $\alpha(\cdot) : \mathbb{R} \to \mathbb{R}$, and positive definite function $\gamma(\cdot) : \mathbb{R}^n \to \mathbb{R}$. $V(x)$ and $b(x)$ are said to be *compatible* if there exists a locally Lipschitz continuous controller $u(x) : \mathcal{X} \to \mathbb{R}^m$, such that for any $x \in \mathcal{X}$:

$$\begin{aligned} &\mathcal{L}_f b(x) + \mathcal{L}_g b(x) u(x) + \alpha(b(x)) \geq 0, \\ &\mathcal{L}_f V(x) + \mathcal{L}_g V(x) u(x) + \gamma(x) \leq 0. \end{aligned} \tag{6.3}$$

In the next section we propose a relaxed version of this condition, which is at the core of our analysis.

Consider system (6.1), and a locally Lipschitz continuous controller $\pi : \mathcal{X} \to \mathbb{R}^m$ satisfying $\pi(0) = 0$. Our goal is to design a locally Lipschitz continuous controller $u^*(x)$ that locally stabilizes system (6.1) around the origin, and guarantees forward invariance of the set $\mathcal{B} = \{x \in \mathbb{R}^n : b(x) \geq 0\}$. Moreover, the optimal controller $u^*(x)$ should be close to $\pi(x)$ as much as possible. For any $x \in \mathcal{X}$, the proposed safety and stability filter takes the form

$$\begin{aligned} (u^*(x), s^*(x)) = \underset{u \in \mathbb{R}^m, s \in \mathbb{R}}{\arg\min} \ & \frac{1}{2}||u - \pi(x)||^2 + \frac{p}{2}(s-1)^2 \\ \text{subject to} \\ F_1(u, s) = \mathcal{L}_f b(x) + \mathcal{L}_g b(x) u + s(\alpha(b(x))) \geq 0, \\ F_2(u) = \mathcal{L}_f V(x) + \mathcal{L}_g V(x) u + \beta(b(x))\gamma(x) \leq 0, \end{aligned} \tag{6.4}$$

where $\alpha(\cdot) : \mathbb{R}^n \to \mathbb{R}$ and $\beta(\cdot) : \mathbb{R}^n \to \mathbb{R}$ are extended class-$\mathcal{K}$ functions, $\gamma(\cdot) : \mathbb{R}^n \to \mathbb{R}$ is a positive definite function. In addition, $\beta(x) \leq 1$, for any $x \in \mathcal{X}$. One typical choice for $\beta(\cdot)$ is $\tanh(\cdot)$. The filter is constructed at the current state x given the value of $\pi(x)$. In this way the optimal solution u^* and s^* are both functions of x. In particular, we have that $u^*(x) : \mathcal{X} \to \mathbb{R}^m$, and $s^*(x) \in \mathcal{X} \to \mathbb{R}$ since we only consider one CBF constraint. Unlike the filters proposed in [69], [117] which relax the CLF constraint with a slack variable, our proposed filter (6.4) uses an adaptive variable s to ensure feasibility while promoting safety, and a term $\beta(b(x))\gamma(x)$ to ensure feasibility while promoting stability.

The intuition for $\beta(b(x))$ is that, for $x \in \{x \in \mathbb{R}^n : b(x) > 0\}$, we have that $\beta(b(x)) > 0$ (since β is an extended class-$\mathcal{K}$ function), and also $\dot{V} < 0$ holds. Moreover, for larger $b(x)$, $\beta(b(x))$ accelerates the convergence speed by amplifying $\beta(b(x))\gamma(x)$. For any x such that $\beta(b(x))$ takes the largest admissible value 1, our adapted CLF constraint is equivalent to the original one. This happens when $b(x)$ (the argument of $\beta(\cdot)$) is large enough, i.e., the system is safe to a certain level. $\beta(x)$ can be thought of as an activation function to trigger the CLF constraint, based on the CBF constraint.

Prior to analyzing our proposed filter, to better understand the benefits of introducing s and $\beta(\cdot)$ in terms of feasibility, safety, and stability, we provide the following motivating example.

Example 6.2.1. Consider a second-order linear system

$$\dot{x} = x + u, \tag{6.5}$$

where $x = [x_1, x_2]^\top$, $u = [u_1, u_2]^\top$. A bounded obstacle is defined by the set $\{x \in \mathbb{R}^2 : ||x - (0, 4)||^2 \leq 2\}$. This benchmark case has been considered in [115, 117, 119]. In these papers, a candidate CLF is $V(x) = x^\top x$, and a CBF

$$b(x) = ||x - (0, 4)||^2 - 4,$$

while $\gamma(x) = V(x)$, $\alpha(b(x)) = b(x)$. The original CBF constraint is given by $\dot{b}(x) + b(x) \geq 0$, which is $\begin{bmatrix} 2x_1 \\ 2x_2 - 8 \end{bmatrix}^\top \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} + 3x_1^2 + 3x_2^2 - 16x_2 + 12 \geq 0$. The original CLF constraint is given by $\dot{V}(x) + V(x) \leq 0$, which is $\begin{bmatrix} 2x_1 \\ 2x_2 \end{bmatrix}^\top \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} + 3x_1^2 + 3x_2^2 \geq 0$. Given a state x, both constraints are affine in u, thus define two half planes on $\mathbb{R}^2$. When

$$\begin{bmatrix} 2x_1 \\ 2x_2 - 8 \end{bmatrix} // \begin{bmatrix} 2x_1 \\ 2x_2 \end{bmatrix},$$

the intersection of the two half planes is potentially empty. Clearly, this happens only when $x_1 = 0$. Under this, the CBF and CLF constraints are then given by $2x_2u_2 + 3x_2^2 \leq 0, (2x_2 - 8)u_2 + 3x_2^2 - 16x_2 + 12 \geq 0$. It is easy to verify that, when $x_2 \leq 4$, there always exist u_2 that satisfies these constraints. Considering $x_2 > 4$,

we have$-\frac{3x_2^2-16x_2+12}{2x_2-8} \leq u_2 \leq -\frac{3x_2}{2}$. However, the simultaneous satisfaction of these inequalities is impossible if

$$-\frac{3x_2^2-16x_2+12}{2x_2-8} > -\frac{3x_2}{2} \iff x_2 > 3.$$

Therefore, for any $x_1 = 0, x_2 > 4$, there does *not* exist $u(x)$ that satisfies the CLF and CBF constraints, and the CLF $V(x)$ and CBF $b(x)$ are not compatible according to Definition 6.2.2.

Now consider the constraint $F_1(u,s) \geq 0$ in (6.4) given by

$$\begin{aligned} 2x_1u_1 + (2x_2-8)u_2 + (x_1^2+x_2^2-8x_2+12)s \\ + 2x_1^2 + 2x_2^2 - 8x_2 \geq 0, \end{aligned}$$

and the CLF constraint $F_2(u) \leq 0$ in (6.4), given by

$$2x_1u_1 + 2x_2u_2 + 2x_1^2 + 2x_2^2 + \beta(b(x))(x_1^2+x_2^2) \leq 0.$$

Given a state x, both constraints are affine in u and s, thus define two half planes on $\mathbb{R}^3$. The intersection of the two half planes is potentially empty if

$$\begin{bmatrix} 2x_1 \\ 2x_2-8 \\ x_1^2+x_2^2-8x_2+12 \end{bmatrix} // \begin{bmatrix} 2x_1 \\ 2x_2 \\ 0 \end{bmatrix}.$$

This happens when $x_1 = 0$ and $x_1^2+x_2^2-8x_2+12 = 0$, which imply $x_1 = 0, x_2 = 2$ or $x_1 = 0, x_2 = 6$. These two points both lie on the boundary of $\{x \in \mathbb{R}^2 : b(x) = 0\}$. Now it is clear that the additional decision variable, s, enlarges the affine constraint space at any $x \notin \{x \in \mathbb{R}^2 : b(x) = 0\}$, thus improving feasibility. On the set $\{x \in \mathbb{R}^2 : b(x) = 0\}$, we clearly have that constraint $F_1(u,s) \geq 0$ is the same as the nominal CBF constraint, i.e. the one without s.

Considering $x_1 = 0$, $x_2 = 2$, the adapted CBF constraint $F_1(u,s) \geq 0$ implies $u_2 \leq -2$, the adapted CLF constraint is

$$4u_2 + 8 + 4\underbrace{\beta(b(x))}_{b(x)=0} \leq 0 \iff u_2 \leq -2.$$

At this point, any $u_1 \in \mathbb{R}$ and $u_2 \leq -2$ satisfy the two constraints. Considering $x_1 = 0, x_2 = 6$, the adapted CBF constraint implies $u_2 \geq -6$, the adapted CLF constraint is

$$12u_2 + 72 + 4\underbrace{\beta(b(x))}_{b(x)=0} \leq 0 \Longleftrightarrow u_2 \leq -6.$$

Similarly, at this point, any $u_1 \in \mathbb{R}$, and $u_2 = -6$ satisfy the CBF and CLF constraints simultaneously. Recall that at $x_1 = 0$, $x_2 = 6 > 4$, there exists no $u(x)$ that satisfies the CBF and CLF constraints (as required by the standard compatibility definition). This illustrates the potential benefit of introducing $\beta(b(x))$, as it promotes feasibility on the set $\{x \in \mathbb{R}^2 : b(x) = 0\}$.

Example 6.2.1 shows that our filter (6.4) introduces s and $\beta(b(x))$ to adapt the original CBF/CLF constraints to guarantee feasibility. What remains to answer is whether the adapted CBF/CLF constraints are feasible on $\{x \in \mathbb{R}^n : b(x) = 0\}$. This motivates the concept of *relaxed compatibility*.

Definition 6.2.3 (Relaxed Compatibility)**.** Consider system (6.1), CLF $V(x)$, and CBF $b(x)$. $V(x)$ and $b(x)$ are said to satisfy the relaxed compatibility condition if there exists a locally Lipschitz continuous $u(x) : \mathcal{X} \to \mathbb{R}^m$, such that for any $x \in \{x \in \mathbb{R}^n : b(x) = 0\}$:

$$\begin{aligned} \mathcal{L}_f b(x) + \mathcal{L}_g b(x) u(x) &\geq 0, \\ \mathcal{L}_f V(x) + \mathcal{L}_g V(x) u(x) &\leq 0. \end{aligned} \tag{6.6}$$

Compatibility is naturally sufficient but not necessary for relaxed compatibility. There are two main differences between them:

1. Conditions for relaxed compatibility only require the existence of a controller $u(x)$ that satisfies the CBF constraint and the relaxed CLF constraint ($\dot{V}(x) \geq 0$) for any $x \in \partial\mathcal{B}$, whereas the existence of controller is required for any $x \in \mathbb{R}^n$ for compatibility.

2. The CLF constraint in (6.6) is non-strict and does not include the convergence rate term $\gamma(x)$ on $\partial\mathcal{B}$, whilst is strict in the sense of $\dot{V} < 0, \forall x \neq 0$ in (6.3).

Relaxed compatibility refers to compatibility of *invariance* and *stability* that only relies on a CBF (control invariant set), a CLF, and the system dynamics. The motivation for (2) comes from a geometric observation in [121]. Let $\mathcal{B}^c$ denote the set complement of $\mathcal{B}$, and suppose $\mathcal{B}^c$ is bounded. It is then shown that there is no locally Lipschitz continuous controller $u(x)$ that guarantees that both $\dot{V} < 0$ and $\dot{b} \geq 0$ on some points on the boundary of the invariant set[121]. As a result, compatibility can not hold on these points. Relaxed compatibility therefore proposes a milder condition that $\dot{V} \leq 0$ and $\dot{b} \geq 0$ on these points.

Theorem 6.2.1. *Consider system* (6.1). *Suppose there exist a CBF* $b(x)$ *and a CBF* $V(x)$ *that satisfy the relaxed compatibility condition as per Definition 6.2.3. Then the optimization problem* (6.4) *is feasible for any* $x \in \mathcal{X}$.

Proof. We will show that both constraints in (6.4) are feasible. To this end, by Definition 6.2.1, for any $x \in \mathcal{X}$, there exists u', such that $\mathcal{L}_f V(x) + \mathcal{L}_g V(x)u' + \gamma(x) \leq 0$. Given that $\gamma(\cdot)$ is non-negative, and $\beta(x) \leq 1$, we have that for any $x \in \mathcal{X}$, $\beta(b(x))\gamma(x) \leq \gamma(x)$. This indicates that for any $x \in \mathcal{X}$,

$$\begin{aligned} F_2(u') &= \mathcal{L}_f V(x) + \mathcal{L}_g V(x)u' + \beta(b(x))\gamma(x) \\ &\leq \mathcal{L}_f V(x) + \mathcal{L}_g V(x)u' + \gamma(x) \leq 0, \end{aligned} \tag{6.7}$$

where the second inequality is due to the property of the selected u'.

Consider the same u' for the CBF constraint, and let

$$s' = \begin{cases} -\frac{\mathcal{L}_f b(x) + \mathcal{L}_g b(x)u'}{\alpha(b(x))} + 1, & \text{if } b(x) > 0, \\ -\frac{\mathcal{L}_f b(x) + \mathcal{L}_g b(x)u'}{\alpha(b(x))} - 1, & \text{if } b(x) < 0, \\ 0, & \text{if } b(x) = 0, \end{cases} \tag{6.8}$$

and notice that the division with $\alpha(b(x))$ is admissible for the values of $b(x)$ in these cases. We have that for any $x \in \{x \in \mathbb{R}^n : b(x) \neq 0\}$,

$$F_1(u', s') = \mathcal{L}_f b(x) + \mathcal{L}_g b(x)u' + s'(\alpha(b(x))) \geq 0, \tag{6.9}$$

where the inequality follows by substituting in place of s' the expressions in the first two branches of that correspond to $x \in \mathcal{X}$ such that $b(x) \neq 0$. Additionally,

for any $x \in \mathcal{X}$ such that $b(x) = 0$, by the relaxed compatibility condition (6.6), we directly have that $F_1(u', s') \geq 0$. As such, in any case feasibility of (6.4) is ensured, thus concluding the proof. □

6.2.2 Equilibrium Characterization

Theorem 6.2.1 shows that under the relaxed compatibility condition of Definition 6.2.3, (6.4) is feasible for any $x \in \mathcal{X}$. We now provide a closed-form expression for the optimal solution of (6.4).

