Learning-Based Safety-Critical Control Under Uncertainty with Applications to Mobile Robots

by

Mohammad Aali

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Applied Mathematics

Waterloo, Ontario, Canada, 2025

© Mohammad Aali 2025

Examining Committee Membership

The following served on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

External Examiner:	Xiangru Xu Assistant Professor, Department of Mechanical Engineering, Department of Electrical and Computer Engineering (by courtesy), University of Wisconsin-Madison
Supervisor:	Jun Liu Professor, Department of Applied Mathematics
Internal Members:	Xinzhi Liu Professor, Department of Applied Mathematics
	Christopher Nielsen Professor, Department of Electrical and Computer Engineering
Internal-External Member:	Amir Khajepour Professor, Department of Mechanical and Mechatronics Engineering

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Control theory is one of the key ingredients of the remarkable rise in robotics. Due to technological advancements, the use of automated robots, which was once primarily limited to industrial and manufacturing settings, has now expanded to impact many different parts of everyday life. Various control strategies have been developed to satisfy a wide range of performance criteria arising from recent applications. These strategies have different characteristics depending on the problem they solve. But, they all have to guarantee stability before satisfying any performance-driven criteria. However, as robotic technologies become increasingly integrated into everyday life, they introduce safety concerns. For autonomous systems to be trusted by the public, they must guarantee safety. In recent years, the concept of set invariance has been incorporated into modern control strategies to enable systematic safety guarantees. In this thesis, we aim to develop safety-critical control methods that can guarantee safety while satisfying performance-driven requirements. In the proposed strategies, we considered formal safety guarantees, robustness to uncertainty, and computational efficiency to be the highest design priorities. Each of them introduces new challenges which are addressed with theoretical contributions. We selected motion control in mobile robots as a use case for proposed controllers which is an active area of research integrating safety, stability, and performance in various scenarios. In particular, we focused on multi-body mobile robots, an area with limited research on safe operation. We provide a comprehensive survey of the recent methods that formalize safety for the dynamical systems via set invariance. A discussion on the strengths and limitations of each method demonstrates the capabilities of control barrier functions (CBFs) as a systematic tool for safety assurance in motion control. A safety filter module is also introduced as a tool to enforce safety. CBF constraints can be enforced as hard constraints in quadratic programming (QP) optimization, which rectifies the nominal control law based on the set of safe inputs.

We propose a multiple CBF scheme that enforces several safety constraints with high relative degrees. Using the multi-input multi-output (MIMO) feedback linearization technique, we derive conditions that ensure all control inputs contribute effectively to safety. This control structure is essential for challenging robotic applications requiring multiple safety criteria to be met simultaneously. To demonstrate the capabilities of our approach, we address reactive obstacle avoidance for a class of multi-body mobile robots, specifically tractor-trailer systems. The lack of fast response due to poor maneuverability makes reactive obstacle avoidance difficult for these systems. We develop a control structure based on a multiple CBFs scheme for a multi-steering tractor-trailer system to ensure a collision-free maneuver for both the tractor and trailer in the presence of several obstacles. Model

predictive control serves as the nominal tracking controller, and we validate the proposed strategy in several challenging scenarios.

Although the CBF method has demonstrated a great potential for ensuring safety, it is a model-based method and its effectiveness is closely tied to an accurate system model. In practice, model uncertainty compromises safety guarantees and may lead to conservative safety constraints, or conversely, allow the system to operate in unsafe regions. To address this, we explore developing safety-critical controllers that account for model uncertainty. Achieving this requires combining the theoretical guarantees of model-based methods with the adaptability of data-driven techniques. For this study, we selected Gaussian processes (GPs) which bring together required capabilities. It provides bounds on the posterior distribution, enabling theoretical analysis, and producing reliable approximations even with a low amount of training data, which is common in data-driven control. The proposed strategy mitigates the adverse effects of uncertainty on high-order CBFs (HOCBFs). A particular structure of the covariance function is designed that enables us to convert the chance constraints of HOCBFs into a second-order cone constraint, which results in a convex constrained optimization as a safety filter. A discussion on the feasibility of the resulting optimization is presented which provides the necessary and sufficient conditions for feasibility. In addition, we consider an alternative approach that uses matrix variate GP (MVGP) to approximate unknown system dynamics. A comparative analysis is presented which highlights the differences and similarities of both methods. The proposed strategy is validated on adaptive cruise control and active suspension systems, common applications in mobile robots.

This study next explores the safety of switching systems, focusing on cases where system stability is assured through control Lyapunov functions (CLFs) and CBFs are applied for safety. We show that the effect of uncertainty on the safety and stability constraint forms piecewise residuals for each switching surface. We introduce a batch multi-output Gaussian process (MOGP) framework to approximate these piecewise residuals, thereby mitigating the adverse effects of uncertainty. We show that by leveraging a specific covariance function, the chance constrained safety filter can be converted to a convex optimization, that is solvable in real-time. We analyze the feasibility of the resulting optimization and provide the necessary and sufficient conditions for feasibility. The effectiveness of the proposed strategy is validated through a simulation of a switching adaptive cruise control system.

Acknowledgements

I would like to express my gratitude to my supervisor, Prof. Jun Liu, for his unwavering support and guidance throughout my research journey. He taught me how to translate my ideas into rigorous mathematical formulations. Despite granting me the freedom to explore my research independently, his expertise, insightful feedback, and constructive suggestions were invaluable in shaping the direction of my work.

I extend my gratitude to my Master's study supervisor, Prof. Ali Khaki Sedigh, and my advisor, Dr. Alireza Fatehi. Their expertise and passion for research have always been a source of inspiration for me.

I would like to sincerely thank my examining committee members, Prof. Xinzhi Liu, Prof. Christopher Nielsen, my internal/external examiner Prof. Amir Khajepour, and my external examiner Dr. Xiangru Xu for taking the time to read my thesis and for their insightful comments and suggestions which have been invaluable in improving the quality of this work. I express my heartfelt appreciation to my friends at the Hybrid Systems Lab for fostering such a vibrant and inspiring environment.

I have been through difficult times during my PhD, but I never felt alone. To my wife, you stood by my side through every step, providing strength and motivation when I needed it most. Your unwavering love and support pushed me forward, and for that, I am forever grateful.

To my father, my mother, and my sister, thank you for the unconditional love and support you have always given me. The greatest challenge during my PhD journey was being far from you, but despite the distance, I could always feel the warmth of your encouragement. You have shaped me into the person I am today, I am carrying each and every moment we spent together with me.

Dedication

To Mom and Dad, who gave me the gift of dreams and the ability to realize them.

Table of Contents

Examining Committee ii

Author's Declaration iii

Abstract iv

Acknowledgements vi

Dedication vii

List of Figures xii

List of Tables xiii

List of Abbreviations xiv

1 Introduction 1

1.1 Motivation 1

1.2 Thesis Contributions 3

1.3 Thesis Organization 5

2 Literature Survey 7

2.1 Safety Filters 7

2.1.1 Set invariance . . . 8
2.1.2 HJ reachability safety filter . . . 8
2.1.3 Control barrier function safety filter . . . 10
2.1.4 Predictive safety filter . . . 11
2.1.5 Comparison of the safety filters . . . 12
2.2 Safety Control with Control Barrier Function . . . 14
2.2.1 Foundational results . . . 14
2.2.2 Principal results . . . 16
2.2.3 Data-driven CBFs . . . 19
2.2.4 Multiple CBFs . . . 20

3 Preliminaries **23**
3.1 System Dynamics . . . 23
3.1.1 Continuous-time system dynamics . . . 23
3.1.2 Discrete-time system dynamics . . . 24
3.1.3 Switching system . . . 25
3.2 Modeling of Mobile Robots . . . 25
3.2.1 Unicycle and bicycle model . . . 26
3.2.2 Modeling tractor-trailer robots . . . 28
3.3 Control Background . . . 33
3.3.1 Stability and control Lyapunov function . . . 33
3.3.2 Safety and control barrier function . . . 37
3.3.3 Uncertainty decomposition and robust safety verification . . . 41
3.3.4 Enforcing CBF and CLF . . . 43
3.3.5 Model predictive control . . . 44
3.4 Gaussian Process Regression . . . 46

4 Multiple Control Barrier Functions **48**
4.1 Introduction . . . 48
4.2 Control of Tractor-trailer Systems . . . 49
4.3 MIMO Feedback Linearization . . . 51
4.3.1 Input-output linearization . . . 53
4.3.2 Full state feedback linearization . . . 56
4.4 Theoretical Analysis of Multiple CBFs . . . 58
4.4.1 Quadratic programming optimization . . . 62
4.5 An Application to Autonomous Tractor-trailer . . . 63
4.5.1 Kinematic model of the tractor-trailer robot . . . 63
4.5.2 Construction of CBFs . . . 64
4.5.3 Controller design . . . 66
4.6 Simulation Results . . . 67
4.6.1 Obstacle avoidance in a 90° turning maneuver . . . 68
4.6.2 Obstacle avoidance in a cluttered workspace . . . 71
4.7 Chapter Summary . . . 73

5 Learning High-Order CBFs for Safety-Critical Control with GP **74**
5.1 Introduction . . . 74
5.2 Background and Preliminaries . . . 76
5.3 Impact of Model Uncertainty on HOCBFs . . . 77
5.4 GP-based HOCBF Design . . . 80
5.4.1 GP for uncertainty-aware HOCBF . . . 81
5.4.2 Confidence bounds for the estimation of the effect of uncertainty . . . 83
5.4.3 Feasibility analysis . . . 86
5.5 Bayesian Learning of System Dynamics . . . 88
5.5.1 Problem formulation . . . 88
5.5.2 MVGP-based HOCBF design . . . 89

5.5.3 Comparison of the SOCP-HOCBF and MVGP . . . 96
5.6 Simulation Results . . . 98
5.6.1 Adaptive cruise control with collision avoidance . . . 98
5.6.2 Active suspension system . . . 100
5.7 Chapter Summary . . . 101

6 Learning Piecewise Residuals of CBFs for Safety of Switching Systems 102
6.1 Introduction . . . 102
6.2 Problem Setup and Statement . . . 104
6.2.1 Control Lyapunov functions for switching systems . . . 104
6.2.2 Control barrier functions . . . 105
6.2.3 Impact of uncertainty on the switching systems . . . 106
6.3 Proposed Batch MOGP-based Design for Piecewise Residuals . . . 108
6.3.1 Single-output GP for real-valued functions . . . 108
6.3.2 Batch multi-output GP for piecewise residuals . . . 108
6.3.3 Confidence bounds for the estimation of the effect of uncertainty . . . 111
6.3.4 Feasibility analysis . . . 114
6.3.5 Data collection and model fitting . . . 116
6.4 Simulation Results . . . 117
6.5 Chapter Summary . . . 120

7 Conclusions and Future Works 121
7.1 Conclusion . . . 121
7.2 Future Works . . . 122

References 124

A Appendix 141
A.1 Tractor and Trailer's Positional Safety Constraint . . . 141
A.2 Discrete Linear Time-varying System . . . 142

List of Figures

2.1 Comparison of safety filter methods 13

4.1 A multi-steering tractor-tractor system. 65
4.2 SSTTR with ECBF in a 90° turning maneuver 69
4.3 The SSTTR system inputs. 70
4.4 MSTTR with multiple CBFs in a 90° turning maneuver 70
4.5 The MSTTR system inputs 71
4.6 MSTTR with multiple CBFs in a cluttered environment 72
4.7 The MSTTR system inputs in a cluttered environment simulation 72

5.1 Comparison of the adaptive cruise control state variables and control input in the presence of uncertainty 99
5.2 Active suspension system. 100
5.3 Comparison of the proposed method GP-based SOCP-HOCBF with nominal QP-HOCBF and true QP-HOCBF for active suspension system. 101

6.1 Comparison of the MOGP-SOCP, single GP-SOCP, nominal QP, and true design 118
6.2 Snapshots of the simulation to show the unsafe and safe distance between cars in different implementations 119

List of Tables

4.1 Maximum values for the MSTTR and SSTTR models 68

6.1 Parameters for the nominal and true system in $\mathcal{R}_1, \mathcal{R}_2$ 118

List of Abbreviations

AMR Autonomous Mobile Robots 10

CBF Control Barrier Function 3–5, 7, 8, 10–22, 33, 38–44, 48, 49, 51, 58, 60–63, 65–69, 71, 73–75, 88–90, 92–94, 96–98, 100, 102, 103, 105–107, 111, 113, 114, 116, 117, 119–123

CDF Cumulative Distribution Function 93, 94, 96

CLBF Control Lyapunov-Barrier Function 16

CLF Control Lyapunov Function 4, 5, 11, 15–17, 19, 33, 35–37, 43, 44, 98, 103–107, 111, 113, 114, 116, 117, 120, 122

CLF-CBF Control Lyapunov Function-Control Barrier Function 102, 103

ECBF Exponential Control Barrier Functions 41, 67–69, 71, 75, 76, 94, 95, 97

GP Gaussian Process 4, 19, 46, 74–76, 81–83, 86, 88–91, 96–99, 101–103, 108, 109, 111, 117, 119, 120, 122, 123

HJ Hamilton-Jacobi 5, 7–14

HOCBF High-Order Control Barrier Function 4, 60, 74–78, 80, 83, 86, 88, 96–99, 101, 122, 123

IQC Integral Quadratic Constraints 18

ISSf Input-to-State Safety 17–19, 42, 43

LQR Linear Quadratic Regulator 100

LTV Linear Time-Varying Model Predictive Control 63, 66–69, 71

MIMO Multi-Input Multi-Output 5, 51, 54, 57, 122

MOGP Multi-Output Gaussian Process 4, 103, 108–110, 114, 116, 119, 120

MPC Model Predictive Control 5, 11, 12, 44, 45, 49–51, 63, 66–69, 71

MSTTR Multi-Steering Tractor-Trailer Robot 28, 30, 32, 48–51, 63, 65, 66, 68, 69, 71, 73

MVG Matrix Variate Gaussian 90–92

MVGP Matrix Variate Gaussian Process 75, 89, 92, 94, 95, 97, 98

NHTSA National Highway Traffic Safety Administration 2

ODE Ordinary Differential Equation 23, 38

PDE Partial Differential Equation 9

PSF Predictive Safety Filter 7, 11–14

QP Quadratic Programming 4, 18, 21, 22, 43, 62, 63, 66, 71, 76, 99, 101, 102, 106, 116, 117, 119, 120, 122

RKHS Reproducing Kernel Hilbert Space 82, 83, 110, 111

RL Reinforcement Learning 20

SOCP Second-Order Cone Program 4, 76, 86, 87, 94, 97, 99, 101, 103, 112, 114–117, 119, 120, 123

SOGP Single-Output Gaussian Process 108

SPD Symmetric Positive Definite 87, 115

SSTTR Single-Steering Tractor-Trailer Robot 28–30, 32, 48, 49, 68, 69

TTR Tractor-Trailer Robot 28, 48–50, 71

Chapter 1

Introduction

1.1 Motivation

Automation is the application of technology to execute tasks with minimal or no human intervention. Although it has existed for centuries, the modern world has witnessed a remarkable rise in technological advancements in recent decades. Software advancements, such as significant progress in artificial intelligence, cloud computing, and big data analytics, have enhanced automation versatility, and productivity. On the hardware side, developments such as powerful processors, faster and higher-capacity memory, and improved sensor technology have facilitated the implementation of modern techniques. As a result, the use of automated technologies, which was once primarily limited to industrial and manufacturing settings, has now expanded to impact many different parts of everyday life. This includes areas such as household tasks, healthcare, transportation, and many other activities that people engage in on a regular basis.

A significant part of the rise in automation has emerged from the integration of robotics into operations of different levels of difficulty. Robotic vacuum cleaners were among the first robots designed for household use [59]. Fully autonomous floor scrubbing robots have been introduced to maximize cleaning productivity in crowded locations such as shopping malls, large retail stores, and airports [121]. In healthcare, medical robots perform a variety of tasks, with advanced surgical robots allowing surgeons to perform high-precision minimally invasive surgery, significantly reducing patient recovery time [65]. Drones are widely used in applications ranging from recreational activities to search and rescue missions. The deployment of autonomous vehicles has begun, with robotaxis offering rides to urban passengers after extensive public road tests. Additionally, self-driving truck companies

have launched several tests to achieve fully autonomous freight transportation. However, as robotic technologies become increasingly integrated into everyday life, they introduce significant safety concerns. For autonomous systems to be trusted by the public, they must address a fundamental question: Are we safe when we delegate tasks to these robots? Part of the robotics literature seeks to answer this important question. In particular, safety-critical control systems are developed to meet the crucial requirement of ensuring safety in robotics. Systems in this field are expected to perform tasks where any failure or malfunction could cause irreparable consequences, such as death or serious injury to people, or damage to expensive equipment.

There are several advantages to enhancing the safety of autonomous systems through safety-critical control. Air transportation is a good example of semi-autonomous airplanes that can safely transport people or goods over long distances in the shortest time. It should be noted that air transportation is the safest mode of travel. Statistics show that flying a specific distance is a hundred times safer than driving the same distance on roads [135]. Factors such as stringent safety regulations, advanced technology, redundancy in components and design, and a structured operating environment collectively ensure that air travel remains safe. In contrast, a statistical report from the National Highway Traffic Safety Administration (NHTSA) indicates that human errors account for 94% of fatal crashes in the United States [5]. This report highlights factors such as misjudgment, speeding, drunk driving, and phone use as common mistakes made by drivers. Self-driving vehicles have the potential to mitigate these shortcomings, provided they are designed within a framework that guarantees a high level of safety. Consequently, autonomous vehicles can make roads safer not only for drivers but also for pedestrians and cyclists.

As a result, there is growing interest in the deployment of autonomous vehicles. Achieving this goal requires the development of accurate control designs that meet performance-driven objectives while ensuring safety. This task becomes increasingly challenging when dealing with high levels of uncertainty. Ensuring that autonomous vehicles can operate reliably in unseen environments justifies the use of artificial intelligence for decision making together with model-based control for executing actions. Integrating the strengths of both paradigms is crucial for the successful deployment of autonomous vehicles in everyday transportation systems. The current trend leverages the capabilities of artificial intelligence to train black-box neural networks that can potentially outperform human drivers by processing data from various sensors at high frequencies. In particular, computer vision made significant progress in tasks such as perception and behavior prediction. While this approach provides satisfactory predictions for the vehicle's surrounding environment, it faces challenges when used to train an end-to-end control system. The primary issue is that safety-critical systems require transparency to formally guarantee safety. A learning-

based controller lacks this crucial feature. Furthermore, a safety-critical system must be interpretable to enable designers to trace back the source of any failure. Without these crucial attributes, deploying learning-based methods in safety-critical applications remains problematic.

The shortcomings of the current research, together with the high demand for a practical solution, motivate our research on safety-critical motion control in general and multi-body mobile robots in particular. We observe that all safety-critical systems require a notion of constraint satisfaction to systematically enforce safety.