Theorem 6.2.2. *Consider system* (6.1)*, and a safe set* $\mathcal{S} \subset \mathbb{R}^n$*. Suppose there exist a CBF* $b(x)$ *and a CLF* $V(x)$ *that satisfy the relaxed compatibility condition. The optimal controller* $u^*(x)$ *obtained by the safety and stability filter* (6.4) *is given by*

$$u^* = \begin{cases} \pi, \forall x \in \Omega^{\overline{cbf}}_{\overline{clf}} \cup \Omega^{\overline{cbf}}_{clf,2} \cup \Omega^{cbf}_{clf,5} \cup \Omega^{cbf}_{\overline{clf},2}, & \text{(6.10a)} \\ \pi - \dfrac{F_b}{F_b'} \mathcal{L}_g b^\top, \forall x \in \Omega^{cbf}_{\overline{clf},1} \cup \Omega^{cbf}_{clf,4}, & \text{(6.10b)} \\ \pi - \dfrac{F_V}{\mathcal{L}_g V \mathcal{L}_g V^\top} \mathcal{L}_g V^\top, \forall x \in \Omega^{\overline{cbf}}_{clf,1}, & \text{(6.10c)} \\ \pi + \begin{bmatrix} \mathcal{L}_g b^\top \\ -\mathcal{L}_g V^\top \end{bmatrix}^\top A(x)^{-1} \begin{bmatrix} -F_b(x) \\ F_V(x) \end{bmatrix}, \\ \quad \forall x \in \Omega^{cbf}_{clf,1}, & \text{(6.10d)} \\ \pi + \dfrac{F_V}{\mathcal{L}_g V \mathcal{L}_g V^\top} \mathcal{L}_g V^\top, \\ \quad \forall x \in \Omega^{cbf}_{clf,2} \cup \Omega^{cbf}_{clf,3}, & \text{(6.10e)} \end{cases}$$

where

$$A(x) = \begin{bmatrix} \mathcal{L}_g b \mathcal{L}_g b^\top + \frac{\alpha(b(x))^2}{p} & -\mathcal{L}_g b \mathcal{L}_g V^\top \\ \mathcal{L}_g V \mathcal{L}_g b^\top & -\mathcal{L}_g V \mathcal{L}_g V^\top \end{bmatrix},$$

$F_b(x) = \mathcal{L}_g b(x)\pi(x) + \mathcal{L}_f b(x) + \alpha(b(x))$, $F_b'(x) = \mathcal{L}_g b(x) \mathcal{L}_g b(x)^\top + \alpha(b(x))^2/p$, $F_V(x) = \mathcal{L}_f V(x) + \mathcal{L}_g V(x)\pi(x) + \beta(b(x))\gamma(x)$. *The critical regions that appear in* (6.10) *are defined by*

$$\Omega^{\overline{cbf}}_{\overline{clf}} = \{x \in \mathbb{R}^n : F_b > 0, F_V < 0\}, \quad \text{(6.11a)}$$

$$\Omega^{cbf}_{\overline{clf},1} = \{x \in \mathbb{R}^n : F_V F_b' - F_b \mathcal{L}_g V \mathcal{L}_g b^\top < 0, \\ F_b \leq 0, F_b' \neq 0\}, \quad \text{(6.11b)}$$

$$\Omega_{\overline{clf},2}^{\overline{cbf}} = \{x \in \mathbb{R}^n : F_V < 0, b(x) = 0, \mathcal{L}_g b = 0\}, \tag{6.11c}$$

$$\Omega_{clf,1}^{\overline{cbf}} = \{x \in \mathbb{R}^n : F_b \mathcal{L}_g V \mathcal{L}_g V^\top - F_V \mathcal{L}_g b \mathcal{L}_g V^\top > 0,$$

$$F_V \geq 0, \mathcal{L}_g V \neq 0\}, \tag{6.11d}$$

$$\Omega_{clf,2}^{\overline{cbf}} = \{x \in \mathbb{R}^n : F_b > 0, \mathcal{L}_g V = 0\}, \tag{6.11e}$$

$$\Omega_{clf,1}^{cbf} = \{x \in \mathbb{R}^n : \mathcal{L}_g V \neq 0, F_b \mathcal{L}_g V \mathcal{L}_g b^\top + F_b' F_v \geq 0$$

$$F_b \mathcal{L}_g V \mathcal{L}_g V^\top + F_V \mathcal{L}_g b \mathcal{L}_g V^\top \geq 0, b(x) \neq 0\}, \tag{6.11f}$$

$$\Omega_{clf,2}^{cbf} = \{x \in \mathbb{R}^n : \mathcal{L}_g V \neq 0, F_V \leq 0, F_b' = 0, F_b = 0\}, \tag{6.11g}$$

$$\Omega_{clf,3}^{cbf} = \{x \in \mathbb{R}^n : \mathcal{L}_g V \neq 0, b(x) = 0,$$

$$F_V \leq 0, \mathcal{L}_g b // \mathcal{L}_g V, F_V \mathcal{L}_g V = -F_b \mathcal{L}_g b\}. \tag{6.11h}$$

$$\Omega_{clf,4}^{cbf} = \{x \in \mathbb{R}^n : \mathcal{L}_g V = 0, F_b \leq 0, F_b' \neq 0, F_V = 0\}. \tag{6.11i}$$

$$\Omega_{clf,5}^{cbf} = \{x \in \mathbb{R}^n : \mathcal{L}_g V = 0, F_b' = 0, F_b = 0, F_V = 0\}. \tag{6.11j}$$

For the ease of notation, we have dropped the dependency of x for $F_b(x)$, $F_V(x)$, $\mathcal{L}_b(x)$, $\mathcal{L}_V(x)$, $F_b'(x)$.

Proof. Dualizing the control barrier function constraint $-\mathcal{L}_f b - \mathcal{L}_g b u - s\alpha(b(x)) \leq 0$ with a multiplier $\lambda_1 \geq 0$, and the control Lyapunov function constraint $\mathcal{L}_f V + \mathcal{L}_g V u + \beta(b(x))\gamma(x) \leq 0$ with a multiplier $\lambda_2 \geq 0$, we obtain the corresponding Lagrangian

$$\begin{aligned} L(u, s, \lambda_1, \lambda_2) = \frac{1}{2}||u - \pi(x)||^2 + \frac{p}{2}(s-1)^2 \\ - \lambda_1(\mathcal{L}_f b + \mathcal{L}_g b u + s\alpha(b(x))) \\ + \lambda_2(\mathcal{L}_f V + \mathcal{L}_g V u + \beta(b(x))\gamma(x)). \end{aligned}$$

The KKT conditions are given by

$$\left.\frac{\partial L}{\partial u}\right|_{u=u^*} = u^* - \pi(x) - \lambda_1 \mathcal{L}_g b^\top + \lambda_2 \mathcal{L}_g V^\top = 0, \tag{6.12a}$$

$$\left.\frac{\partial L}{\partial s}\right|_{s=s^*} = p(s^* - 1) - \lambda_1 \alpha(b(x)) = 0, \tag{6.12b}$$

$$\lambda_1 \left(\mathcal{L}_f b + \mathcal{L}_g b u^* + s^* \alpha(b(x))\right) = 0, \tag{6.12c}$$

$$\lambda_2 \left(\mathcal{L}_f V + \mathcal{L}_g V u^* + \beta(b(x))\gamma(x)\right) = 0. \tag{6.12d}$$

We highlight here that, u^*, s^*, λ_1 and λ_2 are all functions of x. The dependency on x is dropped to simplify notation. By regarding x as a parameter to the quadratic programming problem (6.4), the analytical solution can be evaluated by considering which constraints are active or inactive [69, 117, 171]. We consider the following cases.

Case 1: Both the CBF and the CLF constraint are inactive.

In this case, we have

$$F_1(u^*, s^*) > 0 \text{ and } F_2(u^*) < 0. \tag{6.13}$$

Then we have $\lambda_1 = 0$, and $\lambda_2 = 0$ from the complementary slackness conditions (6.12c) (6.12d). By (6.12a) and (6.12b), we obtain that $u^* = \pi(x)$ and $s^* = 1$, respectively. This case happens for $x \in \Omega_{\overline{clf}}^{\overline{cbf}} \subset \mathbb{R}^n$, where

$$\Omega_{\overline{clf}}^{\overline{cbf}} = \{x \in \mathbb{R}^n : F_1(u^*, s^*) > 0, F_2(u^*) < 0\}. \tag{6.14}$$

Substituting $u^* = \pi(x), s^* = 1$ into $F_1(u^*, s^*), F_2(u^*)$, we thus obtain (6.11a).

Case 2: The CLF constraint is inactive and the CBF constraint is active.

In this case, we have $F_1(u^*, s^*) = 0, F_2(u^*) < 0$. Then we directly have $\lambda_2 = 0$ from (6.12d). From (6.12a) we obtain

$$u^* = \pi(x) + \lambda_1 \mathcal{L}_g b^\top, \tag{6.15}$$

while from (6.12b) we obtain

$$s^* = 1 + \frac{\lambda_1 \alpha(b(x))}{p}. \tag{6.16}$$

We then consider the following two sub-cases.

1) $F_b' = \mathcal{L}_g b \mathcal{L}_g b^\top + \alpha(b(x))^2/p \neq 0$. Substituting (6.15) and (6.16) into $F_1(u^*, s^*) = 0$, we get

$$\lambda_1 = -\frac{\mathcal{L}_g b \pi(x) + \mathcal{L}_f b + \alpha(b(x))}{\mathcal{L}_g b \mathcal{L}_g b^\top + \alpha(b(x))^2/p} = -\frac{F_b}{F_b'}. \tag{6.17}$$

Substituting then (6.17) into (6.15) and (6.16) we obtain

$$u^* = \pi(x) - \frac{F_b}{F_b'}\mathcal{L}_g b^\top, s^* = 1 - \frac{F_b \alpha(b(x))}{pF_b'}. \tag{6.18}$$

The critical region $\Omega^{cbf}_{clf,1} \subset \mathbb{R}^n$ is then defined by

$$\Omega^{cbf}_{clf,1} = \{x \in \mathbb{R}^n : F_2(u^*) < 0, \lambda_1 \geq 0, F_b' \neq 0\}. \tag{6.19}$$

Substituting (6.18) into (6.19), and since F_b' is positive in this case, we obtain (6.11b).

2) $F_b' = 0$. This implies $\mathcal{L}_g b = 0$ and $b(x) = 0$. From (6.15) and (6.16), we have that

$$u^* = \pi(x) \text{ and } s^* = 1. \tag{6.20}$$

For this case, λ_1 can be any arbitrary non-negative scalar. This is since the decision variables u and s do not appear in the CBF constraint. The dual function depends solely on λ_2. The critical region $\Omega^{cbf}_{clf,2}$ is thus

$$\Omega^{cbf}_{clf,2} = \{x \in \mathbb{R}^n : F_V < 0, b(x) = 0, \mathcal{L}_g b = 0\}, \tag{6.21}$$

which establishes (6.11c).

Case 3: The CLF constraint is active and the CBF constraint is inactive.

For this case, we have $F_1(u^*, s^*) > 0, F_2(u^*) = 0$. By (6.12c) we have that $\lambda_1 = 0$. Substituting this into (6.12b), we obtain $s^* = 1$, while substituting $\lambda_1 = 0$ into (6.12a) yields

$$u^* = \pi(x) - \lambda_2 \mathcal{L}_g V^\top. \tag{6.22}$$

Two further sub-cases are considered depending on the different values of $\mathcal{L}_g V$.

1) $\mathcal{L}_g V \neq 0$. Substituting (6.22) into $F_2(u^*) = 0$ we have

$$\lambda_2 = \frac{\mathcal{L}_f V + \mathcal{L}_g V \pi(x) + \beta(b(x))\gamma(x)}{\mathcal{L}_g V \mathcal{L}_g V^\top} = \frac{F_V}{\mathcal{L}_g V \mathcal{L}_g V^\top}. \tag{6.23}$$

By (6.23), (6.22) results in

$$u^* = \pi(x) - \frac{F_V}{\mathcal{L}_g V \mathcal{L}_g V^\top}\mathcal{L}_g V^\top. \tag{6.24}$$

The critical region $\Omega_{clf,1}^{\overline{cbf}}$ is then defined by

$$\Omega_{clf,1}^{\overline{cbf}} = \{x \in \mathbb{R}^n : F_1(u^*, s^*) > 0, \lambda_2 \geq 0, \mathcal{L}_g V \neq 0\}. \tag{6.25}$$

Substituting (6.24) and $s^* = 1$ into (6.25), and since $\mathcal{L}_g V$ is positive in this case, we obtain (6.11d).

2) $\mathcal{L}_g V = 0$. By (6.22) we have $u^* = \pi(x)$. For this case, λ_2 can be any non-negative scalar. The critical region is then defined by

$$\Omega_{clf,2}^{\overline{cbf}} = \{x \in \mathbb{R}^n : F_1(u^*, s^*) > 0, \mathcal{L}_g V = 0\}. \tag{6.26}$$

Substituting $u^* = \pi$ and $s^* = 1$ into (6.26), we obtain (6.11e).

Case 4: Both the CBF and CLF constraint are active. In this case, we have that

$$F_1(u^*, s^*) = 0 \text{ and } F_2(u^*) = 0. \tag{6.27}$$

By (6.12a) and (6.12b), respectively, we obtain that

$$u^* = \pi(x) + \lambda_1 \mathcal{L}_g b^\top - \lambda_2 \mathcal{L}_g V^\top, \tag{6.28}$$

$$s^* = 1 + \frac{\lambda_1 \alpha(b(x))}{p}. \tag{6.29}$$

Substituting (6.28), and (6.29) into (6.27), we obtain a linear equation

$$\begin{bmatrix} \mathcal{L}_g b \mathcal{L}_g b^\top + \frac{\alpha(b(x))^2}{p} & -\mathcal{L}_g b \mathcal{L}_g V^\top \\ \mathcal{L}_g V \mathcal{L}_g b^\top & -\mathcal{L}_g V \mathcal{L}_g V^\top \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix} = \begin{bmatrix} -F_b(x) \\ F_V(x) \end{bmatrix} \tag{6.30}$$

Denote the matrix that pre-multiplies $[\lambda_1, \lambda_2]^\top$ by $A(x)$, and notice that its determinant is given by

$$\begin{aligned} \det(A(x)) = &-\mathcal{L}_g V \mathcal{L}_g V^\top \mathcal{L}_g b \mathcal{L}_g b^\top \\ &- \mathcal{L}_g V \mathcal{L}_g V^\top \alpha(b(x))^2/p + \mathcal{L}_g b \mathcal{L}_g V^\top \mathcal{L}_g V \mathcal{L}_g b^\top. \end{aligned}$$

It can be observed that $\det(A(x)) = 0$ if and only if (i) $\mathcal{L}_g V = 0$; or, (ii) $F_b'(x) = \mathcal{L}_g b(x) \mathcal{L}_g b(x)^\top + \alpha(b(x))^2/p = 0$, which in turn implies that $\mathcal{L}_g b = 0$ and $b = 0$; or, (iii) $b(x) = 0$ and $\mathcal{L}_g V // \mathcal{L}_g b$.

We then consider the following five sub-cases, that capture the situation where $\det(A(x)) \neq 0$, and all possible (distinct) cases for which $\det(A(x)) = 0$, and for each case characterize the resulting optimal solution of (6.4).

1) $\mathcal{L}_g V \neq 0$ and $b(x) \neq 0$. For this case we have $\det(A(x)) \neq 0$. We then have that $A(x)$ is invertible with $A(x)^{-1}$ given by

$$A(x)^{-1} = \frac{\begin{bmatrix} -\mathcal{L}_g V \mathcal{L}_g V^\top & \mathcal{L}_g b \mathcal{L}_g V^\top \\ -\mathcal{L}_g V \mathcal{L}_g b^\top & \mathcal{L}_g b \mathcal{L}_g b^\top + \frac{\alpha(b(x))^2}{p} \end{bmatrix}}{\det(A(x))}. \tag{6.31}$$

We could thus solve the linear system of equations in (6.30) and obtain

$$\begin{bmatrix} \lambda_1 \\ \lambda_2 \end{bmatrix} = A(x)^{-1} \begin{bmatrix} -F_b(x) \\ F_V(x) \end{bmatrix}. \tag{6.32}$$

Substituting (6.32)-(6.31) into (6.28) and (6.29), we obtain the expressions for u^* and s^* given in (6.10d). The critical region $\Omega^{cbf}_{clf,1}$ is defined by

$$\Omega^{cbf}_{clf,1} = \{x \in \mathbb{R}^n : \lambda_1 \geq 0, \lambda_2 \geq 0, \mathcal{L}_g V(x) \neq 0, b \neq 0\}. \tag{6.33}$$

Under the expressions for λ_1 and λ_2 in (6.32), $A(x)^{-1}$ in (6.31), and recalling that $F'_b := \mathcal{L}_g b \mathcal{L}_g b^\top + \frac{\alpha(b(x))^2}{p}$, we obtain (6.11f).