1.2 Thesis Contributions

This thesis presents a theoretical approach to designing reliable and efficient safety-critical control methods by combining model-based formalism with the adaptability of data-driven control strategies. As guaranteeing system safety is one of the highest priorities of safety-critical control design, we tried to develop data-driven methods such that safety is mathematically guaranteed. In the following, we will outline the key contributions of this dissertation.

Multiple control barrier functions

We introduce a framework that systematically enables the design of multiple safety constraints for systems of high relative degrees. This approach generalizes the single Control Barrier Function (CBF) construction method to accommodate multiple CBFs of arbitrary relative degrees. The proposed safety-critical strategy eliminates the application limitations in complex, multi-body systems by ensuring safety for each robot segment. The contributions are summarized as follows.

- We prove that the intersection of all individual safe sets is forward invariance.
- This method establishes a framework such that all control inputs contribute effectively to safety assurance.
- We evaluate the effectiveness of the method by applying it to a tractor-trailer system, where it enabled reactive, collision-free obstacle avoidance for both tractor and trailer.

Data-driven high-order control barrier function

We develop a data-driven approach based on Gaussian Process (GP) to learn the impact of model uncertainty on high-order safety certificates. We then incorporate the uncertainty-aware high-order safety constraint with an arbitrary robust controller in a min-norm quadratic program. The contributions are summarized as follows.

- We characterize the impact of uncertainty on the High-Order Control Barrier Function (HOCBF) and prove that with a proper design of HOCBFs, the residual term in the high-order safety certificate remains control affine.
- We prove that by selecting a particular covariance structure in the GP model, the resulting min-norm Quadratic Programming (QP) with an uncertainty-aware constraint can be converted to a Second-Order Cone Program (SOCP).
- We analyze the feasibility of the resulting constrained optimization problem and determine the necessary and sufficient conditions for feasibility.
- An alternative approach is to fit a GP to the unknown system dynamics. A comparative analysis between our proposed method and this alternative is also presented.

GP-based high-order CBF design for switching systems

We introduce a safety-critical approach that includes switching systems by developing a batch Multi-Output Gaussian Process (MOGP) model that learns the piecewise residuals in each switching surface. Using a multi-output framework, the model simultaneously learns the residuals associated with the Control Lyapunov Function (CLF) and CBF, with batch processing facilitating efficient hyperparameter computation.

- We show that the switching system imposes piecewise residuals on CLF and CBF constraints.
- We develop a batch MOGP model to efficiently approximate residuals in each switching surface.
- We demonstrate that by selecting a particular form of the covariance function in the batch MOGP structure, the resulting min-norm controller with an uncertainty-aware chance constraint can be converted into an SOCP.
- We identify the necessary and sufficient conditions for the feasibility of the constrained optimization.

1.3 Thesis Organization

This thesis is organized to introduce, develop, and analyze the methodologies and contributions in safety-critical control. The structure is as follows:

In Chapter 2, we present a literature survey covering state-of-the-art methods in safety-critical control. The survey begins with the concept of set invariance, followed by an introduction to the safety filter module as a tool for enforcing safety. We discuss three main approaches, namely Hamilton-Jacobi (HJ) reachability, CBFs, and predictive filters which mathematically define safety criteria and enable their implementation. We provide a comparative study on these methods which highlights the strengths and limitations of each. This discussion is followed by a thorough survey on CBFs as the selected method for developing safety-critical control in this study. Finally, a brief overview of control strategies for tractor-trailer systems as a use case is presented.

Chapter 3 reviews key concepts that establish the theoretical foundation of this thesis results. We begin by stating the class of nonlinear dynamical systems that are used in this study, which is followed by the kinematic modeling of the multi-body mobile robots. A discussion on the controllability of these systems is also presented. The second part formalizes the control theories essential to developing safety-critical strategies, including the formal definitions of CBFs and CLFs, followed by a brief overview of model predictive control and Gaussian processes.

Chapter 4 presents the multiple CBFs framework for multi-body mobile robots. We derive the conditions to ensure that all control inputs effectively contribute to safety by using a Multi-Input Multi-Output (MIMO) feedback linearization technique. We also prove that the resulting safe set obtained by intersecting individual sets is forward invariance. Then, this method is combined with an Model Predictive Control (MPC) method to develop a reactive obstacle avoidance tracking controller for multi-body mobile robots.

In Chapter 5, we analyze the effect of uncertainty on the high-order safety constraints and propose a data-driven approach to approximate this effect. We show that our design results in a convex optimization problem that can be solved in real time. A discussion on the feasibility of the resulting optimization, which includes stating necessary and sufficient conditions for feasibility is presented. We explore solving this problem with an alternative approach which learns the system dynamics. We present a comparative analysis that highlights the similarities and differences of each approach.

Chapter 6 explores the safety in the area of switching systems in the presence of uncertainty. We show that the effect of uncertainty on these systems forms piece-wise residuals on stability and safety constraints. We develop a multi-output data-driven framework to

approximate the piece-wise residuals which leads to a convex optimization. A discussion on the feasibility of the resulting optimization is presented which states the necessary and sufficient conditions of feasibility.

Chapter 7 provides concluding remarks and identifies future research in the area of safety assurance in dynamical systems.

Chapter 2

Literature Survey

2.1 Safety Filters

Today, control theory includes well-established methods to efficiently meet multiple requirements, ranging from stability to specific performance criteria. However, modern control systems requirements go beyond stability criteria with respect to set points and reference trajectories and include safety verification, which justifies the employment of safety-critical methods. One important fact about the safety-critical systems is that applying an unsafe control input does not immediately lead to a violation of safety certificates (which is commonly represented by state constraints) but rather directs the system into a region of the state space from which violating safety is unavoidable in the future. The need to identify such behavior drives the development of safety filters, which are designed to be integrated with uncertified control strategies in a modular fashion. These filters systematically enforce safety by performing two complementary tasks: monitoring and intervention. The safety filter constantly monitors the intended control actions to ensure they meet safety specifications. If necessary, the filter can adjust or fully override the planned control inputs to maintain safety.

This paradigm often involves minimizing an objective function under safety constraints, a challenging problem in control. Over the last two decades, research on this topic has remained a hot spot. As a result, three main research directions, namely, HJ reachability, CBF, and Predictive Safety Filter (PSF) have evolved to address the problem. Although each method attempts to tackle safety verification from a specific point of view, they are all developed based on the set invariance theorem and belong to the family of model-based

methods. This section focuses on the most recent advances in safety filters. Then, we explain the reason that CBF was selected as the safety method in this thesis.

2.1.1 Set invariance

Before discussing the three (main) safety methods, it would be beneficial if we review the concept of set invariance which plays a pivotal role in developing the safety methods. Set invariance [25] is a fundamental concept in control theory and systems engineering, focusing on ensuring that the state of a dynamical system remains within a predefined set, known as an invariant set, for all future time. This is particularly important in safety-critical applications where it is essential to prevent the system from reaching unsafe states. A set is defined as (forward) invariant for a dynamical system if, for any initial condition in the set, the trajectory of the system also lies within the set for all time. This definition implies not only a strong requirement for the initial state to lie in the predefined set but also imposes a more restrictive constraint on the trajectory to remain in the set rather than the whole state space. Thus, the construction of the feedback controller to achieve set invariance usually impacts the overall performance of the system due to the use of a conservative predetermined set. Consequently, a set is called controlled invariant set [24] for a system, if for any initial condition in the set, there exists a piece-wise continuous control signal such that the resulting trajectory lies in the set for all time. We say a set is a maximal controlled invariant set in the state space if it contains all the initial conditions such that the previous conditions hold.

Given a controlled invariant set, we can propose various safety filter designs based on this definition. However, the size of the controlled invariant set impacts the performance obtained by applying the resulting control law. In the next sections, we will review three techniques that produce different feedback controllers and controlled invariant sets with varying levels of performance.

2.1.2 HJ reachability safety filter

In this safety filter design, HJ reachability is exploited to synthesize the maximal controlled invariant set in the state space. By maximizing the controlled invariant set, the resulting safety filter reduces the conservatism by its design. HJ reachability [18] is a constructive tool for obtaining optimal solutions for reachability problems based on HJ equations. The focus on a reachability problem in control is on the set of states that can be encountered by a trajectory of a dynamical system, which includes a vast majority of the problems useful

for system verification. A good example is reach-avoid problems that are concerned with reaching a goal region while avoiding obstacles [138].

Now, we shift our focus to the reachability problems that find the maximal controlled invariant set within a constrained state space (safe region). In this approach, we are concerned with state trajectories that are inevitably reaching the unsafe region (the complement of the constrained state space), regardless of the control effort. Collecting these trajectories forms a region from which violating the safety constraints is inevitable. On the other hand, the complement of this set leads to a set where safety violations can be avoided by an appropriate choice of control. Thus, this approach leads to the maximal controlled invariant set. This complementary relationship is formally defined in [22]. Based on this method, a value function can be obtained by defining a distance function to the set whose result for the interior points indicates how far they are from the boundary of the set. The negative value for the value function shows a safety violation. Therefore, the value function defines an optimal control problem that seeks to maximize it over the set of all feasible control inputs with the goal to ensure that it is non-negative which implies the satisfaction of safety constraints. Thus, the value function plays a vital role in the reachability-based safety filters in the sense that it can be used as a metric for safety verification with negative values indicating safety violation and larger positive values ensuring a more robust safety margin. In [104], it has been shown that we can construct the maximal controlled invariant set using the value function. Consequently, the boundary of this set (which is characterized by the zero superlevel set of the value function) separates the set of states in which violating the safety constraints is inevitable from the region where guaranteeing safety is feasible. In practice, an ε-superlevel set of the value function is used to introduce a level of conservatism to the design due to numerical error in approximating the value function. In the case that we have a differentiable value function, we can directly use it to synthesize a control policy in the safety filter framework. This policy should be activated for a short period of time when the trajectory reaches the boundary of the set to avoid chattering. This behavior is problematic in many practical use cases, where a sub-optimal solution is achieved at the boundary of the controlled invariant set.