2) $\mathcal{L}_g V \neq 0$ and $F'_b = 0$. $F'_b = 0$ implies that $\mathcal{L}_g b = 0$ and $b = 0$. Therefore, the linear system of equations in (6.30) reduces to

$$0 = -F_b \text{ and } -\mathcal{L}_g V \mathcal{L}_g V^\top \lambda_2 = F_V.$$

We then have $\lambda_2 = -\frac{F_V}{\mathcal{L}_g V \mathcal{L}_g V^\top}$, and $F_b = \mathcal{L}_f b = 0$, while λ_1 takes any arbitrary non-negative value. Substituting these identities into (6.28) and (6.29), we obtain

$$u^* = \pi + \frac{F_V}{\mathcal{L}_g V \mathcal{L}_g V^\top} \mathcal{L}_g V^\top \text{ and } s^* = 1. \tag{6.34}$$

The critical region $\Omega^{cbf}_{clf,2}$ is then given by

$$\Omega^{cbf}_{clf,2} := \{x \in \mathbb{R}^n : \lambda_2 \geq 0, \mathcal{L}_g V \neq 0, F'_b = 0, F_b = 0\}. \tag{6.35}$$

Substituting in the latter $\lambda_2 = -\frac{F_V}{\mathcal{L}_g V \mathcal{L}_g V^\top}$ yields (6.11g).

3) $\mathcal{L}_g V \neq 0, b(x) = 0$, and $\mathcal{L}_g V // \mathcal{L}_g b$. For this case, matrix $A(x)$ has two linearly dependent rows. Therefore, the linear equation is feasible if and only if

$$F_V \mathcal{L}_g V = -F_b \mathcal{L}_g b.$$

Under this condition, the linear equation is feasible and admits infinitely many solutions. Fixing $\lambda_1 = 0$, by (6.30) we have that $\lambda_2 = -\frac{F_V}{\mathcal{L}_g V \mathcal{L}_g V^\top}$. Substituting these identities into (6.28) and (6.29), we have

$$u^* = \pi + \frac{F_V}{\mathcal{L}_g V \mathcal{L}_g V^\top} \mathcal{L}_g V^\top \text{ and } s^* = 1. \tag{6.36}$$

The critical region $\Omega^{cbf}_{clf,3}$ is given by

$$\begin{aligned} \Omega^{cbf}_{clf,3} := \{&x \in \mathbb{R}^n : \mathcal{L}_g V \neq 0, b(x) = 0, \\ &\lambda_2 \geq 0, \mathcal{L}_g b // \mathcal{L}_g V, F_V \mathcal{L}_g V = -F_b \mathcal{L}_g b\}. \end{aligned} \tag{6.37}$$

Substituting $\lambda_2 = -\frac{F_V}{\mathcal{L}_g V \mathcal{L}_g V^\top}$ into (6.37), we obtain (6.11h).

4) $\mathcal{L}_g V = 0$ and $F_b' \neq 0$. For this case, the linear equation (6.32) results in

$$F_b' \lambda_1 = -F_b, \quad 0 = F_V.$$

We thus have that $\lambda_1 = -\frac{F_b}{F_b'}$, while λ_2 takes any arbitrary non-negative value. Substituting these into (6.28) and (6.29), we have

$$u^* = \pi - \frac{F_b}{F_b'} \mathcal{L}_g b^\top, \text{ and } s^* = 1 - \frac{F_b \alpha(b(x))}{p F_b'}. \tag{6.38}$$

The critical region $\Omega^{cbf}_{clf,4}$ is given by

$$\Omega^{cbf}_{clf,4} := \{x \in \mathbb{R}^n : \lambda_1 \geq 0, \mathcal{L}_g V = 0, F_b' \neq 0, F_V = 0\}. \tag{6.39}$$

Substituting $\lambda_1 = -\frac{F_b}{F_b'}$ into (6.39), we obtain (6.11i).

5) $\mathcal{L}_g V = 0$ and $F_b' = 0$. For this case, both λ_1 and λ_2 take arbitrary non-negative values. Besides, as a consequence of (6.30), we have $F_b = 0, F_V = 0$. By (6.28) and (6.29), we then have

$$u^* = \pi(x), s^* = 1. \tag{6.40}$$

The critical region $\Omega^{cbf}_{clf,5}$ is given by

$$\Omega^{cbf}_{clf,5} := \{x \in \mathbb{R}^n : \mathcal{L}_g V = 0, F'_b = 0, F_b = 0, F_V = 0\},$$

which is (6.11j), thus concluding the proof. □

In the next theorem, we analyze the equilibrium points of our the closed-loop system $\dot{x} = f(x) + g(x)u^*(x)$, where $u^*(x)$ is obtained by solving (6.4).

Theorem 6.2.3. *Consider system* (6.1)*. Suppose there exist a CBF* $b(x)$ *and a CLF* $V(x)$ *that satisfy the relaxed compatibility condition. Let* $u^*(x)$ *be the optimal solution of* (6.4)*. Then, the set of equilibrium points of system* $\dot{x} = f(x)+g(x)u^*(x)$ *in* $\mathcal{B}$ *are given by*

$$\mathcal{E} = \mathcal{E}^{clf}_{cbf,1} \cup \mathcal{E}^{clf}_{cbf,2} \cup \mathcal{E}^{clf}_{cbf,3} \cup \{0\} \tag{6.41}$$

where

$$\begin{aligned}
\mathcal{E}^{clf}_{cbf,1} &= \{x \in \{\Omega^{clf}_{cbf,2} \cup \Omega^{clf}_{cbf,3}\} \cap \partial\mathcal{B} : f(x) + g(x)\pi(x) \\
&\quad + g(x)\frac{F_V}{\mathcal{L}_g V \mathcal{L}_g V^\top}\mathcal{L}_g V^\top = 0\}, && \text{(6.42a)} \\
\mathcal{E}^{clf}_{cbf,2} &= \{x \in \{\Omega^{clf}_{cbf,4} \cap \partial\mathcal{B}\} : f(x) + g(x)\pi(x) \\
&\quad - g(x)\frac{F_b}{\mathcal{L}_g b \mathcal{L}_g b^\top}\mathcal{L}_g b^\top = 0\}. && \text{(6.42b)} \\
\mathcal{E}^{clf}_{cbf,3} &= \{x \in \{\Omega^{clf}_{cbf,5} \cap \partial\mathcal{B}\} : f(x) + g(x)\pi(x) = 0\}, && \text{(6.42c)}
\end{aligned}$$

Proof. We first show that $x = 0$ is included in the set of equilibrium points. To this end, notice that $f(0) + g(0)\pi(0) = 0$, as the optimal solution of (6.4) at the origin is given by $u^*(0) = \pi(0)$, and π is defined such that $\pi(0) = 0$. Therefore, the origin is an equilibrium point.

Notice first that for any $x \neq 0$ that satisfies $b(x) > 0$, the CLF constraint in (6.4) implies $\dot{V}(x) = \frac{\partial V(x)}{\partial x}(f(x)+g(x)u^*(x)) < 0$. Therefore, we have $f(x)+g(x)u^*(x) \neq 0$. This in turn implies that there are no equilibrium points in $\text{Int}(\mathcal{B})$, and all equilibrium points have to belong to $\partial\mathcal{B}$.

We next investigate which constraints can be active for points on $\partial\mathcal{B}$. To this end, suppose that either the CBF constraint $\dot{b}(x) \geq 0$ or the CLF constraint $\dot{V}(x) \leq 0$ is

inactive. We have either $\dot{b}(x) = \frac{\partial b(x)}{x}(f(x)+g(x)u^*(x)) > 0$ or $\dot{V}(x) = \frac{\partial V(x)}{\partial x}(f(x)+g(x)u^*(x)) < 0$. It thus follows that $f(x)+g(x)u^*(x) \neq 0$ for either case. Therefore, we conclude that the CBF constraint and the CLF constraint have to be both active, and $b(x) = 0$ at all equilibrium points. This corresponds to the second, third, forth, and fifth sub-cases of Case 4 in the proof of Theorem 6.2.2. We show next that all these cases are encompassed in the sets of (6.41).

In particular:

(i) For the second and third sub-case of Case 4 in the proof of Theorem 6.2.2, $x \in \Omega^{clf}_{cbf,2} \cup \Omega^{clf}_{cbf,3}$. By Theorem 6.2.2, we thus have that $u^* = \pi + \frac{F_V}{\mathcal{L}_g V \mathcal{L}_g V^\top}\mathcal{L}_g V^\top$. Under this choice of u^* we obtain (6.42a) for any x such that $f(x)+g(x)u^*(x) = 0$.

(ii) For the fourth sub-case of Case 4 in the proof of Theorem 6.2.2, we have that $x \in \Omega^{clf}_{cbf,4}$. We then have that $\mathcal{L}_g V = 0, F'_b = 0$. Recalling that $F'_b = \mathcal{L}_g b \mathcal{L}_g b^\top + \alpha(b)^2/p = \mathcal{L}_g b \mathcal{L}_g b^\top$, following (6.10b), the optimal controller $u^*(x)$ is given by $u^*(x) = \pi - \frac{F_b}{\mathcal{L}_g b \mathcal{L}_g b^\top}\mathcal{L}_g b^\top$. Under this choice of u^* we obtain (6.42b) for any x such that $f(x) + g(x)u^*(x) = 0$.

(iii) For the fifth sub-case of Case 4 in the proof of Theorem 6.2.2, we have that $x \in \Omega^{clf}_{cbf,5}$. By Theorem 6.2.2, we then have $u^* = \pi(x)$. Under this choice of u^* we obtain (6.42c) for any x such that $f(x) + g(x)u^*(x) = 0$. □

6.2.3 Continuity Analysis

We now analyze $u^*(x)$, the optimal solution of (6.4), in terms of its continuity properties. We first provide two definitions that will be employed in the sequel and are widely encountered in the literature [77, 117].

Definition 6.2.4 (*Small Control Property*)**.** A CLF $V(x)$ is said to have the *Small Control Property* (SCP) if there exists a compact set Ω and a locally Lipschitz continuous control law $u_c(x) : \mathcal{X} \to \mathbb{R}^m$, such that $0 \in \Omega$, $\text{Int}(\Omega) \neq \emptyset$, $\lim_{x\to 0} u_c(x) \to 0$ and

$$\mathcal{L}_f V(x) + \mathcal{L}_g V(x) u_c(x) + \gamma(x) \leq 0, \forall x \in \Omega.$$

Definition 6.2.5 (*Strict Complementary Slackness*)**.** The CLF constraint in (6.4) is said to have *Strict Complementary Slackness* (SCS) if

$$\mathcal{L}_f V(x) + \gamma(x) < 0,\ \forall x \in \mathcal{X}\backslash\{0\} \text{ such that } \mathcal{L}_g V(x) = 0.$$

SCS can be easily satisfied for CLFs. To see this, consider CLF $V(x)$, feedback controller $u(x)$, and function $\gamma(x)$ that satisfy (6.2). Suppose there exists a set $\tilde{\mathcal{X}} \subseteq \{\mathcal{X} \cap \{x : \mathcal{L}_g V(x) = 0\}\backslash\{0\}\}$, such that $\mathcal{L}_f V(x) + \gamma(x) = 0, \forall x \in \tilde{\mathcal{X}}$. Define a new function $\tilde{\gamma}(x) = \gamma(x)/2$. Given that $0 < \tilde{\gamma}(x) < \gamma(x), \forall x \in \tilde{\mathcal{X}}$, we deduce that $\mathcal{L}_f V(x) + \tilde{\gamma}(x) < 0, \forall x \in \tilde{\mathcal{X}}$. Hence, the CLF constraint in (6.4) can satisfy SCS by replacing $\gamma(x)$ with $\tilde{\gamma}(x)$.

Theorem 6.2.4. *Consider system* (6.1)*. Suppose there exist CBF* $b(x)$ *and CLF* $V(x)$ *that satisfy the relaxed compatibility condition, and* $0 \in \text{Int}(\mathcal{B})$*. Suppose* $f(x)$*,* $g(x)$ *and* $\pi(x)$ *are locally Lipschitz continuous on every compact subset of* $\text{Int}(\mathcal{B})\backslash\{0\}$*. Let* $u^*(x)$ *be the optimal controller obtained by* (6.4)*. If SCS holds, then* $u^*(x)$ *is locally Lipschitz continuous* $\forall x \in \text{Int}(\mathcal{B})\backslash\{0\}$*. Additionally, if SCP holds, then* $u^*(x)$ *is also locally Lipschitz continuous at 0.*

Proof. The proof follows a similar process as that for [77, Theorem 1]. To simplify notation, we let $\Omega_{\overline{clf}}^{cbf} = \Omega_{\overline{clf},1}^{cbf} \cup \Omega_{\overline{clf},2}^{cbf}$, $\Omega_{clf}^{\overline{cbf}} = \Omega_{clf,1}^{\overline{cbf}} \cup \Omega_{clf,2}^{\overline{cbf}}$, $\Omega_{clf}^{cbf} = \Omega_{clf,1}^{cbf} \cup \Omega_{clf,2}^{cbf} \cup \Omega_{clf,3}^{cbf} \cup \Omega_{clf,4}^{cbf} \cup \Omega_{clf,5}^{cbf}$. Therefore, the coefficient matrix $M(x)$ that multiplies $[u^\top, s]^\top$ in the active constraints of (6.4) is given by

$$M(x) = \begin{cases} 0, & \forall x \in \Omega_{\overline{clf}}^{\overline{cbf}}, \quad (6.43a) \\ \begin{bmatrix} \mathcal{L}_g V & 0 \end{bmatrix}, & \forall x \in \Omega_{clf}^{\overline{cbf}}, \quad (6.43b) \\ \begin{bmatrix} \mathcal{L}_g b & \alpha(b(x)) \end{bmatrix}, & \forall x \in \Omega_{\overline{clf}}^{cbf}, \quad (6.43c) \\ \begin{bmatrix} \mathcal{L}_g b & \alpha(b(x)) \\ \mathcal{L}_g V & 0 \end{bmatrix}, & \forall x \in \Omega_{clf}^{cbf}. \quad (6.43d) \end{cases}$$

Select a convex and compact set $\mathcal{D} \subset \text{Int}(\mathcal{B})\backslash\{0\}$. The solution of (6.4) in $\mathcal{D}$ is unique since the cost function is strictly convex and the constraints are linear. Given that $\mathcal{D}$ is compact, $\mathcal{L}_g b$, $\mathcal{L}_g V$, and $\alpha(b(x))$ are all locally Lipschitz continuous

(since $f(x)$, $g(x)$, $V(x)$, $b(x)$, and $\alpha(\cdot)$ are all locally Lipschitz continuous), there exists $\zeta > 0$ such that $||M(x)|| < \zeta$ for every $x \in \mathcal{D}$. By affinity of the constraints this then implies that $u^*(x)$ is locally Lipschitz continuous for $x \in \text{cl}(\Omega_{clf}^{cbf}) \cap \mathcal{D}$ since $u^*(x) = \pi(x)$. If $\mathcal{L}_g V = 0$ for some $x \in \Omega_{clf}^{\overline{cbf}} \cap \mathcal{D}$, the CLF constraint is necessarily inactive since $\mathcal{L}_f V + \beta(b(x))\gamma(x) \leq \mathcal{L}_f V + \gamma(x) < 0$ according to (6.43a), which contradicts the definition of $\Omega_{clf}^{\overline{cbf}}$ in (6.11). Therefore, for every $x \in \Omega_{clf}^{\overline{cbf}} \cap \mathcal{D}$, we have $\mathcal{L}_g V \neq 0$. Similarly, for every $x \in \text{cl}(\Omega_{\overline{clf}}^{cbf}) \cap \mathcal{D}$, we have $\mathcal{L}_g V \neq 0, \alpha(b(x)) \neq 0$. In summary, $M(x)$ is of full row rank in all cases. The conditions of [77] are satisfied, and the optimal solution $u^*(x)$ is locally Lipschitz continuous in every compact set $\mathcal{D}$.