Computing the value function is a crucial task in this approach as the maximal controlled invariant set is obtained by the ε-superlevel set of this function. In this framework, we can characterize the value function as the solution of a HJ Partial Differential Equation (PDE). This PDE can be obtained by the dynamic programming principle [57]. As the uniqueness of the solution of the HJ-PDE is not guaranteed, approximations such as using a discounted formulation of the HJ-PDE [7] or introducing finite-horizon value function for a sufficiently large horizon [56] has been proposed. In addition, numerical algorithms have been introduced for com based on the notion of viscosity solutions for PDE, which are

considered weak solutions to the problem [45]. A common approach in these algorithms is to create a state space grid and compute the value function, its gradient, and the Hamiltonian at each node. In consequence, applying this method to dynamical systems with high-dimensional state spaces leads to the curse of dimensionality. Although several methods have been introduced to address this challenge [36, 85], it has restricted the widespread application of this safety filter in practice. Nevertheless, the efficacy of this method has been evaluated in several real-world applications such as real-time motion planning [35] and navigation for Autonomous Mobile Robots (AMR) [17].

2.1.3 Control barrier function safety filter

CBFs established based on the fundamental concept of set invariance. Similar to HJ reachability, constructing a safe set is essential in this approach. However, a key advantage that sets CBFs apart from other safety filters is their ability to modify the desired control input smoothly as the system states approach the boundary of the safe set, rather than abruptly switching to a safe control policy at the boundary. This technique leverages the comparison theorem in its design, enabling the use of a wide range of tools developed for stabilization (in the sense of Lyapunov) to formally guarantee set invariance.

These features have contributed to the growing use of CBF-based safety filters in real-world applications, including robotic swarm [161], mobile robots [85], lane keeping [174], robotic grasping [43], quadrupedal robots [68], and autonomous vehicles [8]. To meet the demands of practical applications, several enhancements to the modified CBF design have been introduced in recent years, such as stochastic CBFs [41] and robust design [76]. Moreover, CBFs have been formulated for discrete-time [6] and sampled-data systems [43], enabling the integration of continuous-time systems with controllers implemented in a discrete-time framework.

Despite recent advances, CBF-based safety filters still face significant challenges. A key challenge is the constructive synthesis of CBFs and their validation across the state space (or a limited portion of it), particularly when inputs are bounded. While this task is manageable for relatively simple systems, such as maintaining a safe distance from the car ahead [3], avoiding obstacles in complex environments [1], or ensuring safe foot placement on variable footholds [68], it becomes analytically difficult for high-dimensional systems. Recent approaches have tackled this problem using numerical optimization methods like sum-of-squares [163, 42] and by learning CBFs from expert demonstrations or collected data [125, 48, 119, 100]. While these methods show promise, a well-established solution to this challenge has yet to be found.

2.1.4 Predictive safety filter

One of the recent approaches in designing safety filters is to exploit the methods that ensure safety from a predictive viewpoint, which includes PSFs [156], model predictive safety verification [155], and predictive shielding [19]. In PSF, the safety filter constantly checks the existence of a predictive backup safe control sequence from the current state to a controlled invariant set inside the state space. The goal is to find a trade-off between scalability and performance. Additionally, due to the similarity between this design and predictive control, it inherits the persistent feasibility and stability of the closed-loop system in MPC design. However, despite the similarities, safety filters solve a fundamentally different problem than the standard MPC design. In comparison to the MPC framework that tries to minimize an objective function to achieve stabilization or reference tracking (or performance criteria), safety filters provide a modular framework to verify that a trajectory to a safe controlled invariant set exists.

Consider a dynamical system at the initial state and a desired control input at the initial time. We assume that there exists a (conservative) control invariant terminal safe set for this system. The safety of the desired input is verified by searching for a dynamically feasible pair of state trajectory and control sequence that starts at the initial state with the initial desired control input and ends in the control invariant terminal set within a predefined prediction horizon. If such a trajectory exists, then we have the option to apply the desired input and bring the system to the safe terminal set within the finite prediction horizon. Thus, the system is proved to be safe in the sense that it has a predictive safe backup plan. In the case that a desired control input violates this condition, the closest input to the desired value is chosen to follow the same behavior. This safety filter is established based on the receding horizon fashion. Therefore, only the first element of the resulting safe control sequence will be applied to the system. This implies that the actual system trajectory is not required to follow the backup plan to the safe terminal set. On the contrary, it seeks to achieve the performance criteria applied by the desired control input (or the closest safe input) as long as it ensures returning to the controlled invariant set. Otherwise, the filter provides the closest control signal to the desired value that has this feature. Therefore, the backup control sequence is an alternative plan in the case that the desired input derives the system to the unsafe region.

The advantage of designing a safety filter in this predictive manner is that it eliminates the need for explicitly constructing a safe set for the system, as required in CBFs and HJ reachability. A trivial choice for the controlled invariant set can be the equilibrium point. However, a large terminal set leads to a large region in which the MPC problem is feasible even with a short prediction horizon [31]. As this choice is too conservative, CLFs can be

employed to expand the set, making it less restrictive by demonstrating the existence of a locally stabilizing controller for the equilibrium point [92]. Other methods include terminal set design using safe periodic orbits [153], adaptive enlargement of the terminal set [31], or exploiting discrete time CBFs [177].

Implementation of this method involves solving a predictive control problem online. In practice, we need to make sure that this optimization can be solved reliably in real time. This topic has been theoretically discussed in [123] and is the main challenge that restricts the application of MPC. Another practical challenge is applying this method in the presence of disturbances. The effect of disturbances can drive the system to a state where no safe control input can be computed. In this circumstance, the optimization problem becomes infeasible. A common approach to deal with infeasibility is to consider soft safety constraints by introducing slack variables. However, this may lead to a situation in which the backup control does not drive the system to the implicit safe set. There is ongoing research in MPC [115, 120] and PSF [64] by augmenting the PSF and CBF.

2.1.5 Comparison of the safety filters

We briefly introduced three main safety filter methodologies. All of the previously described methods use a form of forward invariant set to guarantee safety. The primary distinction between these methods lies in how each constructs the invariant set and how the safe input is applied to the system. In this section, we highlight the differences between the introduced methods and compare their capabilities.

Each method considers a subset of the state space that can be rendered forward invariant to satisfy the safety constraint. The most challenging approach is taken by HJ reachability, which requires finding an explicit representation of the maximal controlled invariant set. CBFs, on the other hand, do not necessarily aim for the maximal forward invariant set; instead, they use an inner approximation of the state space to verify the satisfaction of the CBF constraint. In PSFs, the controlled invariant set is obtained implicitly through the feasibility of the predictive optimization problem, which can ideally approximate the maximal invariant set with a sufficiently large prediction horizon. The mathematical characterization of the safe set is not available for all safety-critical problems in the case of CBFs and the resulting safe set appears to be a conservative approximation of the actual set which degrades the performance. Also, in the case of HJ reachability, the characterization of a controlled invariant set becomes problematic when we have a high dimensional system.

There are also different approaches to enforcing the safety input. In HJ reachability-

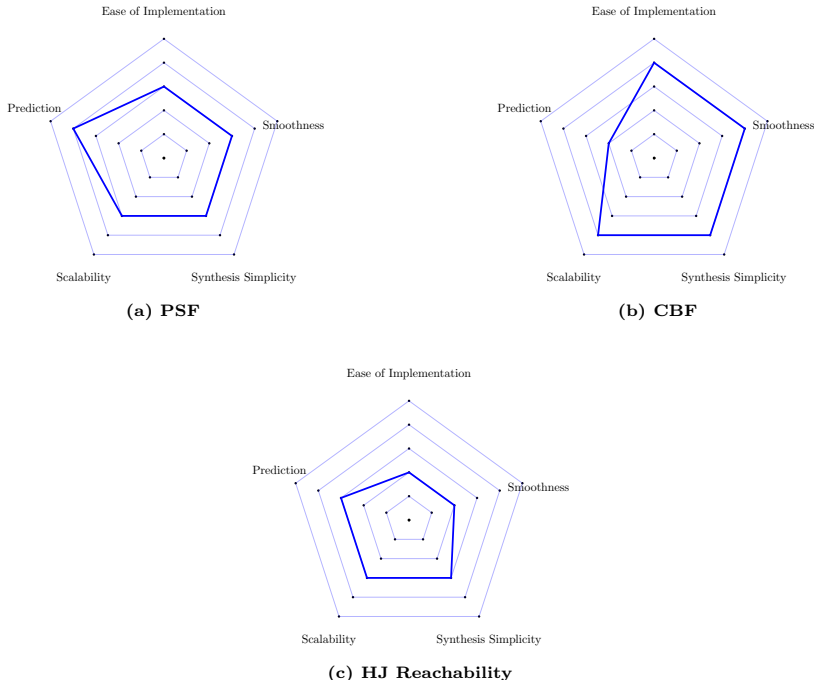


Figure 2.1: Comparison of safety filter methods

based safety filters, a value function is used to switch between the desired input and the safe input. Solving the HJ partial differential equation to find the value function often requires forming a grid on the state space. In consequence, this approach faces challenges with high dimensional systems. Even when the value function is efficiently computed, the method may result in constant switching at the boundary of the controlled invariant set, which leads to poor practical performance.

In contrast, CBF enforces the safety constraint through a min-norm optimization that minimizes the instantaneous deviation from the desired input. Since the derivative of the CBF preserves the control-affine structure of the dynamical system, it results in a quadratic programming problem. This fact contributes to the ease of implementation of this method. The other benefit is the smoothness of the filtering between the desired and safe input. However, constructing the CBF remains a challenge and is not always straightforward in all safety-critical scenarios.

Finally, PSF minimizes the deviation from the desired input over a prediction horizon. This predictive capability reduces the need to instantaneous switching to safe input, as long as the backup safe plan exists within the prediction horizon. However, the receding

horizon structure of PSFs requires solving the optimization problem reliably in real time, which introduces practical challenges, especially in high-dimensional systems.

In summary, selecting a safety filter is highly dependent on the specific requirements of the problem. For instance, HJ reachability provides a comprehensive safety guarantee but struggles with high-dimensional systems due to computational complexity. PSFs offer flexibility through prediction but require real-time optimization, which can be challenging for practical applications. On the other hand, CBFs have implementation and performance balance, but, they are applicable to problems that construction of a CBF is feasible. Based on the overview of each method, a comparison is presented in Figure 2.1.

2.2 Safety Control with Control Barrier Function

2.2.1 Foundational results

In the last two decades, safety verification of dynamical systems has gained a lot of attention in control systems. Reachability analysis was one of the earliest methods employed for safety verification. In principle, safety verification in the sense of reachability is defined as a method to show that starting at some initial conditions, a system cannot evolve to some unsafe region in the state space. One of the initial challenges of this approach was the explicit computation of exact or approximate reachable sets. To address this problem, exact computation of the reachable set was introduced for linear systems with certain eigenstructures and semi-algebraic initial sets [86, 14]. This was later extended to approximate reachable set for linear systems with almost arbitrary eigenstructures [150]. However, reachability analysis still could not reliably solve the problem of safety verification for nonlinear systems in general. Motivated by this trend, Prajna and Jadbabaie in [118] proposed a safety verification method that relied on the methodology that they called barrier certificates. Originally, barrier certificates were used in the context of nonlinear model validation [117]. In this paper, the idea was to eliminate the need to compute reachable sets, thus enabling safety verification of nonlinear systems.

They characterized the barrier certificate as "a function of state satisfying a set of inequalities on both the function itself and its time derivative along the flow of the system". Based on this characterization, they proposed that the zero positive level set of this function separates the safe states from the unsafe region in the state space. Thus, the existence of such a method acts like a certificate or proof of the system's safety. They made a connection between the capability of the barrier certificates in the verification of a property of the

system (reachability or safety in this case) without the need to compute the flow explicitly. Consequently, the function represented by the barrier certificate must be negative inside the safe region (strictly positive outside) and its derivative must be negative at the boundary of the set. Due to the constraint on the derivative of the function, the set of valid barrier certificates became non-convex. They addressed this issue at the expense of considering a more restrictive constraint on the derivative of the function, which was being negative for all states. The challenging step, similar to the Lyapunov methods, was finding the barrier certificate. The authors in [118] also formulated the conditions on the barrier certificates as the sum of squares problems and used semidefinite programming to numerically obtain the function.

The barrier certificate method introduced an alternative approach to certifying the safety of the closed-loop systems. Instead of explicitly computing the safe set, the zero positive level set of a valid barrier certificate implicitly defined the safe set. However, similar to the original Lyapunov theory, this method did not establish a direct connection between safety verification and control synthesis. Therefore, the only claim of this method was that the existence of a barrier certificate certified the safety of the system in the resulting set. Inspired by the CLF, Wieland and Allgöwer in [166] defined the concept of CBF. The structure of the proposed method is akin to the CLF, where the Lyapunov function is replaced by the barrier certificate. The proposed method had the capability of steering the system from a set of initial states to a terminal set without violating the safety defined by the barrier certificate. They considered control-affine systems and their key contribution was the construction of a safety feedback controller and barrier certificates for the corresponding closed-loop system. This method could successfully solve the problem of designing a safe feedback control law. However, due to its strict adherence to CLF structure, the proposed method lacked the flexibility to incorporate controllers that meet performance criteria, limiting its application.

The notion of CBFs was refined in [12]. Ames et al. introduced a modification to the definition provided by [118, 166], which does not require that sub-level sets of the CBF be invariant within the safe set. They also proposed a convex optimization approach to enforce the safe control input which enabled CBFs approach to be viewed as a safety filter method. The rationale behind these changes was to keep the safety verification capability of the original method while reducing its conservatism and adding flexibility to accommodate a desired controller focused on performance. The first aspect of this modification involved incorporating an extended class $\mathcal{K}$ function [82] on the right-hand side of the inequality for the derivative of the CBF. This modification allowed the system state to approach the boundary of the safe set, resulting in a less conservative design compared to the original barrier certificate approach. The second aspect took advantage

of the fact that for control-affine systems, the time derivative of the CBF is affine in the control input. This realization allowed the safety constraint to be incorporated into a convex optimization problem, enabling the filtering of a desired control input while ensuring safety. The similarity between the structure of CLF and CBF (both are affine in control) enabled them to propose a unified approach that could guarantee stability and safety simultaneously. Despite these modifications, the CBF introduced in this paper exhibited characteristics different from traditional barrier certificates. Motivated by the concept of barrier functions in optimization methods [27], the candidate CBF, known as reciprocal CBF, was required to be positive within the interior of the safe set and approach infinity at the boundary. This definition was later revised with the introduction of zeroing CBFs in [13]. The latter definition has since been widely adopted by researchers in the field of safety-critical control design, with the term CBF commonly referring to zeroing CBFs.

The idea of unifying the stabilization property of the CLF with the safety aspect of the CBF has been also proposed by Romdlony and Jayawardhana [1] in [127]. They acknowledged the features of CLFs for stabilizing the origin, such as (local) convexity and a global minimum at the origin, and observed that CBFs exhibit different characteristics, including (local) concavity with the level set of zero belonging to the safe domain. They concluded that, unlike CLFs, which can be combined while retaining their original properties, CBFs cannot be directly merged with CLFs. They proposed a novel unification approach, which led to the concept of Control Lyapunov-Barrier Function (CLBF), representing another important safety-critical control method.

2.2.2 Principal results

The CBF method showed great potential for addressing safety verification in control systems. However, several idealistic assumptions initially limited its application. In this section, we review research that complemented the initial CBF design, enhancing its relevance for real-world safety-critical problems.

A restrictive assumption was that the system must have relative degree one with respect to the CBF. Many robotic systems, which were the primary targets for safety verification, had relative degrees greater than one for common tasks such as obstacle avoidance or constraint satisfaction. As a result, researchers had to use a simplified version of the dynamical system, such as quadrotor and mobile robots, to be able to apply CBFs method. To address this limitation, a backstepping approach was introduced in [72], extending the

[1]Both references [12] and [127] are published in the same conference issue. However, the definition of CBF in [127] follows that of [166].

method to systems with a relative degree of two. Shortly after this improvement, Nguyen and Sreenath completely eliminated the assumption of relative degree one in [114] by introducing a novel approach which is called exponential CBFs. In this work, the authors proposed a method that employs input-output linearization and pole placement techniques to stabilize the barrier function to zero. Following up on this improvement, Xiao and Belta used a different approach to develop a more general definition of high-order CBFs in [168]. More recently, the assumptions of forward completeness and uniform relative degree have been relaxed for high-order CBFs [145].

These contributions open the door to work on the advanced control problems based on the CBFs method. Several important extensions to the notion of CBFs are developed in the subsequent years. Robust CBF was one of the first improvements that has been a topic of significant research interest. Xu et al. investigated the robustness of the CBFs under model perturbations using the input-to-state stability property of the safe sets in [173]. This approach is also followed by Kolathaya and Ames in [84], where the idea of Input-to-State Safety (ISSf) [128] is adopted to develop robust implementations of real-time safety-critical controllers in nonlinear systems. A novel approach for robust CBFs is introduced by Jankovic [76] that combined with the existing methodology for ISS CLFs to build safety-critical controllers for nonlinear systems. Taylor et al. proposed an adaptive version of the method in [148], which allows the system to adaptively ensure safety by keeping the system's state within a safe set, even in the presence of parametric model uncertainty. Santoyo et al. in [133] addressed the quantification of the probability that a system exists in a given safe region of the state space within finite time. This work extends the notion of CBFs to the domain of stochastic systems. Motivated by this research, Clark in [41] presented a framework for CBFs in stochastic systems in the presence of Gaussian process and measurement noise.

In recent years, several improvements have been introduced on the basis of the aforementioned works, making CBFs more versatile in practical safety-critical control design. In the following sections, we focus on the directions most aligned with the contributions of this thesis.