We then prove that if SCP holds, $u^*(x)$ is also locally Lipschitz continuous at the origin. Given that $0 \in \text{Int}(\mathcal{B})$ and $f(0) + g(0)\pi(0) = 0$, we have $F_1(u_1(x), 1)|_{x=0} = F_b(0) > 0$ for any locally Lipschitz continuous control $u_1(x)$ that satisfies $u_1(0) = 0$, which implies $0 \in \text{cl}(\Omega_{\overline{clf}}^{\overline{cbf}}) \cup \text{cl}(\Omega_{clf}^{\overline{cbf}})$. Therefore there exists a compact set $\mathcal{F}_b(u_1(x)) \subset \text{cl}(\Omega_{\overline{clf}}^{\overline{cbf}}) \cup \text{cl}(\Omega_{clf}^{\overline{cbf}})$ that satisfies $0 \in \text{Int}(\mathcal{F}_b(u_1(x)))$ and $F_1(u_1, 1) > 0 \, \forall x \in \mathcal{F}_b(u_1(x))$. From the SCP of $V(x)$, there exists a locally Lipschitz continuous control $u_2(x)$ that satisfies $u_2(0) = 0$, and a compact set $\mathcal{F}_V(u_2(x))$ that satisfies $0 \in \text{Int}(\mathcal{F}_V(u_2(x)))$, such that $F_2(u_2(x) \leq 0 \; \forall x \in \mathcal{F}_V(u_2(x))$. Define $\mathcal{F}(u_2(x)) = \mathcal{F}_V(u_2(x)) \cap \mathcal{F}_b(u_2(x))$. $\mathcal{F}(u_2(x))$ has a non-empty interior since $0 \in \text{Int}(\mathcal{F}_b(u_1(x)))$ and $0 \in \text{Int}(\mathcal{F}_V(u_2(x)))$. Then, for any $x \in \mathcal{F}(u_2(x))$,

$$
\begin{aligned}
& F_2(u_2(x)) = \mathcal{L}_f V + \mathcal{L}_g V u_2(x) + \beta(b(x))\gamma(x) \leq 0 \\
& \iff F_V + \mathcal{L}_g V(u_2(x) - \pi(x)) \leq 0 \\
& \iff |F_V| \leq ||\mathcal{L}_g V|| \cdot ||u_2(x) - \pi(x)|| \\
& \iff \frac{|F_V|}{||\mathcal{L}_g V||} \leq ||u_2(x) - \pi(x)|| \qquad (6.44)
\end{aligned}
$$

Given that $\lim_{x \to 0} u_2(x) \to 0$, $\lim_{x \to 0} \pi(x) \to 0$, we have $\lim_{x \to 0} ||u_2(x) - \pi(x)|| \to 0$. Hence, $\lim_{x \to 0} \frac{|F_V|}{||\mathcal{L}_g V||} \to 0$. Moreover, we have that the CBF constraint is inactive on $\mathcal{F}(u_2(x))$ since $\mathcal{F}(u_2(x)) \subseteq \mathcal{F}_b(u_1(x))$. From (6.10a) and (6.10c), we have $\lim_{x \to 0} u^*(x) \to 0$, for any $x \in \text{cl}(\Omega_{\overline{clf}}^{\overline{cbf}}) \cup \text{cl}(\Omega_{clf}^{\overline{cbf}})$. Therefore, $u^*(x)$ is locally

Lipschitz continuous on a compact set $\mathcal{F}(u^*) \subset \text{cl}(\Omega_{\overline{clf}}^{\overline{cbf}}) \cup \text{cl}(\Omega_{clf}^{\overline{cbf}})$ that contains the origin. Hence, we conclude the proof. □

Lipschitz continuity of $u^*(x)$, $f(x)$ and $g(x)$ guarantees uniqueness of solution for system (6.1). Using Theorem 6.2.4, we can then analyze the safety and stability performance of system (6.1) using the optimal controller $u^*(x)$.

Lemma 6.2.1. *Consider system* (6.1). *Suppose there exist a CBF* $b(x)$ *and a CLF* $V(x)$ *that satisfy the relaxed compatibility condition. Let* $u^*(x)$ *be the optimal controller obtained by solving* (6.4). *Then, for the closed-loop system* $\dot{x} = f(x) + g(x)u^*(x)$ *we have that:*

1. *The set* $\mathcal{B}$ *is forward invariant;*
2. *The CLF* $V(x)$ *is decreasing for every* $x \in \text{Int}(\mathcal{B})\backslash\{0\}$.

Proof. Forward invariance is guaranteed since for any $x \in \partial\mathcal{B}$, $\dot{b}(x) = \mathcal{L}_f b + \mathcal{L}_g b u^*(x) \geq 0$, and $f(x) + g(x)u^*(x)$ is locally Lipschitz continuous. For every $x \in \text{Int}(\mathcal{B})\backslash\{0\}$, the CLF $V(x)$ is decreasing since $\dot{V} = \mathcal{L}_f V + \mathcal{L}_g V u^*(x) < F_2(u^*(x)) \leq 0$. □

Theorem 6.2.5. *Consider system* (6.1). *Suppose there exist a CBF* $b(x)$ *and a CLF* $V(x)$ *that satisfy the relaxed compatibility condition following Definition 6.2.2. Let* $u^*(x)$ *be the optimal controller obtained from* (6.4). *If* $0 \in \text{Int}(\mathcal{B})$, *then the closed-loop system* (6.1) $\dot{x} = f(x) + g(x)u^*(x)$ *is locally asymptotically stable at the origin. Moreover, for any* $l > 0$, *the set*

$$\mathcal{O}_l := \{x \in \mathbb{R}^n : V(x) \leq l\} \tag{6.45}$$

is a region of attraction if $\mathcal{O}_l \cap \partial\mathcal{B} = \emptyset$.

Proof. For every $x \in \text{Int}(\mathcal{B})\backslash\{0\}$, we have $\dot{V}(x) = \mathcal{L}_f V + \mathcal{L}_g V u^*(x) \leq -\beta(b(x))\gamma(x) < 0$. From Lasalle's invariance principle, any closed invariant set $\mathcal{R}$ that contains the origin is a region of attraction for the closed loop system. For any $l > 0$ such that $\mathcal{O}_l \cap \partial\mathcal{B} = \emptyset$, we have $\dot{V}(x) \leq 0$. Therefore, $\mathcal{O}_l$ is an invariant set for system $\dot{x} = f(x) + g(x)u^*$, and therefore a region of attraction. □

6.3 CLF & CBF Design and Verification

The proposed filter (6.4) relies on pre-designed CLF $V(x)$ and CBF $b(x)$ that satisfy the relaxed compatibility condition. In this section, we show how to design these functions for a polynomial dynamical system and semi-algebraic sets $\mathcal{S}$, $\mathcal{X}$. After designing the functions offline, a controller that guarantees safety and local stability can be synthesized by solving (6.4) in an online manner.

Assumption 6.3.1. *Consider system* (6.1). *$f(x)$ and $g(x)$ are polynomial functions. Assume that $\mathcal{S}$ and $\mathcal{X}$ are semi-algebraic sets, defined by $\mathcal{S} = \{x \in \mathbb{R}^n : s(x) \geq 0\}$, and $\mathcal{X} := \{x \in \mathbb{R}^n : w(x) \geq 0\}$, respectively.*

To begin with the design method, we first restrict our decision variables such that $\sigma_1(x)$, $\sigma_2(x)$, $\sigma_3(x)$, $\lambda_1(x)$, $\lambda_2(x)$, $b(x)$, $V(x)$, $u_b(x)$, $u_V(x)$. Additionally, we predefine two sum-of-squares polynomials $\varepsilon_1(x)$ and $\varepsilon_2(x)$ with no constant terms. Clearly, we have

$$\varepsilon_i(x) > 0, \forall x \neq 0 \text{ and } \varepsilon_i(0) = 0, \forall i \in \{1, 2\}. \tag{6.46}$$

We impose the following requirements on our design:

(i) Polynomial constraints: Using the previous identities, we require all decision variables to be polynomials. Namely,

$$\begin{aligned} &\sigma_1(x), \sigma_2(x), \sigma_3(x) \in \Sigma[x], \\ &\lambda_1(x), \lambda_2(x), b(x), V(x), u_b(x), u_V(x) \in \mathbb{R}[x], b(0) > 0. \end{aligned} \tag{6.47}$$

Specifically, $\{\sigma_i(x)\}_{i=1}^3$ are further restricted to be sum-of-squares polynomials, as they will be used to deduce set containment conditions using the first part of Lemma 2.2.5. The value of CBF $b(x)$ is positive at $x = 0$, which implies that $0 \in \text{Int}(\mathcal{B})$.

(ii) $\mathcal{B} \subseteq \mathcal{S}$: To enforce this, we introduce $\sigma_1(x) \in \Sigma[x]$ and use the first part of Lemma 2.2.5. We then obtain

$$-b(x) + \sigma_1(x)s(x) \in \Sigma[x]. \tag{6.48}$$

(iii) $\frac{\partial b(x)}{\partial x}(f(x)+g(x)u_b(x)) \geq 0, \forall x \in \partial\mathcal{B}$: A sufficient condition to ensure this is by means of constraint (6.49). This involves introducing $\lambda_1(x) \in \mathbb{R}[x]$, and $u_b(x) \in \mathbb{R}[x]$, and considering the second part of Lemma 2.2.5.

$$\frac{\partial b(x)}{\partial x}(f(x)+g(x)u_b(x)) + \lambda_1(x)b(x) \in \Sigma[x] \tag{6.49}$$

(iv) $\frac{\partial V(x)}{\partial x}(f(x)+g(x)u_V(x)) \leq -\varepsilon_1(x) < 0, \forall x \in \mathcal{X}\backslash\{0\}$: A sufficient condition to ensure this is by means of constraint (6.50). This involves introducing $u_V(x) \in \mathbb{R}[x]$, $\sigma_2(x) \in \Sigma[x]$, and considering (6.46) and the first part of Lemma 2.2.5.

$$-\frac{\partial V(x)}{\partial x}(f(x)+g(x)u_V(x)) - \sigma_2(x)w(x) - \varepsilon_1(x) \in \Sigma[x]. \tag{6.50}$$

(v) $V(x) \geq \varepsilon_2(x) > 0, \forall x \in \mathcal{X}\backslash\{0\}$: A sufficient condition to ensure this is by means of constraint (6.51). This involves introducing $\sigma_3(x) \in \Sigma[x]$, and considering (6.46) and the first part of Lemma 2.2.5.

$$V(x) - \varepsilon_2(x) - \sigma_3(x)w(x) \in \Sigma[x]. \tag{6.51}$$

(vi) $\frac{\partial V(x)}{\partial x}(f(x)+g(x)u_b(x)) \leq 0, \forall x \in \partial\mathcal{B}$: A sufficient condition to ensure this is by means of constraint (6.52). This involves introducing $u_b(x) \in \mathbb{R}[x]$, and $\lambda_2(x) \in \mathbb{R}[x]$, and considering the second part of Lemma 2.2.5.

$$-\frac{\partial V(x)}{\partial x}(f(x)+g(x)u_b(x)) - \lambda_2(x)b(x) \in \Sigma[x]. \tag{6.52}$$

By incorporating all constraints in (i)-(vi) into one optimization (feasibility) problem, we propose the following program to design a CBF $b(x)$ and CLF $V(x)$ that satisfy the relaxed compatibility condition (6.6).

$$\begin{aligned} \text{find } & \{\sigma_i(x)\}_{i=1}^3, \{\lambda_i(x)\}_{i=1}^2, \\ & b(x), V(x), u_b(x), u_V(x) \\ \text{subject to constraints } & (6.47) - (6.52). \end{aligned} \tag{6.53}$$

The following theorem provides guarantees for the solution of (6.53).

Theorem 6.3.1. *Consider Assumption 6.3.1, and further assume that a solution to* (6.53) *exists and is denoted by* $\{\sigma_i(x)\}_{i=1}^{3}$, $\{\lambda_i(x)\}_{i=1}^{2}$, $b(x)$, $V(x)$, $u_b(x)$, $u_V(x)$. *Then* $b(x)$ *is a CBF,* $V(x)$ *is a CLF, and they satisfy the relaxed compatibility condition as per Definition 6.2.3..*

Proof. We will prove that: (i) if the solution satisfies constraints (6.47), (6.48), and (6.49), then $b(x)$ is a CBF; (ii) if the solution satisfies constraints (6.47), (6.50), and (6.51), then $V(x)$ is a CLF; (iii) if the solution satisfies constraints (6.47), (6.49), and (6.52), then $b(x)$ and $V(x)$ satisfy relaxed compatibility condition. As such, if all constraints of (6.53) are satisfied, all assertions of the theorem follow.

(i) Equation (6.48) implies that for any $x \in \mathbb{R}^n$, $-b(x) + \sigma_1(x)s(x) \geq 0$, therefore for any $x \in \mathbb{R}^n$, $-b(x) + \sigma_1(x)s(x) > 0$. From this, we have for any $x \in \mathbb{R}^n$ such that $s(x) < 0$, we have that $b(x) < 0$. This implies that $\mathcal{B} \subseteq \mathcal{S}$. Equation (6.49) guarantees that for any x such that $b(x) = 0$, $\dot{b}(x) = \frac{\partial b(x)}{\partial x}(f(x) + g(x)u(x)) \geq 0$. Thus, $b(x)$ is a CBF.

(ii) Equation (6.50) implies that for any $x \in \mathcal{X}$, $\dot{V}(x) + \varepsilon_1(x) \leq 0$. (6.51) indicates $V(x) - \varepsilon_2(x) \geq 0, \forall x \in \mathcal{X}$. Given that $\varepsilon_1(x)$ and $\varepsilon_2(x)$ satisfy (6.46), we hence conclude that $V(x)$ is a CLF.