Robust CBFs: classical approach

CBF is a model-based approach, which requires the calculation of the time derivatives of the barrier function along the system's trajectory. This process assumes an accurate model of the system. However, in most practical settings, constructing high-fidelity models is challenging, and systems are often subject to external disturbances. Dynamical systems are often modeled by physical principles, but discrepancies between the actual system and

the nominal model often exist. These model mismatches can lead to inaccurate barrier conditions, compromising the system's safety. Therefore, addressing model mismatches and uncertainties in safety-critical control design is essential. Many researchers approach this problem from different perspectives. The robustness of the CBFs under model mismatch (e.g., $\dot{x} = f(x) + \Delta f(x) + g(x)u$) is first investigated in [173]. In this paper, the extent to which the forward invariance of a given safe set is robust in the presence of model errors is investigated. In this work, possible safety violation in terms of the size of the uncertainty is quantified using ISSf. However, the assumptions made regarding the errors were overly restrictive, limiting the method's ability to generalize to a wide range of dynamical systems with uncertainty. A more general form of disturbances to the system is considered in [76]. In this work, uncertainties are modeled as an unknown disturbance (e.g., $\dot{x} = f(x) + g(x)u + p(x)w$), which covers the previous model mismatch error as a special case. A modified version of the QP problem is introduced based on the input-to-state stability, and the safety is encoded to the system in the presence of disturbances. This approach was subsequently applied in a robot grasping scenario with bounded disturbances in [43]. While these works considered bounded additive disturbances, one common assumption is that the system states are measurable without any error. This assumption is relaxed in [63]. In addition to additive disturbance to the system dynamics, they considered bounded state-estimation errors and incorporated them into their robust design of CBFs method. In this work, an upper bound for the gradient of the barrier function is obtained based on the assumption that we have knowledge about the bounds of the disturbance and state estimation error. In [137], the effect of unmodeled dynamics and delays in CBFs method performance is addressed, where the true state of the system is partially observed (as the state of the unmodeled dynamics is not measured). A robust design is provided using the Integral Quadratic Constraints (IQC) approach.

The extensive research on robust CBFs highlights the importance of this topic. However, reviewing the aforementioned works, one can identify certain drawbacks to using classical methods. These methods often address specific classes of uncertainties and, in many cases, they cannot be generalized well to a wide range of real-world dynamical systems. Additionally, they typically rely on assumptions about the bounds of uncertainties to ensure safety. However, these bounds may not always hold across different environments and situations for a given system. Also, constructing a state (or input) dependent set of model errors is often significantly over-approximated, leading to excessively conservative safety verification. Next, we review the data-driven methods in robust CBFs, which aim to overcome these limitations.

2.2.3 Data-driven CBFs

The use of data-driven techniques for reliable safety verification with CBFs has become an active area of research. Leveraging the recent successes of learning-based control, researchers have incorporated machine learning into CBF methods to combine the strengths of both approaches. Most of the works in this field of research focus on reducing the model uncertainty as it negatively impacts the safe behavior of the system. Almost all of them are established on the robustness property of CBF-based safety verification known as ISSf.

A class of data-driven approaches uses probabilistic models of the residual learning error to ensure safety in the presence of uncertainty. In particular, these techniques leverage the distributional information of the residual model error and incorporate the learned residual model into the safety filter design process. The application of GPs to model uncertainty in safety-critical systems was first introduced in [20]. Using the GP approximation, the region of attraction of the system is computed in a discrete domain. In [164], the unknown additive dynamics of the system are represented by GPs, and a high probability confidence bound is obtained by the learned system dynamics. This bound allows the safe region to gradually expand as uncertainty decreases. In [33], it was noted that previous methods did not account for input-dependent model uncertainty. The authors addressed this by developing a specific GP kernel structure to overcome the limitation. In [83], the drift term and input gain of a control-affine system are assumed to be unknown, with the goal of learning the dynamics while ensuring system safety. This was achieved using a matrix-variate Gaussian distribution and Bayesian learning. Following this work, an extension to safe multi-agent interactions was developed in [38].

Another approach that addresses the impact of uncertainty on the evolution of CBFs, rather than focusing on learning the invariant set or unmodeled dynamics, is introduced in [147]. This method uses episodic learning to model the effect of uncertainty on the time derivative of the CBF, which helps reduce the curse of dimensionality. The approach has been extended to CBFs with higher relative degrees in [157]. Safety in the presence of measurement map uncertainties is addressed in [49], where the concept of measurement-robust CBFs was introduced. This work proposed a method to ensure safety even when the system's states are not directly observed. The same technique was applied in [44], providing empirical results using a Segway robot. In [40], reinforcement learning is employed to learn model uncertainties. In this work, the uncertainty in the dynamics of CBF and CLF constraints are learned to compensate for its effect on the safety and stability of the system. A different approach is taken in [96], where the agent learns a safe output feedback control law based on expert demonstrations. This method addresses the challenge of finding a CBF from a given dataset, as no systematic method exists for constructing a barrier function in

general.

In contrast to the aforementioned works, authors in [39] used CBFs to encode safe exploration to Reinforcement Learning (RL). This work is significant as it aims to move beyond simulation, enabling physical systems to learn safely in real-world environments. Specifically, they combine the safety verification capabilities of model-based CBFs with model-free RL algorithms. As a result, the RL-CBF method provides safety guarantees throughout the entire learning process while improving exploration efficiency. Although this method relies on a nominal dynamical model (based on first principles), safety is ensured from the beginning, even in the presence of uncertainty. Additionally, the concept of safe value functions was introduced in [108], using CBFs to derive optimal and safe value functions in RL. This work applies CBF-based safety verification from a different perspective, further expanding the utility of CBFs in reinforcement learning.

The works discussed above illustrate the ongoing research at the intersection of safety-critical systems using robust CBFs and learning-based methods. We also refer to [154] for a comprehensive survey on data-driven safety filters. This interest is partly driven by the development of powerful function approximators that enable the learning of accurate models for unknown phenomena, such as uncertainties. These methods offer a flexible framework for learning various types of uncertainties within a system and have shown promising results in real-world experiments. Recent advancements in RL also demonstrate significant potential for systems to learn while interacting with their environment, providing valuable opportunities to ensure safety across diverse settings. However, despite the advantages of data-driven approaches, they often lack formal theoretical guarantees—a limitation that classical robust CBF methods do not face. In this research, one of the objectives is to develop a data-driven approach for safety verification with formal guarantees.

2.2.4 Multiple CBFs

Ensuring the safety of real-world, safety-critical systems requires considering multiple safety criteria, which cannot always be achieved with a single CBF. Although multiple CBFs design provides a more comprehensive approach to safety verification, the concept has emerged in a few studies. In this section, we review the key contributions in this area and address the need for a more general framework that can handle multiple CBFs with arbitrary relative degrees.

Two commonly used approaches have been employed to handle multiple CBFs in safety filter design. One approach is to combine all CBFs into a single one. The first approach combines all CBFs into a single one. In [142, 116], multiple Lyapunov-like barrier functions

were incorporated into a set-theoretic framework for collision avoidance, using smooth minimum operators to combine CBFs and satisfy multiple objectives. Glotfelter et al. [66] proposed one of the first methods focusing on multiple CBFs by utilizing non-smooth operators, which guaranteed forward invariance of Boolean composition of safety objectives. The second approach involves enforcing multiple CBFs simultaneously in the safety filter design. In [152], forward invariance of all CBFs is achieved by using discontinuous control inputs. In [165], multiple CBFs are enforced for a motion planner, where a constraint relaxation scheme is introduced to prevent infeasibility.

Other works followed this methodology. In [122], multivariate CBFs are designed to control a manipulator. In this work, CBFs are used for time-varying constraint satisfaction in a QP scheme, and the uniqueness and Lipschitz continuity of the resulting control law is addressed. However, they provided theoretical results for CBFs with relative degree up to two, as at that time, the results of exponential CBFs for higher relative degrees were not published yet. Multi-agent systems are among applications that require multiple CBFs for safety verification. This idea is formulated in [159], where a safe swarm of a group of mobile robots is considered. The pair-wise distance between robots is considered as the safety constraint to avoid collision between agents. All the resulting barrier certificates are of relative degree one. The efficiency of the method has been shown by experimental results. However, the multiple CBFs scheme does not give theoretical guarantees for the existence of a common solution when we have a multi-objective problem. This issue is addressed in [160], where multiple CBFs are combined in a single CBF using Boolean compositions through product and sum of CBFs. In [66], it has been claimed that the robustness properties of the ZCBFs are removed by using Boolean compositions of barrier functions in [160]. Thus, they used min and max operators of multiple components barrier function to address this issue. This leads to non-smooth barrier function analysis for the first time in this work. The multiple CBFs have also been addressed in the context of robust CBFs in [70]. An active set invariance method is used to compensate for the effect of disturbance and parametric errors. However, the results are limited to barrier functions of relative degree one. The authors in [43] developed a robust safety-critical control design based on multiple CBFs scheme for robot grasping. The procedure of a robot grasping scenario is defined based on several constraints which are satisfied by using robust CBFs. Multiple safety constraints with relative degree one are used in some applications such as obstacle avoidance in manipulator [88], robot grasping [58], and trajectory planning under temporal constraints [62].

The idea of using multiple CBFs is an effective way of safety verification in dynamic systems. However, it brings up some challenges too. CBFs are generally enforced to the system by solving QP optimization problem in a minimally invasive fashion. Subsequently,

each CBF imposes a hard constraint on the optimization problem. Thus, feasibility analysis of the resulting convex optimization problem in the presence of several hard constraints is essential. The mentioned works did not address this problem and assumed that the problem was feasible (e.g., an optimal control law exists that satisfies the constraints). This issue is one of the most recent topics in the field of safety verification with CBFs.

In [172], the notion of control-sharing barrier functions is introduced, which provides the necessary and sufficient conditions for feasibility. However, the results are limited to the systems with scalar input and barrier functions with relative degree one. In [177], the low decay rate of CBF constraint is introduced as the reason behind potential infeasibility in QP optimization with CBFs. They proposed an optimal-decay CBF scheme that ensures the point-wise feasibility of the problem. In the most recent publication on this topic, sufficient conditions for the feasibility of optimal control problems with CBFs are proposed in [171]. In this method, the sufficient conditions are represented by a single constraint that is enforced on the QP optimization problem by a CBF. This so-called feasibility constraint makes the problem constraints compatible and guarantees feasibility. Based on the sign of the Lie derivative of the barrier function with respect to the input-gain dynamics, different cases are analyzed, and sufficient conditions are obtained for arbitrarily high relative degree CBFs.

As mentioned above, using multiple CBFs scheme is inevitable in many dynamical systems to guarantee safety. However, the existing methods consider CBFs with relative degree one or two. In Chapter 4, we will explore a solution to this problem and highlight the importance of such a safety-critical method in the control design of tractor-trailer systems. The next section provides a summary of related works on control design for tractor-trailer systems.

Chapter 3

Preliminaries

3.1 System Dynamics

In this section, we present the foundational concepts of system dynamics, which are essential for understanding how systems operate over time. System dynamics provides the mathematical framework for modeling, analyzing, and controlling robotic systems, forming the backbone of the analysis and design methodologies employed in this research. As the focus of this thesis is on designing safety-critical control methods for mobile robots, particularly tractor-trailer systems, we will mainly introduce control-affine system dynamics. Then, we formalize the equivalent discrete-time control-affine system. Finally, we will formally define switching systems which cover a wide range of real-world dynamical robots.

3.1.1 Continuous-time system dynamics

The vast majority of dynamical systems are naturally defined in continuous-time, reflecting the continuous nature of physical processes. We define state vector as the smallest set of variables that fully characterize system's current condition and determine its future behavior, given the system's dynamics and inputs. Control inputs, are variables that can be adjusted to manipulate the system's evolution over time. Dynamical systems are generally modeled by fundamental physical laws that govern the system. Applying the first principles usually leads to a set of Ordinary Differential Equation (ODE), which describe the dynamic behavior of the system in terms of the rates of change of the state vector with respect to time. Thus, a mathematical representation of a dynamic system can be

expressed in the general form:

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x}, \boldsymbol{u}),$$
$$\boldsymbol{y} = h(\boldsymbol{x}, \boldsymbol{u}).$$

This representation makes control synthesis analytically intractable. It is important to note that most robotic systems are modeled using Lagrangian equations of motion, which result in vector field that is affine in control vector and does not depend explicitly on time. These types of systems are generally known as time-invariant control-affine dynamics and are widely used to describe robotic systems. Consider the nonlinear control-affine system:

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

where $\boldsymbol{x}$ is the state vector which belongs to the state space manifold $\mathcal{X} \subseteq \mathbb{R}^n$. The input vector $\boldsymbol{u} = \begin{bmatrix} u_1 & \dots & u_m \end{bmatrix}^\top$ is constrained to lie in $\mathcal{U} \subseteq \mathbb{R}^m$. Input functions $u_j(\cdot), j = 1, \dots, m$, satisfying this constraint is called admissible input functions or control. Both $\mathcal{X}$ and $\mathcal{U}$ are compact and convex sets. The vector field $f \colon \mathcal{X} \to \mathbb{R}^n$ is referred to as drift vector field and each column of $g \colon \mathcal{X} \to \mathbb{R}^{n \times m}$ is referred to as input vector field. f and g are locally Lipschitz in $\boldsymbol{x}$. Given a locally Lipschitz continuous state feedback control $\pi \colon \mathbb{R}^n \to \mathbb{R}^m$, the closed-loop system is:

$$\dot{\boldsymbol{x}} = F_{cl}(\boldsymbol{x}) \triangleq f(\boldsymbol{x}) + g(\boldsymbol{x})\pi(\boldsymbol{x}), \tag{3.2}$$

which is locally Lipschitz continuous. Then, for any initial state $\boldsymbol{x}(0) = \boldsymbol{x}_0$, the system (3.2) has a unique solution $\boldsymbol{x}(t)$ defined on a maximal interval of existence $I(\boldsymbol{x}_0) = [0, I_{max})$. In this thesis, we consider forward complete closed-loop systems, thus $I_{max} = \infty$.

3.1.2 Discrete-time system dynamics

Discrete-time systems describe either inherently discrete systems (e.g., the balance of a bank savings account at the k-th month) or the equivalent transformation of a continuous-time system into the discrete domain. The latter is widely used in control systems, as controllers are almost always implemented using microprocessors, which can only handle computations at discrete time intervals. Subsequently, states, inputs and outputs of a discrete-time system are a sequence whose elements are defined only at particular instances $t_k \in \mathbb{R}_+, k \in \mathbb{Z}_+$ in time. Consider the discrete-time control-affine system defined by difference equations:

$$\boldsymbol{x}_{k+1} = f(\boldsymbol{x}_k) + g(\boldsymbol{x}_k)\boldsymbol{u}_k, \tag{3.3}$$

where $\boldsymbol{x}_k \in \mathcal{X} \subseteq \mathbb{R}^n$ is the state at time step t_k, $\boldsymbol{u}_k \in \mathcal{U} \subseteq \mathbb{R}^m$ is the input at t_k under the assumption of a constant $\boldsymbol{u}(t)$ during a sampling interval. The sets and state-transition functions are defined as before.

3.1.3 Switching system

Consider the control-affine switching system of the form

$$\dot{\boldsymbol{x}} = \sum_{r=1}^{R} \delta_r \left(f_r(\boldsymbol{x}) + g_r(\boldsymbol{x})\boldsymbol{u} \right), \tag{3.4}$$
$$\delta_r = \begin{cases} 1 & \text{if} \quad \boldsymbol{x} \in \mathcal{R}_r, \\ 0 & \text{otherwise}, \end{cases}$$

where $\boldsymbol{x} \in \mathcal{X} \subseteq \mathbb{R}^n$ is the state, and $\boldsymbol{u} \in \mathbb{R}^m$ is the input, and $\mathcal{X}$ is a compact and convex set. The function $\delta_r \colon \mathcal{X} \to \{0, 1\}, r \in \mathcal{I}$ is a state-dependent switching signal with the finite index set $\mathcal{I} = \{1, \ldots, R\}$ which indicates the index of the active switching surface $\mathcal{R}_r \subset \mathcal{X}$. The vector fields $f_r \colon \mathbb{R}^n \to \mathbb{R}^n$ and $g_r \colon \mathbb{R}^n \to \mathbb{R}^{m \times n}, r \in \mathcal{I}$, are assumed to be locally Lipschitz. Let $\pi \colon \mathbb{R}^n \to \mathbb{R}^m$ be a locally Lipschitz continuous state feedback control law. We assume that the impulse effects are absent, i.e., the reset map is the identity. Furthermore, we assume that the closed-loop system

$$\dot{\boldsymbol{x}} = F_{cl}(\boldsymbol{x}) \triangleq \sum_{r=1}^{R} \delta_r \left(f_r(\boldsymbol{x}) + g_r(\boldsymbol{x})\pi(\boldsymbol{x}) \right), \tag{3.5}$$

satisfies Caratheodory condition [71]. Then, for any initial state $\boldsymbol{x}(t_0) = \boldsymbol{x}_0$, system (3.5) admits a solution $\boldsymbol{x}(t)$ in the sense of Caratheodory defined on a maximal interval of existence $I(\boldsymbol{x}_0) = [t_0, I_{max})$. We consider complete solutions, thus $I_{max} = \infty$.

We assume that the state space is partitioned into R non-overlapping regions labeled as $\mathcal{R}_r \subset \mathcal{X}$, such that they cover the whole state space, i.e., $\cup_{r=1}^{R} \mathcal{R}_r = \mathcal{X}$, and $\mathcal{R}_i \cap \mathcal{R}_j = \emptyset$ for all $i, j \in \{1, \ldots, R\}, i \neq j$. This assumption requires system (3.4) to be well-posed, i.e., for all $\boldsymbol{x} \in \mathcal{X}$ there exists only one active index $r \in \mathcal{I}$ satisfying the membership condition.

3.2 Modeling of Mobile Robots

The first step in controlling a mobile robot is to develop a mathematical model. One common method for modeling mechanical systems involves using kinematic equations, which

describe the motion of a robot without considering forces. The kinematic model is widely used to describe the low-speed behavior of mobile robots and is primarily based on non-integrable rolling constraints, also known as nonholonomic constraints, which involve the derivatives of the configuration variables. These constraints are defined in the tangent space at each configuration, specifying the allowable velocities of the system. They cannot be eliminated by defining a more restricted configuration manifold.

For mobile robots, common kinematic models include car-like, differential drive, and omnidirectional models. The state of the robot is typically defined by its position and orientation. We start by obtaining the car-like model (without a trailer), and the same approach will be used to model tractor-trailer systems as multi-body mobile robots. A multi-body mobile robot consists of a sequence of two or several hinged bodies moving on flat ground. The first body in the sequence is called the tractor and the other bodies are called the trailers. Models which are derived in this section will be used in the next chapter as a benchmark for the proposed safety-critical control method.