(iii) Equations (6.49), (6.52) imply that $\mathcal{L}_f V + \mathcal{L}_g V u_b \leq 0$, and $\mathcal{L}_f b + \mathcal{L}_g b u_b \geq 0$ for any x such that $b(x) = 0$. By Definition 6.2.3, $b(x)$ and $V(x)$ satisfy the relaxed compatibility condition. □

Naturally, the CLF $V(x)$ designed by (6.53) will satisfy the SCP if $u_V(x)$ has no constant term. The CLF constraint in (6.4) with $V(x)$ can satisfy the SCS by using $\gamma(x) = \varepsilon_1(x)/2$.

The program (6.53) cannot be transformed into a semi-definite program due to the cross-product of the decision variables, e.g. $\frac{\partial b(x)}{\partial x} g(x) u_b(x)$. One tractable way to solve the problem is using an alternating directional algorithm, which solves the problem by alternating between the decision variables in iterations to handle

bilinearities. The bilinearities in (6.53) come from $\frac{\partial b(x)}{\partial x}u_b(x)$, and $\lambda_1(x)b(x)$ in (6.49); $\frac{\partial V(x)}{\partial x}u_V(x)$, and $\sigma_2(x)b(x)$ in (6.50); $\lambda_2(x)b(x)$, and $\frac{\partial V(x)}{\partial x}g(x)u_b(x)$ in (6.52). The decision variables can be separated into two groups: (i) $b(x)$ and $V(x)$; (ii) the others. If either group of variables is fixed, In the sequel, we will use the superscript t to represent the corresponding fixed value of a decision variable at iteration t.

Algorithm 4 CLF and CBF design algorithm

Initialization Functions $b^0(x)$ and $V^0(x)$, $t = 1$.

Output: CBF $b(x)$ and CLF $V(x)$ that satisfy the relaxed compatibility condition (6.6)

1: **while** If (6.54) or (6.55) is infeasible **do**
2: **Fix** $b^{t-1}(x)$ and $V^{t-1}(x)$.
Solve (6.54) for $\sigma_2^t(x)$, $\{\lambda_i^t(x)\}_{i=1}^2$, $u_b^t(x)$, and $u_V^t(x)$).
3: **Fix** $\{\lambda_i^t(x)\}_{i=1}^2$, $u_b^t(x)$, and $u_V^t(x)$.
Solve (6.55) for $\{\sigma_i^t(x)\}_{i=1}^3$, $b^t(x)$, and $V^t(x)$.
4: **end while**

Here we propose Algorithm 4 to design a CLF $V(x)$ and CBF $b(x)$ that satisfy the relaxed compatibility condition 6.6. For initialization, we consider to design a valid local CLF $V^0(x)$ using the sum-of-squares techniques proposed by [98], and a valid CBF $b^0(x)$ using the methods proposed by [122]. In the algorithm, there are two main steps to iteratively solve two programs in each of which part of the variables in (6.53) are fixed. At Step 2, we fix $b(x)$ and $V(x)$ by $b^{t-1}(x)$ and $V^{t-1}(x)$, respectively, to derive a convex program. For the first iteration $t = 1$, $b^0(x)$ and $V^0(x)$ are obtained by the initialization step. The program is given by

$$\begin{aligned}
&\text{find } \sigma_2(x), \{\lambda_i(x)\}_{i=1}^2, u_b(x), u_V(x)) \\
&\sigma_2(x) \in \Sigma[x], \lambda_1(x), \lambda_2(x), u_b(x), u_V(x) \in \mathbb{R}[x], && \text{(6.54a)} \\
&\frac{\partial b^{t-1}(x)}{\partial x}(f(x) + g(x)u_b(x)) + \lambda_1(x)b^{t-1}(x) \in \Sigma[x], && \text{(6.54b)} \\
&- \frac{\partial V^{t-1}(x)}{\partial x}(f(x) + g(x)u_V(x)) - \\
&\sigma_2(x)w(x) - \varepsilon_1(x) \in \Sigma[x], && \text{(6.54c)} \\
&- \frac{\partial V^{t-1}(x)}{\partial x}(f(x) + g(x)u_b(x)) - \lambda_2(x)b(x) \in \Sigma[x]. && \text{(6.54d)}
\end{aligned}$$

After solving (6.54) at Step 2, we fix $\{\lambda_i(x)\}_{i=1}^2$, $u_b(x)$ and $u_V(x)$ by $\{\lambda_i^t(x)\}_{i=1}^2$, $u_b^t(x)$ and $u_V^t(x)$, respectively, and solve another convex program at Step 3. The program is given by

$$\text{find } \{\sigma_i(x)\}_{i=1}^3, b(x), V(x) \tag{6.55a}$$

$$\sigma_1(x), \sigma_2(x), \sigma_3(x) \in \Sigma[x], b(x), V(x) \in \mathbb{R}[x], b(0) > 0, \tag{6.55b}$$

$$-b(x) + \sigma_1(x)s(x) \in \Sigma[x], \tag{6.55c}$$

$$\frac{\partial b(x)}{\partial x}(f(x) + g(x)u_b^t(x)) + \lambda_1^t(x)b(x) \in \Sigma[x], \tag{6.55d}$$

$$-\frac{\partial V(x)}{\partial x}(f(x) + g(x)u_V^t(x)) - \sigma_2(x)w(x) - \varepsilon_1(x) \in \Sigma[x], \tag{6.55e}$$

$$V(x) - \varepsilon_2(x) - \sigma_3(x)w(x) \in \Sigma[x], \tag{6.55f}$$

$$-\frac{\partial V(x)}{\partial x}(f(x) + g(x)u_b^t(x) - \lambda_2^t(x)b(x)) \in \Sigma[x]. \tag{6.55g}$$

Programs (6.54) and (6.55) are both sum-of-squares programs; they are transformed into semi-definite programs and can be solved efficiently by an interior-point method. The algorithm terminates if both program (6.54) and (6.55) are feasible across the same iteration.

Given a CBF $b(x)$ and a CLF $V(x)$, we can check the relaxed compatibility by a single SOS program:

$$\text{find } \lambda_1(x), \lambda_2(x), u(x) \qquad \text{subject to}$$

$$\lambda_1(x), \lambda_2(x), u(x) \in \mathbb{R}[x], \tag{6.56a}$$

$$\frac{\partial b(x)}{\partial x}(f(x) + g(x)u(x)) + \lambda_1(x)b(x) \in \Sigma[x], \tag{6.56b}$$

$$-\frac{\partial V(x)}{\partial x}(f(x) + g(x)u(x)) - \lambda_2(x)b(x) \in \Sigma[x]. \tag{6.56c}$$

6.4 Simulation Results

In this section, we demonstrate the performance of our designed filter (6.4) by comparative studies over numerical examples. To solve the quadratic programming problem (6.4) we leverage CVX [172] for MATLAB. For the sum-of-squares programming problem we use SOSTOOLS v4.03 [173] to formulate the equivalent semi-definite programming problems, then solve these by using Mosek [47].

6.4.1 Benchmark Case

Consider the same setting as Example 6.2.1. The term $\beta(b(x))$ is chosen to be $\beta(b(x)) = \tanh(1000b(x))$. Relaxed compatibility of $b(x)$ and $V(x)$ can be validated by solving the proposed compatibility verification program (6.56).

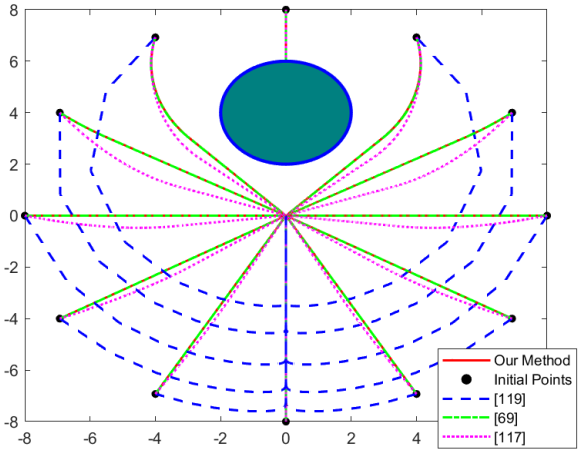


Figure 6.1: Trajectories of the closed-loop system with different methods. The black points are initial points, the green set is the obstacle. The penalty parameter p_d is set to 100 for the method of [117], [69], and our method. For the method of [119], we set $\epsilon = 0.01$. Our method, [117] and [119] achieve stabilization from all the initial points except for the top one, while [69] only achieves inexact convergence. From the very top initial point, all the trajectories converge to a point on the boundary of the obstacle. Safety is ensured by every method.

In Figure 6.1 we show trajectories from different initial points to compare our method (6.4) with (i) the method of [69]; (ii) the method of [117], and (iii) the method of [119]. Trajectories generated by our method are almost aligned with these generated by the method of [69]. The nominal controller $\pi(x)$ is set to zero to minimize the energy consumption in our method, the method of [69] and [119]. To achieve local stability, a stabilizing nominal controller $\pi(x) = [-2x_1, -2x_2]^\top$ is chosen for the method of [117]. The trajectories generated by our method are almost aligned with these by [69].

In Figure 6.2, we amplify the trajectories of our method and the methods of [69] method near the origin. It can be observed that our method achieves convergence to the origin, whilst [69] converges to equilibrium points away from the origin.

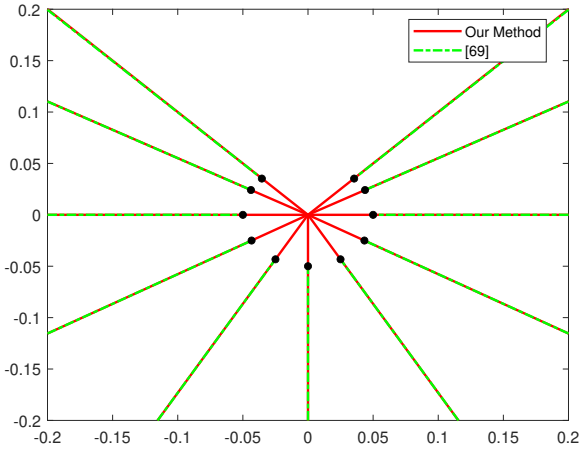


Figure 6.2: Trajectories near the origin, the red ones correspond to our method, while the green dotted ones correspond to the method of [69]. All the trajectories of our methods converge to the origin, while these of [69] converge to the black dots, which are equilibrium points away from the origin.

We then compare the filter's performance for different methods. Performance is measured by the magnitude of $||u^*(x)||^2$. Denote the optimal controller obtained from other methods by u_o^* (with u_o^* taking the value according to the different alternatives outlined above), and the one from our method by u^*. We conduct 100 experiments, in each of which we randomly pick a point from $\mathcal{B}$, and calculate the value of the optimal controller obtained from each method. To enable a comparison, we set the performance of our filter $||u^*(x)||^2$ to be the base line, and plot the relative performance difference $\log(||u_o^*(x)||^2/||u^*(x)||^2)$ in Figure 6.3. It can be seen that our filter shows a better performance than the method of [119] and [117], and similar (better in some experiments) performance as the method of [69]. Only in one experiment [119] exhibits better performance. We evaluate the controller $u^*(x)$ solved by the method of [119] at this point, find that $\mathcal{L}_f V(x)+\mathcal{L}_g V(x)u^*(x)+\gamma(x) > 0$. To have the CLF constraint satisfied for the method of [119] at this point, a smaller penalty coefficient ϵ, *i.e.*, $\epsilon = 0.001$ is required.

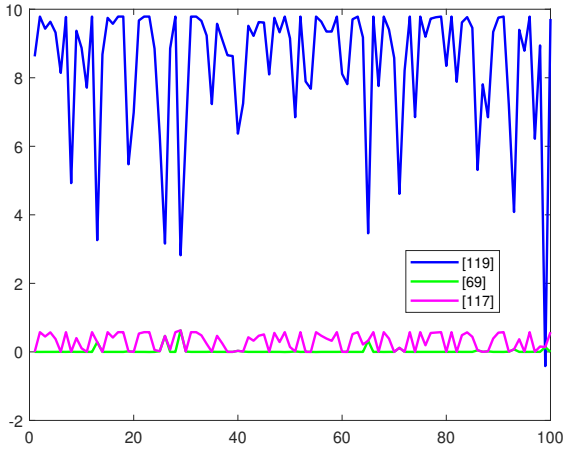


Figure 6.3: Comparison of filter performance by 100 Monte-Carlo experiments. The vertical axis represents $\log(||u_o^*(x)||^2/||u^*(x)||^2)$, while the horizontal axis represents number of experiments.

6.4.2 Polynomial System

Consider a second-order polynomial system with

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\end{bmatrix} = \begin{bmatrix}x_2\\ x_1 + \frac{1}{3}x_1^3 + x_2\end{bmatrix} + \begin{bmatrix}(0.2x_1^2 + 0.2x_2 + 1)u_1\\ (-0.2x_2^2 + 0.2x_1 + 4)u_2\end{bmatrix}. \tag{6.57}$$

The safe set is defined as $\mathcal{S} = \{x \in \mathbb{R}^2 : x_1^2 + (x_2 - 1)^2 - 0.25\}$. The local region is defined as $\mathcal{X} = -x_1^2 - x_2^2 + 100$. We first verify the candidate CLF $V(x) = x^\top x$ for this system. We have $\mathcal{L}_f V(x) = 4x_1x_2 + \frac{2}{3}x_1^3x_2 + 2x_2^2$, $\mathcal{L}_g V(x) = [0.4x_1^3 + 0.4x_1x_2 + 2x_1, -0.4x_2^3 + 0.4x_1x_2 + 8x_2]^\top$. At state $x_1 = 0, x_2 = \sqrt{20}$, we have $\mathcal{L}_g V(x) = 0$, $\mathcal{L}_f V(x) = 40 > 0$. By Definition 6.2.1, it follows that $V(x)$ is not a CLF.

Table 6.2: Degree of polynomial variables in (6.53).

$\sigma_1(x)$	$\lambda_1(x)$	$\lambda_2(x)$	$\sigma_2(x)$	$\sigma_3(x)$
2	1	4	8	8
$b(x)$	$V(x)$	$u_b(x)$	$u_V(x)$	
4	10	8	8	

Using the Algorithm 4, a CBF $b(x)$ and a CLF $V(x)$ that satisfy the relaxed compatibility condition 6.6 can be designed. Degrees of polynomials in (6.53) are shown in Table 6.2. The sum-of-squares polynomial $\varepsilon_1(x) = \varepsilon_2(x) = 0.1(x_1^2 + x_2^2)$.

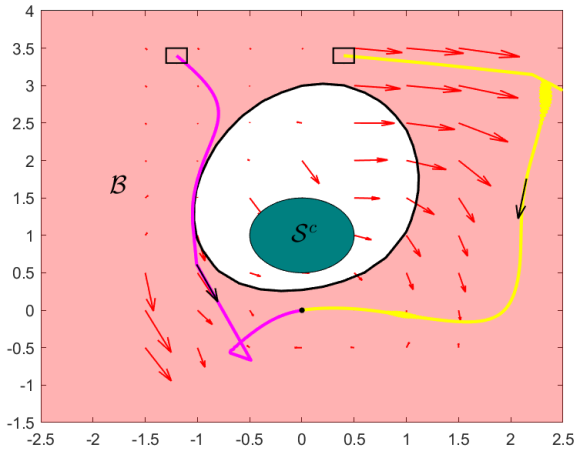


Figure 6.4: Phase portrait of system (6.57) using a CLF $V(x)$ and a CBF $b(x)$ that satisfy the *relaxed* compatibility condition. The controller $u^*(x)$ is synthesized by solving (6.4), using the designed CLF and CBF. The green set represents the obstacle. The control invariant set $\mathcal{B}$ is filled in red while its boundary curve $\partial\mathcal{B}$ is highlighted in black. The red arrows represent the vector field $f(x) + g(x)u^*(x)$. Two trajectories start from the black rectangle, avoid $\mathcal{B}^c$, and finally converge to the origin.

The control invariant set $\mathcal{B}$ and level sets of $V(x)$ are shown in Figure 6.4. It can be observed that the complementary set $\mathcal{B}^c$ is bounded that contains the obstacle, compatibility can not hold for this $b(x)$ and $V(x)$. The red arrows in the figure represent vector field $f(x) + g(x)u^*(x)$. On $\partial\mathcal{B}$, the arrows all point inwards $\mathcal{B}$, which reveals control invariance.

We then compare our method with [131], which proposes a SOS program to design $b(x)$ and $V(x)$ that satisfy the strict compatibility condition, following Defition 6.2.2. Here we use the same degrees of polynomials for $b(x)$ and $V(x)$, and still consider $\varepsilon_1(x) = \varepsilon_2(x) = 0.1(x_1^2 + x_2^2)$. Using the new $b(x)$, set $\mathcal{B}$ is shown in Figure 6.5. It can be seen that $\mathcal{B}$ is a bounded set, while our method generates an unbounded control invariant set. For this case, our method is shown to guarantee safety in a much larger region.

6.5 Conclusion

In this chapter, we have proposed a novel filter to design a safe and stable controller given a locally Lipschitz continuous reference signal. The filter is guaranteed to

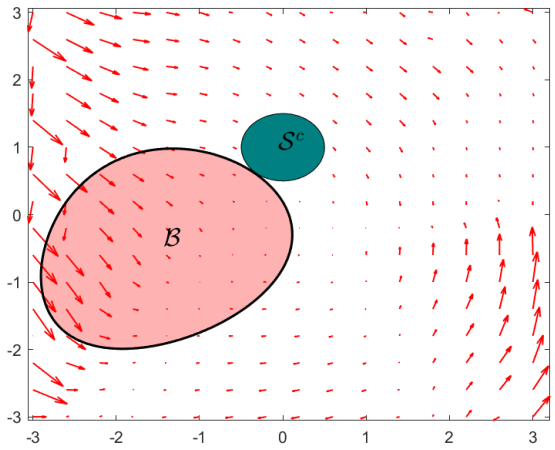


Figure 6.5: Phase portrait of system (6.57) using a CLF $V(x)$ and a CBF $b(x)$ that satisfy the *strict* compatibility condition (Definition 6.2.2), designed by the algorithm proposed in [131]. The green set represents the obstacle, while the control invariant set $\mathcal{B}$ is filled in red. The red arrows represent the vector field $f(x) + g(x)u^*(x)$.

be feasible if the CBF and the CLF satisfy a relaxed compatibility condition. We have shown that the closed-loop system is safe, and locally stable at the origin. Any level set of the CLF that does not intersect with the control invariant set is guaranteed to be a region of attraction. By characterizing the closed-form solution of the filter, we have shown that there are no interior equilibrium points except for the origin. Moreover, we show that the designed optimal controller is locally Lipschitz continuous under mild regularity conditions. To obtain a CLF and CBF that satisfy the relaxed compatibility condition, we developed a sum-of-squares program for nonlinear polynomial dynamics and a semi-algebraic safe set. Future work concentrates on analyzing the stability properties of the boundary equilibrium points, and considering control input saturation constraints.

7 Conclusion

Research into safety verification and control design is now more crucial than ever. In this thesis, we propose safety analysis and design methodologies for nonlinear and large-scale dynamical systems, utilizing advanced optimization techniques. Below, we summarize the key contributions of this thesis and outline future research directions.

7.1 Thesis Summary

Chapter 3 and 4: Co-design of CBFs and feedback controllers

In these chapters, we proposed co-design approaches for control barrier functions (CBFs) and feedback controllers using sum-of-squares programming and the Positivstellensatz (Psatz). We first investigated linear dynamical systems in Chapter 3. Two design approaches were proposed, namely global and local design, depending on the settings of the initial set and the safe set. For the case where the unsafe set is bounded on a subspace of the state space, we proposed parameterizing the CBF as a difference of convex quadratic functions. For the case where the safe set is bounded on a subspace and the initial set is also bounded, we proposed parameterizing the CBF as a convex quadratic function. In both cases, the feedback

controller is in an affine form. These parameterizations enable deriving a convex reformulation of nonconvex design constraints and handling mixed (high)-relative degree without the need for backstepping. We also explored extensions for possible input constraints, demonstrating the efficiency of the convex co-design approaches with numerical examples.

In Chapter 4, the above convex design approaches were extended to nonlinear systems. For convexification, we proposed different parameterizations for the CBF and the feedback controller, depending on the problem settings and the nonlinear dynamics. Similarly to the linear systems, we proposed convex conditions that can be incorporated into the design programs for input constraints. Given that the design programs are based on sum-of-squares programming and Psatz, the system dynamics are limited to polynomials. To overcome this limitation, we proposed using Taylor expansion for non-polynomial dynamics and treating the remainders as bounded disturbances in a robust design program. A major drawback of the convex programs is that we fixed the functional basis for parameterizations. Nonconvex programs that allow arbitrary polynomial bases were then proposed to reduce conservativeness.

Chapter 5: Distributed Safety Verification and Safe Control Design

Chapter 5 introduced a distributed framework for safe control design and safety verification using control barrier functions (CBFs). The proposed control design algorithm introduces auxiliary and slack variables to guarantee feasibility across iterations. Moreover, the algorithm is guaranteed to converge to a global optimal point asymptotically. Safety is ensured using the distributed controller as all the CBF constraints are satisfied. To accelerate computation, a truncated algorithm has been proposed to terminate the distributed algorithm before convergence. However, this may sacrifice safety as some of the CBF constraints may be violated. To analyze safety for multi-agent systems using such controllers, we have also proposed a distributed scenario program for probabilistic safety verification.

Chapter 6: Safe and Stable Filter Design

In this chapter, we have studied the problem of designing a safe and stable controller that tracks a reference input signal. Chapter 6 presents an optimal filter based on a control barrier function (CBF) and a control Lyapunov function (CLF) that satisfy the relaxed compatibility condition. The optimal controller is proven to be locally Lipschitz continuous while guaranteeing safety and local stability. Moreover, we have shown that there will be no undesired equilibrium points in the interior of the control invariant set. To obtain a CBF and a CLF that satisfy the relaxed compatibility condition, we proposed a design program using sum-of-squares optimization and an iterative algorithm to solve the underlying program. We also conducted comparison studies, demonstrating that our filter exhibits superior performance and results in a much larger control invariant set.

7.2 Future Work

There are many interesting directions to extend the scope of this thesis. Subsequently, we present two representatives.

Model-free Safe Control Synthesis

In this thesis, we have proposed several methods to design safe controllers for single and multi-agent systems, under different settings. All these methods are *model-based*, which require solving optimization problems that reply on the system dynamics. If the dynamics are accurately modeled, these model-based methods provide a rigorous safety guarantee. In real-world settings, however, high-fidelity system models are usually difficult to construct and influenced by external disturbances. For example, different robots with the same mechanical structure may share identical model configurations but could vary in kinetic model parameters due to manufacturing tolerances and the degree of wear on components. Using a biased dynamical model for control design can lead to a significant loss of safety guarantees when implementing such a controller on the nominal system. This is because the model discrepancy and uncertainty are propagating along the trajectory. Therefore,

it is of uttermost importance to develop safety-certified control techniques in the situation where part or all the components of the system dynamics are unknown.

This problem naturally lies in the scope of data-driven control, which tends to design control using proper measurements of the system. Data-driven control has shown great success in feedback stabilization and robust control design, but research towards safety is still in its early stages. Very recently, the authors of [174] proposed a data-driven safe control design framework by first identifying the system model from data. However, it was shown that utilizing the data directly with a predictive control design offers more advanced closed-loop behavior compared with an indirect approach that identifies the system model from data first [175]. When applying to the safe control problem, a direct data-driven design is very challenging since safety relates to both the system dynamics and state-wise constraints. Research on safety-certified control design directly from data remains an open problem. A recent attempt at this problem is [176], which proposed to refine a given safe controller directly from data for an unknown linear system. We argue that even finding a conservative safe controller is extremely arduous, given that the system model is unknown, and most of the real-world autonomous systems are nonlinear. Formulating the SOS constraints requires the dynamical system model, ensuring conditions for the CBF to hold. Despite its theoretical rigor, this method cannot be directly applied to the case when the system model is unknown, a common scenario encountered in the real-world situations.

For our future work, we will explore developing a data-based approach, as opposed to the model-based ones, to design a CBF without requiring knowledge of the dynamical system model. The key idea is to use the behavioral approach, which parameterizes the input-output behavior of the system based on data rather than the dynamical model [177]. According to Willem' s fundamental lemma, utilizing collected data with a persistent exciting input of sufficient length, any trajectory of the linear system can be parameterized by a linear transformation of the Hankel matrix composed of raw data. After collecting persistent exciting data via proper

measurements, we will utilize this data into the CBF conditions along with a data-parameterized state feedback control law to eliminate the dependence on the system model. Subsequently, these conditions will be transformed into data-driven SOS constraints. By solving the data-driven SOS program using the iterative algorithm and interior-point methods, a candidate CBF can be obtained if the algorithm converges to a feasible solution. To further enhance convergence of this data-driven CBF design algorithm, we will consider warm-starting the iterative algorithm. This can be accomplished by leveraging findings from Chapters 3, 4. The adoption of convex warm-starting will significantly improve the convergence and computational efficiency of the iterative algorithm. As an alternative to the method in Chapter 4, addressing challenges posed by complex nonlinear dynamics will also be explored using the recent proposed methodology on data-driven nonlinear cancellation [178].

Violation-free Distributed Optimization

Another interesting questions to be explored in the future is violation-free distributed optimization. In Chapter 5, we proposed a distributed optimization algorithm that guarantees feasibility of every local optimization problem across iterations. This is achieved by introducing slack variables into the local problems. However, the nominal constraints that are defined over the MAS may not be satisfied across iterations, as the slack variables may not be zero. Recently, this problem has been investigated in [179–181]. All these approaches, including ours, are aligned with the primal-decomposition framework [182]. In our experiments, we found that the practical convergence rate is not satisfactory for real-time applications, such as multi-robot collision avoidance. Other distributed optimization methods, such as primal-dual trades [183, 184], exhibit empirically faster convergence rates but do not guarantee constraint satisfaction across iterations. It is interesting to propose an accelerated primal-dual framework to solve distributed optimization problems while ensuring constraint violation-free operation.

References

[1] David Philip Miller. *The Life and Legend of James Watt: Collaboration, Natural Philosophy, and the Improvement of the Steam Engine*. University of Pittsburgh Press, 2019.

[2] Vance J VanDoren. "PID: still the one". In: *Control engineering* 50.10 (2003), pp. 32–+.

[3] Antonio Visioli. *Practical PID control*. Springer Science & Business Media, 2006.

[4] John C Doyle, Bruce A Francis, and Allen R Tannenbaum. *Feedback control theory*. Courier Corporation, 2013.

[5] Huibert Kwakernaak and Raphael Sivan. *Linear optimal control systems*. Vol. 1. Wiley-interscience New York, 1972.

[6] Stephen Boyd et al. *Linear matrix inequalities in system and control theory*. SIAM, 1994.

[7] Frank L Lewis, Draguna Vrabie, and Vassilis L Syrmos. *Optimal control*. John Wiley & Sons, 2012.

[8] Basil Kouvaritakis and Mark Cannon. "Model predictive control". In: *Switzerland: Springer International Publishing* 38 (2016), pp. 13–56.

[9] Kemin Zhou and John Comstock Doyle. *Essentials of robust control*. Vol. 104. Prentice hall Upper Saddle River, NJ, 1998.

[10] Pramod P Khargonekar, Ian R Petersen, and Kemin Zhou. "Robust stabilization of uncertain linear systems: quadratic stabilizability and H_∞ control theory". In: *IEEE Transactions on Automatic control* 35.3 (1990), pp. 356–361.

[11] IS Khalil, JC Doyle, and K Glover. *Robust and optimal control*. Prentice hall, 1996.

[12] Pascal Gahinet and Pierre Apkarian. "A linear matrix inequality approach to H_∞ control". In: *International journal of robust and nonlinear control* 4.4 (1994), pp. 421–448.

[13] Matthew M Peet, Antonis Papachristodoulou, and Sanjay Lall. "Positive forms and stability of linear time-delay systems". In: *SIAM Journal on Control and Optimization* 47.6 (2009), pp. 3237–3258.

[14] Emilia Fridman and Uri Shaked. "An improved stabilization method for linear time-delay systems". In: *IEEE transactions on automatic control* 47.11 (2002), pp. 1931–1937.

[15] Frédéric Gouaisbaut and Dimitri Peaucelle. "Delay-dependent stability analysis of linear time delay systems". In: *IFAC Proceedings Volumes* 39.10 (2006), pp. 54–59.

[16] Ben M Chen et al. "Composite nonlinear feedback control for linear systems with input saturation: theory and an application". In: *IEEE Transactions on automatic control* 48.3 (2003), pp. 427–439.

[17] Francesco Bullo and Andrew D Lewis. *Geometric control of mechanical systems: modeling, analysis, and design for simple mechanical control systems*. Vol. 49. Springer, 2019.

[18] Xuda Ding et al. "Safety-critical optimal control for robotic manipulators in a cluttered environment". In: *arXiv preprint arXiv:2211.04944* (2022).

[19] Scott Kuindersma et al. "Optimization-based locomotion planning, estimation, and control design for the atlas humanoid robot". In: *Autonomous robots* 40 (2016), pp. 429–455.

[20] Florian Dorfler and Francesco Bullo. "Synchronization and transient stability in power networks and nonuniform Kuramoto oscillators". In: *SIAM Journal on Control and Optimization* 50.3 (2012), pp. 1616–1642.

[21] Na Li, Lijun Chen, and Steven H Low. "Optimal demand response based on utility maximization in power networks". In: *2011 IEEE power and energy society general meeting*. IEEE. 2011, pp. 1–8.

[22] Alejandro F Villaverde, Antonio Barreiro, and Antonis Papachristodoulou. "Structural identifiability of dynamic systems biology models". In: *PLoS computational biology* 12.10 (2016), e1005153.

[23] Domitilla Del Vecchio, Aaron J Dy, and Yili Qian. "Control theory meets synthetic biology". In: *Journal of The Royal Society Interface* 13.120 (2016), p. 20160380.

[24] Anuradha M Annaswamy, Karl H Johansson, George J Pappas, et al. "Control for societal-scale challenges: Road map 2030". In: *IEEE Control Systems Society* (2023).

[25] Louis Breger and Jonathan P How. "Safe trajectories for autonomous rendezvous of spacecraft". In: *Journal of Guidance, Control, and Dynamics* 31.5 (2008), pp. 1478–1489.

[26] Dieter Fox, Wolfram Burgard, and Sebastian Thrun. "The dynamic window approach to collision avoidance". In: *IEEE Robotics & Automation Magazine* 4.1 (1997), pp. 23–33.

[27] Yan Wang et al. "A review on research status and key technologies of battery thermal management and its enhanced safety". In: *International Journal of Energy Research* 42.13 (2018), pp. 4008–4033.

[28] Jean-Pierre Aubin, Alexandre M Bayen, and Patrick Saint-Pierre. *Viability theory: new directions*. Springer Science & Business Media, 2011.

[29] John Lygeros. "On reachability and minimum cost optimal control". In: *Automatica* 40.6 (2004), pp. 917–927.

[30] Claire J Tomlin, John Lygeros, and S Shankar Sastry. "A game theoretic approach to controller design for hybrid systems". In: *Proceedings of the IEEE* 88.7 (2000), pp. 949–970.

[31] Kostas Margellos and John Lygeros. "Hamilton–Jacobi formulation for reach–avoid differential games". In: *IEEE Transactions on automatic control* 56.8 (2011), pp. 1849–1861.

[32] Jaime F Fisac et al. "Reach-avoid problems with time-varying dynamics, targets and constraints". In: *Proceedings of the 18th international conference on hybrid systems: computation and control*. 2015, pp. 11–20.

[33] Ian M Mitchell. "Comparing forward and backward reachability as tools for safety analysis". In: *International Workshop on Hybrid Systems: Computation and Control*. Springer. 2007, pp. 428–443.

[34] Sylvia Herbert et al. "Scalable learning of safety guarantees for autonomous systems using Hamilton-Jacobi reachability". In: *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE. 2021, pp. 5914–5920.

[35] Jeremy H Gillula et al. "Applications of hybrid reachability analysis to robotic aerial vehicles". In: *The International Journal of Robotics Research* 30.3 (2011), pp. 335–354.

[36] Sven R Schepp et al. "Sara: A tool for safe human-robot coexistence and collaboration through reachability analysis". In: *2022 International Conference on Robotics and Automation (ICRA)*. IEEE. 2022, pp. 4312–4317.

[37] Michael G Crandall, Lawrence C Evans, and P-L Lions. "Some properties of viscosity solutions of Hamilton-Jacobi equations". In: *Transactions of the American Mathematical Society* 282.2 (1984), pp. 487–502.

[38] Ian M Mitchell. "The flexible, extensible and efficient toolbox of level set methods". In: *Journal of Scientific Computing* 35 (2008), pp. 300–329.

[39] Kim P Wabersich et al. "Data-driven safety filters: Hamilton-jacobi reachability, control barrier functions, and predictive methods for uncertain systems". In: *IEEE Control Systems Magazine* 43.5 (2023), pp. 137–177.

[40] Mo Chen et al. "Decomposition of reachable sets and tubes for a class of nonlinear systems". In: *IEEE Transactions on Automatic Control* 63.11 (2018), pp. 3675–3688.

[41] Sylvia L Herbert et al. "Reachability-based safety guarantees using efficient initializations". In: *2019 IEEE 58th Conference on Decision and Control (CDC)*. IEEE. 2019, pp. 4810–4816.

[42] David Q Mayne et al. "Constrained model predictive control: Stability and optimality". In: *Automatica* 36.6 (2000), pp. 789–814.

[43] Manfred Morari and Jay H Lee. "Model predictive control: past, present and future". In: *Computers & chemical engineering* 23.4-5 (1999), pp. 667–682.

[44] Kazumune Hashimoto, Shuichi Adachi, and Dimos V Dimarogonas. "A collision-free communication scheduling for nonlinear model predictive control". In: *IFAC-PapersOnLine* 50.1 (2017), pp. 8939–8944.

[45] Alina Eqtami, Dimos V Dimarogonas, and Kostas J Kyriakopoulos. "Novel event-triggered strategies for model predictive controllers". In: *2011 50th IEEE Conference on Decision and Control and European Control Conference*. IEEE. 2011, pp. 3392–3397.

[46] Alexandros Nikou and Dimos V Dimarogonas. "Decentralized tube-based model predictive control of uncertain nonlinear multiagent systems". In: *International Journal of Robust and Nonlinear Control* 29.10 (2019), pp. 2799–2818.

[47] MOSEK ApS. *The MOSEK optimization toolbox for MATLAB manual. Version 10.1.* 2024. URL: `http://docs.mosek.com/latest/toolbox/index.html`.

[48] Artemiy Oleinikov et al. "Safety-aware nonlinear model predictive control for physical human-robot interaction". In: *IEEE Robotics and Automation Letters* 6.3 (2021), pp. 5665–5672.

[49] Pierre-Brice Wieber, Russ Tedrake, and Scott Kuindersma. "Modeling and control of legged robots". In: *Springer handbook of robotics*. Springer, 2016, pp. 1203–1234.

[50] Francesco Cursi, Valerio Modugno, and Petar Kormushev. "Model predictive control for a tendon-driven surgical robot with safety constraints in kinematics

and dynamics". In: *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2020, pp. 7653–7660.

[51] Paolo Falcone et al. "A hierarchical model predictive control framework for autonomous ground vehicles". In: *2008 American Control Conference*. IEEE. 2008, pp. 3719–3724.

[52] Wilbur Langson et al. "Robust model predictive control using tubes". In: *Automatica* 40.1 (2004), pp. 125–133.

[53] David Q Mayne, Eric C Kerrigan, and Paola Falugi. "Robust model predictive control: advantages and disadvantages of tube-based methods". In: *IFAC Proceedings Volumes* 44.1 (2011), pp. 191–196.

[54] Mark Cannon et al. "Robust tubes in nonlinear model predictive control". In: *IEEE Transactions on Automatic Control* 56.8 (2011), pp. 1942–1947.

[55] Jack Umenberger. "Closed-loop data-enabled predictive control". In: *2021 American Control Conference (ACC)*. IEEE. 2021, pp. 3358–3365.

[56] Jeremy Coulson, John Lygeros, and Florian Dörfler. "Data-enabled predictive control: In the shallows of the DeePC". In: *2019 18th European Control Conference (ECC)*. IEEE. 2019, pp. 307–312.

[57] Julian Berberich et al. "Data-driven model predictive control with stability and robustness guarantees". In: *IEEE Transactions on Automatic Control* 66.4 (2020), pp. 1702–1717.

[58] Ali Mesbah. "Stochastic model predictive control: An overview and perspectives for future research". In: *IEEE Control Systems Magazine* 36.6 (2016), pp. 30–44.

[59] Sadra Sadraddini and Calin Belta. "Robust temporal logic model predictive control". In: *2015 53rd Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. IEEE. 2015, pp. 772–779.

[60] Lars Lindemann and Dimos V Dimarogonas. "Robust motion planning employing signal temporal logic". In: *2017 American Control Conference (ACC)*. IEEE. 2017, pp. 2950–2955.

[61] Lars Lindemann and Dimos V Dimarogonas. "Robust control for signal temporal logic specifications using discrete average space robustness". In: *Automatica* 101 (2019), pp. 377–387.

[62] Liqun Zhao, Konstantinos Gatsis, and Antonis Papachristodoulou. "Stable and Safe Reinforcement Learning via a Barrier-Lyapunov Actor-Critic Approach". In: *2023 62nd IEEE Conference on Decision and Control (CDC)*. IEEE. 2023, pp. 1320–1325.

[63] Yang Shi and Kunwu Zhang. "Advanced model predictive control framework for autonomous intelligent mechatronic systems: A tutorial overview and perspectives". In: *Annual Reviews in Control* 52 (2021), pp. 170–196.

[64] Daniel Limón et al. "On the stability of constrained MPC without terminal constraint". In: *IEEE transactions on automatic control* 51.5 (2006), pp. 832–836.

[65] Stephen Prajna and Ali Jadbabaie. "Safety verification of hybrid systems using barrier certificates". In: *International Workshop on Hybrid Systems: Computation and Control*. Springer. 2004, pp. 477–492.

[66] Stephen Prajna. "Barrier certificates for nonlinear model validation". In: *Automatica* 42.1 (2006), pp. 117–126.

[67] Stephen Prajna, Ali Jadbabaie, and George J Pappas. "A framework for worst-case and stochastic safety verification using barrier certificates". In: *IEEE Transactions on Automatic Control* 52.8 (2007), pp. 1415–1428.

[68] Aaron D Ames, Jessy W Grizzle, and Paulo Tabuada. "Control barrier function based quadratic programs with application to adaptive cruise control". In: *53rd IEEE Conference on Decision and Control*. IEEE. 2014, pp. 6271–6278.

[69] Aaron D Ames et al. "Control barrier function based quadratic programs for safety critical systems". In: *IEEE Transactions on Automatic Control* 62.8 (2016), pp. 3861–3876.

[70] Franco Blanchini. "Set invariance in control". In: *Automatica* 35.11 (1999), pp. 1747–1767.

[71] Mitio Nagumo. "Über die lage der integralkurven gewöhnlicher differentialgleichungen". In: *Proceedings of the Physico-Mathematical Society of Japan. 3rd Series* 24 (1942), pp. 551–559.

[72] Pierre-François Massiani et al. "Safe value functions". In: *IEEE Transactions on Automatic Control* (2022).

[73] Jason J Choi et al. "Robust control barrier–value functions for safety-critical control". In: *2021 60th IEEE Conference on Decision and Control (CDC)*. IEEE. 2021, pp. 6814–6821.

[74] Shao-Chen Hsu, Xiangru Xu, and Aaron D Ames. "Control barrier function based quadratic programs with application to bipedal robotic walking". In: *2015 American Control Conference (ACC)*. IEEE. 2015, pp. 4542–4548.

[75] Andrew Clark. "Control barrier functions for stochastic systems". In: *Automatica* 130 (2021), p. 109688.

[76] Andrew Singletary, Mohamadreza Ahmadi, and Aaron D Ames. "Safe control for nonlinear systems with stochastic uncertainty via risk control barrier functions". In: *IEEE Control Systems Letters* 7 (2022), pp. 349–354.

[77] Mrdjan Jankovic. "Robust control barrier functions for constrained stabilization of nonlinear systems". In: *Automatica* 96 (2018), pp. 359–367.

[78] Shishir Kolathaya and Aaron D Ames. "Input-to-state safety with control barrier functions". In: *IEEE control systems letters* 3.1 (2018), pp. 108–113.

[79] Wei Xiao and Calin Belta. "Control barrier functions for systems with high relative degree". In: *2019 IEEE 58th conference on decision and control (CDC)*. IEEE. 2019, pp. 474–479.

[80] Xiao Tan, Wenceslao Shaw Cortez, and Dimos V Dimarogonas. "High-Order Barrier Functions: Robustness, Safety, and Performance-Critical Control". In: *IEEE Transactions on Automatic Control* 67.6 (2021), pp. 3021–3028.

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

[82] Lars Lindemann and Dimos V Dimarogonas. "Control barrier functions for signal temporal logic tasks". In: *IEEE control systems letters* 3.1 (2018), pp. 96–101.

[83] Lars Lindemann and Dimos V Dimarogonas. "Control barrier functions for multi-agent systems under conflicting local signal temporal logic tasks". In: *IEEE control systems letters* 3.3 (2019), pp. 757–762.

[84] Jun Zeng et al. "Safety-critical control using optimal-decay control barrier function with guaranteed point-wise feasibility". In: *2021 American Control Conference (ACC)*. IEEE. 2021, pp. 3856–3863.

[85] Zhe Wu et al. "Control lyapunov-barrier function-based model predictive control of nonlinear systems". In: *Automatica* 109 (2019), p. 108508.

[86] Somil Bansal et al. "Hamilton-jacobi reachability: A brief overview and recent advances". In: *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*. IEEE. 2017, pp. 2242–2253.

[87] Yujie Yang et al. "Synthesizing Control Barrier Functions With Feasible Region Iteration for Safe Reinforcement Learning". In: *IEEE Transactions on Automatic Control* (2023).

[88] Liqun Zhao et al. "NLBAC: A Neural Ordinary Differential Equations-based Framework for Stable and Safe Reinforcement Learning". In: *arXiv preprint arXiv:2401.13148* (2024).

[89] Max H Cohen and Calin Belta. "Safe exploration in model-based reinforcement learning using control barrier functions". In: *Automatica* 147 (2023), p. 110684.

[90] Jaime F Fisac et al. "Bridging hamilton-jacobi safety analysis and reinforcement learning". In: *2019 International Conference on Robotics and Automation (ICRA)*. IEEE. 2019, pp. 8550–8556.

[91] Pablo A Parrilo. *Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization*. California Institute of Technology, 2000.

[92] A. Papachristodoulou et al. *SOSTOOLS: Sum of squares optimization toolbox for MATLAB*. Available from `http://www.eng.ox.ac.uk/control/sostools`, `http://www.cds.caltech.edu/sostools` and `http://www.mit.edu/~parrilo/sostools`. `http://arxiv.org/abs/1310.4716`, 2013.

[93] Johan Lofberg. "YALMIP: A toolbox for modeling and optimization in MATLAB". In: *2004 IEEE international conference on robotics and automation (IEEE Cat. No. 04CH37508)*. IEEE. 2004, pp. 284–289.

[94] Mosek ApS. "Mosek optimization toolbox for matlab". In: *User' s Guide and Reference Manual, Version* 4.1 (2019).

[95] Jos F Sturm. "Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones". In: *Optimization methods and software* 11.1-4 (1999), pp. 625–653. URL: `%5Curl%7Bhttps://doi.org/10.1080/10556789908805766%7D`.

[96] Antonis Papachristodoulou and Stephen Prajna. "On the construction of Lyapunov functions using the sum of squares decomposition". In: *Proceedings of the 41st IEEE Conference on Decision and Control, 2002.* Vol. 3. IEEE. 2002, pp. 3482–3487.

[97] Stephen Prajna, Antonis Papachristodoulou, and Fen Wu. "Nonlinear control synthesis by sum of squares optimization: A Lyapunov-based approach". In: *2004 5th Asian control conference (IEEE Cat. No. 04EX904)*. Vol. 1. IEEE. 2004, pp. 157–165.

[98] James Anderson and Antonis Papachristodoulou. "ADVANCES IN COMPUTATIONAL LYAPUNOV ANALYSIS USING SUM-OF-SQUARES PROGRAMMING." In: *Discrete & Continuous Dynamical Systems-Series B* 20.8 (2015).

[99] Anders Rantzer. "A dual to Lyapunov's stability theorem". In: *Systems & Control Letters* 42.3 (2001), pp. 161–168.

[100] Yuxiao Chen, Mohamadreza Ahmadi, and Aaron D Ames. "Optimal safe controller synthesis: A density function approach". In: *2020 American Control Conference (ACC)*. IEEE. 2020, pp. 5407–5412.

[101] Andrew Zheng, Sriram SKS Narayanan, and Umesh Vaidya. "Safe navigation using density functions". In: *IEEE Robotics and Automation Letters* (2023).

[102] Xiangru Xu. "Constrained control of input–output linearizable systems using control sharing barrier functions". In: *Automatica* 87 (2018), pp. 195–201.

[103] Andrew Clark. "Verification and synthesis of control barrier functions". In: *2021 60th IEEE Conference on Decision and Control (CDC)*. IEEE. 2021, pp. 6105–6112.

[104] Andrew Clark. "A Semi-Algebraic Framework for Verification and Synthesis of Control Barrier Functions". In: *arXiv preprint arXiv:2209.00081* (2022).

[105] Li Wang, Dongkun Han, and Magnus Egerstedt. "Permissive barrier certificates for safe stabilization using sum-of-squares". In: *2018 Annual American Control Conference (ACC)*. IEEE. 2018, pp. 585–590.

[106] Mehran Mesbahi, Michael G Safonov, and George P Papavassilopoulos. "Bilinearity and complementarity in robust control". In: *Advances in linear matrix inequality methods in control*. SIAM, 2000, pp. 269–292.

[107] Carl-Magnus Fransson and Bengt Lennartson. "Low order multicriteria/spl Hscr//sub/spl infin//design via bilinear matrix inequalities". In: *42nd IEEE International Conference on Decision and Control (IEEE Cat. No. 03CH37475)*. Vol. 5. IEEE. 2003, pp. 5161–5167.

[108] Arash Hassibi, Jonathan How, and Stephen Boyd. "A path-following method for solving BMI problems in control". In: *Proceedings of the 1999 American control conference (Cat. No. 99CH36251)*. Vol. 2. IEEE. 1999, pp. 1385–1389.

[109] Yurii Nesterov and Arkadii Nemirovskii. *Interior-point polynomial algorithms in convex programming*. SIAM, 1994.

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

[111] Yuxiao Chen, Andrew Singletary, and Aaron D Ames. "Guaranteed obstacle avoidance for multi-robot operations with limited actuation: A control barrier function approach". In: *IEEE Control Systems Letters* 5.1 (2020), pp. 127–132.

[112] Urs Borrmann et al. "Control barrier certificates for safe swarm behavior". In: *IFAC-PapersOnLine* 48.27 (2015), pp. 68–73.

[113] Xiao Tan and Dimos V Dimarogonas. "Distributed Implementation of Control Barrier Functions for Multi-agent Systems". In: *IEEE Control Systems Letters* 6 (2021), pp. 1879–1884.

[114] Xiao Tan and Dimos V Dimarogonas. "On the undesired equilibria induced by control barrier function based quadratic programs". In: *Automatica* 159 (2024), p. 111359.

[115] Matheus F Reis, A Pedro Aguiar, and Paulo Tabuada. "Control barrier function-based quadratic programs introduce undesirable asymptotically stable equilibria". In: *IEEE Control Systems Letters* 5.2 (2020), pp. 731–736.

[116] Muhammad Zakiyullah Romdlony and Bayu Jayawardhana. "Stabilization with guaranteed safety using control Lyapunov–barrier function". In: *Automatica* 66 (2016), pp. 39–47.

[117] Xiao Tan and Dimos V Dimarogonas. "On the undesired equilibria induced by control barrier function based quadratic programs". In: *Automatica* 159 (2024), p. 111359.

[118] Wenceslao Shaw Cortez and Dimos V Dimarogonas. "On compatibility and region of attraction for safe, stabilizing control laws". In: *IEEE Transactions on Automatic Control* 67.9 (2022), pp. 4924–4931.

[119] Pol Mestres and Jorge Cortés. "Optimization-based safe stabilizing feedback with guaranteed region of attraction". In: *IEEE Control Systems Letters* 7 (2022), pp. 367–372.

[120] Michael Schneeberger, Florian Dörfler, and Silvia Mastellone. "SOS construction of compatible control Lyapunov and barrier functions". In: *IFAC-PapersOnLine* 56.2 (2023), pp. 10428–10434.

[121] Philipp Braun and Christopher M Kellett. "Comment on "stabilization with guaranteed safety using control Lyapunov–barrier function" ". In: *Automatica* 122 (2020), p. 109225.

[122] Han Wang, Kostas Margellos, and Antonis Papachristodoulou. "Safety verification and controller synthesis for systems with input constraints". In: *IFAC-PapersOnLine* 56.2 (2023), pp. 1698–1703.

[123] Hongkai Dai and Frank Permenter. "Convex synthesis and verification of control-Lyapunov and barrier functions with input constraints". In: *2023 American Control Conference (ACC)*. IEEE. 2023, pp. 4116–4123.

[124] Shucheng Kang et al. "Verification and synthesis of robust control barrier functions: Multilevel polynomial optimization and semidefinite relaxation". In: *2023 62nd IEEE Conference on Decision and Control (CDC)*. IEEE. 2023, pp. 8215–8222.

[125] Binghan He and Takashi Tanaka. "Barrier Pairs for Safety Control of Uncertain Output Feedback Systems". In: *2023 American Control Conference (ACC)*. IEEE. 2023, pp. 3669–3674.

[126] H. Wang et al. "Convex co-design of control barrier functions and safe feedback controllers under input constraints". In: *under review* (2024).

[127] H. Wang, K. Margellos, and A. Papachristodoulou. *Assessing safety for control systems using sum-of-squares programming*. Polynomial Optimization, Moments, and Applications (M. Kocvara, B. Mourrain, C. Riener, eds.), Springer-Verlag, 2023.

[128] Simone Garatti and Marco C Campi. "Risk and complexity in scenario optimization". In: *Mathematical Programming* (2019), pp. 1–37.

[129] H. Wang, A. Papachristodoulou, and K. Margellos. "Distributed safe control design and safety verification for multi-agent systems". In: *under review* (2024).

[130] H. Wang, A. Papachristodoulou, and K. Margellos. "Distributed control design and safety verification for multi-agent systems". In: *IEEE Conference on Decision and Control*. 2023, pp. 1–6.

[131] Michael Schneeberger, Silvia Mastellone, and Florian Dörfler. "Advanced safety filter based on SOS Control Barrier and Lyapunov Functions". In: *arXiv preprint arXiv:2401.06901* (2024).

[132] H. Wang, K. Margellos, and A. Papachristodoulou. "Safe and Stable Filter Design Using a Relaxed Compatibitlity Control Barrier –Lyapunov Condition". In: *under review* (2024).

[133] Han Wang, Kostas Margellos, and Antonis Papachristodoulou. "Relaxed Compatibility Between Control Barrier and Lyapunov Functions". In: *2024 UKACC 14th International Conference on Control (CONTROL)*. IEEE. 2024, pp. 125–126.

[134] Amir Ali Ahmadi et al. "NP-hardness of deciding convexity of quartic polynomials and related problems". In: *Mathematical programming* 137 (2013), pp. 453–476.

[135] Jacek Bochnak, Michel Coste, and Marie-Françoise Roy. *Real algebraic geometry*. Vol. 36. Springer Science & Business Media, 2013.

[136] Pablo Parrilo and Sanjay Lall. "The Positivestellensatz". In: *European Control Conference* (2003).

[137] Julius Farkas. "Uber die Theorie des einfachen Ungleichungen, J. Reine Angew. Mathematik". In: *Reine Angew* (1902).

[138] Peter Wieland and Frank Allgöwer. "Constructive safety using control barrier functions". In: *IFAC Proceedings Volumes* 40.12 (2007), pp. 462–467.

[139] Miroslav Krstic. "Inverse optimal safety filters". In: *IEEE Transactions on Automatic Control* (2023).

[140] Andrew J Taylor et al. "Safe Backstepping with Control Barrier Functions". In: *2022 IEEE 61st Conference on Decision and Control (CDC)*. IEEE. 2022, pp. 5775–5782.

[141] Ian M Mitchell et al. "A toolbox of level set methods". In: *UBC Department of Computer Science Technical Report TR-2007-11* 1 (2007), p. 6.

[142] Paul Glotfelter, Jorge Cortés, and Magnus Egerstedt. "Nonsmooth barrier functions with applications to multi-robot systems". In: *IEEE control systems letters* 1.2 (2017), pp. 310–315.

[143] N Dinh and V Jeyakumar. "Farkas' lemma: three decades of generalizations for mathematical optimization". In: *Top* 22.1 (2014), pp. 1–22.

[144] R Tyrrell Rockafellar and Roger J-B Wets. *Variational analysis*. Vol. 317. Springer Science & Business Media, 2009.

[145] Andrea Bisoffi, Claudio De Persis, and Pietro Tesi. "Data-driven control via Petersen' s lemma". In: *Automatica* 145 (2022), p. 110537.

[146] Meichen Guo, Claudio De Persis, and Pietro Tesi. "Data-driven stabilizer design and closed-loop analysis of general nonlinear systems via Taylor's expansion". In: *arXiv preprint arXiv:2209.01071* (2022).

[147] Xiangru Xu et al. "Correctness guarantees for the composition of lane keeping and adaptive cruise control". In: *IEEE Transactions on Automation Science and Engineering* 15.3 (2017), pp. 1216–1229.

[148] Torbjørn Cunis and Ilya Kolmanovsky. "Viability, viscosity, and storage functions in model-predictive control with terminal constraints". In: *Automatica* 131 (2021), p. 109748.

[149] Alessandro Luppi et al. "Data-driven design of safe control for polynomial systems". In: *European Journal of Control* 75 (2024), p. 100914.

[150] Fan Ding et al. "Configuration-Aware Safe Control for Mobile Robotic Arm with Control Barrier Functions". In: *arXiv preprint arXiv:2204.08265* (2022).

[151] Han Wang et al. "Moving Obstacle Avoidance and Topology Recovery for Multi-agent Systems". In: *American Control Conference (ACC)*. IEEE. 2019, pp. 2696–2701.

[152] Jakob Axelsson. "Safety in vehicle platooning: A systematic literature review". In: *IEEE Transactions on Intelligent Transportation Systems* 18.5 (2016), pp. 1033–1045.

[153] Assad Alam et al. "Guaranteeing safety for heavy duty vehicle platooning: Safe set computations and experimental evaluations". In: *Control Engineering Practice* 24 (2014), pp. 33–41.

[154] Wei Xiao and Christos G Cassandras. "Decentralized optimal merging control for Connected and Automated Vehicles with safety constraint guarantees". In: *Automatica* 123 (2021), p. 109333.

[155] David G Wright. "Tychonoff' s theorem". In: *Proceedings of the American Mathematical Society* 120.3 (1994), pp. 985–987.

[156] Xiao Tan and Dimos V Dimarogonas. "Compatibility checking of multiple control barrier functions for input constrained systems". In: *2022 IEEE 61st Conference on Decision and Control (CDC)*. IEEE. 2022, pp. 939–944.

[157] James Usevitch, Kunal Garg, and Dimitra Panagou. "Strong invariance using control barrier functions: A Clarke tangent cone approach". In: *2020 59th IEEE Conference on Decision and Control (CDC)*. IEEE. 2020, pp. 2044–2049.

[158] Axton Isaly et al. "On the feasibility and continuity of feedback controllers defined by multiple control barrier functions". In: *IEEE Transactions on Automatic Control* (2024).

[159] Kunal Garg et al. "Advances in the Theory of Control Barrier Functions: Addressing practical challenges in safe control synthesis for autonomous and robotic systems". In: *Annual Reviews in Control* 57 (2024), p. 100945.

[160] Ivano Notarnicola and Giuseppe Notarstefano. "Constraint-coupled distributed optimization: A relaxation and duality approach". In: *IEEE Transactions on Control of Network Systems* 7.1 (2019), pp. 483–492.

[161] Angelia Nedić and Asuman Ozdaglar. "Approximate primal solutions and rate analysis for dual subgradient methods". In: *SIAM Journal on Optimization* 19.4 (2009), pp. 1757–1780.

[162] Giuseppe Notarstefano, Ivano Notarnicola, Andrea Camisa, et al. "Distributed optimization for smart cyber-physical networks". In: *Foundations and Trends® in Systems and Control* 7.3 (2019), pp. 253–383.

[163] Stephen Boyd, Stephen P Boyd, and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[164] Sham Kakade, Shai Shalev-Shwartz, Ambuj Tewari, et al. "On the duality of strong convexity and strong smoothness: Learning applications and matrix regularization". In: *Unpublished Manuscript, http://ttic. uchicago. edu/shai/papers/KakadeShalevTewari09. pdf* 2.1 (2009), p. 35.

[165] Yurii Nesterov. *Introductory lectures on convex optimization: A basic course*. Vol. 87. Springer Science & Business Media, 2003.

[166] Kostas Margellos et al. "Distributed constrained optimization and consensus in uncertain networks via proximal minimization". In: *IEEE Transactions on Automatic Control* 63.5 (2017), pp. 1372–1387.

[167] Andrew Alleyne et al. "Control for Societal-scale Challenges: Road Map 2030". In: *2022 IEEE CSS Workshop on Control for Societal-Scale Challenges*. IEEE Control Systems Society. 2023.

[168] Hassan K Khalil. "Lyapunov stability". In: *Control systems, robotics and automation* 12 (2009), p. 115.

[169] Zvi Artstein. "Stabilization with relaxed controls". In: *Nonlinear Analysis: Theory, Methods & Applications* 7.11 (1983), pp. 1163–1173.

[170] Eduardo D Sontag et al. "Smooth stabilization implies coprime factorization". In: *IEEE transactions on automatic control* 34.4 (1989), pp. 435–443.

[171] Han Wang, Kostas Margellos, and Antonis Papachristodoulou. "Explicit solutions for safety problems using control barrier functions". In: *2022 IEEE 61st Conference on Decision and Control (CDC)*. IEEE. 2022, pp. 5680–5685.

[172] Michael Grant and Stephen Boyd. *CVX: Matlab software for disciplined convex programming, version 2.1*. 2014.

[173] Antonis Papachristodoulou et al. "SOSTOOLS version 4.00 sum of squares optimization toolbox for MATLAB". In: *arXiv preprint arXiv:1310.4716* (2013).

[174] Alessandro Luppi et al. "Data-driven design of safe control for polynomial systems". In: *European Journal of Control* 75 (2024), p. 100914.

[175] Florian Dörfler, Pietro Tesi, and Claudio De Persis. "On the certainty-equivalence approach to direct data-driven LQR design". In: *IEEE Transactions on Automatic Control* (2023).

[176] Marjan Khaledi, Pouria Tooranjipour, and Bahare Kiumarsi. "Data-Driven Safety-Certified Predictive Control for Linear Systems". In: *IEEE Control Systems Letters* (2023).

[177] Claudio De Persis and Pietro Tesi. "Formulas for data-driven control: Stabilization, optimality, and robustness". In: *IEEE Transactions on Automatic Control* 65.3 (2019), pp. 909–924.

[178] Claudio De Persis et al. "Data-Driven Feedback Linearization with Complete Dictionaries". In: *2023 62nd IEEE Conference on Decision and Control (CDC)*. IEEE. 2023, pp. 3037–3042.

[179] Xiao Tan et al. "A continuous-time violation-free multi-agent optimization algorithm and its applications to safe distributed control". In: *arXiv preprint arXiv:2404.07571* (2024).

[180] Ahmed Allibhoy and Jorge Cortés. "Control barrier function-based design of gradient flows for constrained nonlinear programming". In: *IEEE Transactions on Automatic Control* (2023).

[181] Changxin Liu et al. "Achieving violation-free distributed optimization under coupling constraints". In: *arXiv preprint arXiv:2404.07609* (2024).

[182] Andrea Camisa et al. "Distributed constraint-coupled optimization via primal decomposition over random time-varying graphs". In: *Automatica* 131 (2021), p. 109739.

[183] Allan Andre do Nascimento, Antonis Papachristodoulou, and Kostas Margellos. "A game theoretic approach for safe and distributed control of unmanned aerial vehicles". In: *2023 62nd IEEE Conference on Decision and Control (CDC)*. IEEE. 2023, pp. 1070–1075.

[184] Guido Carnevale et al. "Tracking-based distributed equilibrium seeking for aggregative games". In: *IEEE Transactions on Automatic Control* (2024).