3.2.1 Unicycle and bicycle model

Consider a single wheel moving along a flat surface while keeping its body vertical. This system is commonly known as a unicycle. Its configuration can be defined by a vector $\boldsymbol{x} \in \mathbb{R}^3$ consisting of three generalized coordinates: the positional coordinates x, y of the point of contact with the ground in a fixed frame, and the angle θ which denotes the wheel's orientation relative to the x-axis. The linear velocity of the wheel's center is restricted to lie within the plane of the wheel's body (zero lateral velocity). Specifically, they must satisfy

$$\begin{bmatrix} -\sin\theta & \cos\theta & 0 \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = 0, \tag{3.6}$$

which is linear in the generalized velocities of the form $C(\boldsymbol{x})\dot{\boldsymbol{x}} = 0$, which is called a Pfaffian constraint. Consequently, all admissible generalized velocities are contained in the null space of constraint matrix $C(\boldsymbol{x})$, which yields

$$\dot{\boldsymbol{x}} = \begin{bmatrix} \cos\theta \\ \sin\theta \\ 0 \end{bmatrix} v + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \omega, \tag{3.7}$$

where v and ω are linear and angular velocity of the wheel, respectively. This choice of the basis for the null space of the matrix $C(\boldsymbol{x})$ is not unique and other choices lead to

different representations of the system, in which the input components may have no direct relationship with the actual control inputs which are forces and torques. Thus, equation (3.6) is called the kinematic model of the unicycle.

Now, we consider a car-like robot with the assumption that two wheels on each axle collapse into a single wheel at the midpoint of the axle. The front wheel is steerable while the rear wheel orientation is fixed. The vector of the generalized coordinates is $\boldsymbol{x} = \begin{bmatrix} x & y & \theta & \delta \end{bmatrix}^\top \in \mathbb{R}^4$, where x, y are the Cartesian coordinates of the rear wheel and θ is the orientation of the car body with respect to the x-axis and δ represents the steering angle of the front wheel. Each wheel imposes a nonholonomic constraint on the system.

$$\begin{aligned} \dot{x}_f \sin(\theta + \delta) + \dot{y}_f \cos(\theta + \delta) &= 0, \\ \dot{x} \sin\theta - \dot{y} \cos\theta &= 0, \end{aligned}$$

where x_f and y_f are the Cartesian coordinates of the front wheel and have the following relationship with the car's reference point.

$$\begin{aligned} x_f &= x + \ell \cos\theta, \\ y_f &= y + \ell \sin\theta. \end{aligned}$$

By substituting this formula the first constraint becomes

$$\dot{x} \sin(\theta + \delta) - \dot{y} \cos(\theta + \delta) - \ell\dot{\theta} \cos\delta = 0.$$

Thus, the constraint matrix is

$$C(\boldsymbol{x}) = \begin{bmatrix} \sin(\theta + \delta) & -\cos(\theta + \delta) & -\ell \cos\delta & 0 \\ \sin\theta & -\cos\theta & 0 & 0 \end{bmatrix}, \tag{3.8}$$

which has rank two for all $\boldsymbol{x} \in \mathbb{R}^4$. Thus, for the car-like robot with rear-wheel driving, the kinematic model is obtained as

$$\dot{\boldsymbol{x}} = \begin{bmatrix} \cos\theta \\ \sin\theta \\ \frac{\tan\delta}{\ell} \\ 0 \end{bmatrix} v + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \omega, \tag{3.9}$$

where v, and ω are the driving and steering velocities, respectively. The model has a singularity at $\delta = \pm\frac{\pi}{2}$, which corresponds to the case that the front wheel is normal to the longitudinal axis of the body. However, the importance of the singularity is limited as the steering angle is restricted in most practical cases. This kinematic model is widely known as the bicycle (car-like) model. It will be our reference in deriving the kinematic model of the multi-body robots.

3.2.2 Modeling tractor-trailer robots

In the following, we derive a kinematic model of the Tractor-Trailer Robot (TTR) and then analyze the fundamental properties of the corresponding system from a control viewpoint. Based on the trailer articulation structure, TTRs are divided into Single-Steering Tractor-Trailer Robot (SSTTR) and Multi-Steering Tractor-Trailer Robot (MSTTR), respectively. Note that we derive the kinematic model for a tractor towing one trailer, but the same approach can be used to extend the model to the N-trailer system.

Single-steering tractor-trailer robots

In an SSTTR, the tractor is fully specified in the Cartesian plane by its reference point located on the midpoint of its rear axle. Let $\boldsymbol{p} = [x_1, y_1]^\top$, be the coordinates of the reference point. The angles θ_1 and θ_2 denote the orientation of the tractor and trailer, respectively. SSTTR has a steerable front wheel and its rear wheel is fixed. The steering angle of the front wheel is denoted by δ_1. The generalized coordinates of the system denotes by $\boldsymbol{x} = [x_1, y_1, \theta_1, \theta_2, \delta_1]^\top \in \mathbb{R}^5$.

We start by using the bicycle model for the tractor. The same approach leads to the nonholonomic constraints of the form $C(\boldsymbol{x})\dot{\boldsymbol{x}} = 0$ for SSTTR, with the constraint matrix

$$C(\boldsymbol{x}) = \begin{bmatrix} \sin(\theta_1 + \delta_1) & -\cos(\theta_1 + \delta_1) & -\ell_1 \cos\delta_1 & 0 & 0 \\ \sin\theta_1 & -\cos\theta_1 & 0 & 0 & 0 \end{bmatrix} \tag{3.10}$$

Consider matrix $S(\boldsymbol{x})$ that spans the null space of $C(\boldsymbol{x})$ as

$$S(\boldsymbol{x}) = \begin{bmatrix} \cos\theta_1 & 0 \\ \sin\theta_1 & 0 \\ \frac{\tan\delta_1}{\ell_1} & 0 \\ \frac{\sin(\theta_1-\theta_2)}{\ell_2} & 0 \\ 0 & 1 \end{bmatrix}.$$

Based on this representation, the kinematic model of the SSTTR can be obtained as

$$\dot{\boldsymbol{x}} = \underbrace{\begin{bmatrix} \cos\theta_1 \\ \sin\theta_1 \\ \frac{\tan\delta_1}{\ell_1} \\ \frac{\sin(\theta_1-\theta_2)}{\ell_2} \\ 0 \end{bmatrix}}_{g_1(\boldsymbol{x})} v + \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}}_{g_2(\boldsymbol{x})} \omega_1, \tag{3.11}$$

where v and ω_1 are the linear and steering velocity, respectively.

The kinematic model (3.11) is a driftless system (i.e., $f(\boldsymbol{x})$ is zero vector), thus any state $\boldsymbol{x}_e \in \mathcal{X}$ represents an equilibrium point under zero input. Linearization around $\boldsymbol{x}_e$ yields $\dot{\boldsymbol{x}} = g_1(\boldsymbol{x}_e)v + g_2(\boldsymbol{x}_e)\omega_1$, which fails to pass the controllability rank condition

$$\operatorname{rank}\left(\begin{bmatrix} B & AB & A^2B & A^3B & A^4B \end{bmatrix}\right) = 5, \tag{3.12}$$

since A matrix is zero and the number of control inputs is less than the number of states (i.e., $m < n$). This analysis shows that the linearized system of SSTTR is not locally controllable at a point. Thus, we cannot infer the local controllability of this system based on the linearized system. It is important to notice that a nonlinear system may be controllable, even if its linearized system is not. Intuitively, we know that a car-like robot (and tractor-trailer robots) can be steered to any configuration in an obstacle-free state space. In particular, the nonlinear controllability of driftless systems can be investigated by testing the Lie algebra rank condition [73]:

$$\operatorname{rank}\left(\begin{bmatrix} g_1 & g_2 & [g_1,g_2] & [g_1,[g_1,g_2]] & [g_2,[g_1,g_2]] & \dots \end{bmatrix}\right) = 5, \tag{3.13}$$

which based on Chow's theorem [134] shows that the system can reach nearby points in arbitrary small times and stay near a given state $\boldsymbol{x}_e$, which is widely known as small-time local controllability. Note that (3.13) is a local condition. If the rank condition holds everywhere in the state space, then the robot is said to be controllable.

For the SSTTR system (3.11), the first three Lie brackets[1] are computed as

$$[g_1,g_2] = \begin{bmatrix} 0 \\ 0 \\ -\frac{\sec(\delta_1)^2}{\ell_1} \\ 0 \\ 0 \end{bmatrix}, [g_1,[g_1,g_2]] = \begin{bmatrix} -\frac{\sin(\theta_1)(\sec(\delta_1)^2)}{\ell_1} \\ \frac{\cos(\theta_1)(\sec(\delta_1)^2)}{\ell_1} \\ 0 \\ \frac{\cos(\theta_1-\theta_2)(\sec(\delta_1)^2)}{\ell_1\ell_2} \\ 0 \end{bmatrix}, [g_2,[g_1,g_2]] = \begin{bmatrix} 0 \\ 0 \\ -\frac{2\tan(\delta_1)(\sec(\delta_1)^2)}{\ell_1} \\ 0 \\ 0 \end{bmatrix}.$$

Now, we can easily verify that away from the model singularity $\delta_1 = \pm\frac{\pi}{2}$, the Lie algebra rank condition (3.13) for system (3.11). The fact that the linearized system does not pass the controllability rank condition poses the limitation that the exponential stability in the

[1] Let f, g be two vector fields in $\mathbb{R}^n$. The Lie bracket of f and g is a third vector field defined by

$$[f,g] = \frac{\partial g}{\partial \boldsymbol{x}} f - \frac{\partial f}{\partial \boldsymbol{x}} g.$$

sense of Lyapunov cannot be obtained by smooth feedback [139]. Also, based on Brocket's theorem [29], as (3.11) is a driftless regular system (i.e., g_1 and g_2 are linearly independent vector fields), a necessary condition for smooth stability is having the same number of input and states, which is not the case for SSTTR. Thus, a time-varying control law must be designed for point stabilization. We also refer to [90] for proof of controllability of a tractor towing N trailers.

Next, we study the controllability analysis of a trajectory. In this case, the linearized model of (3.13) around a desired reference trajectory $\boldsymbol{x}_d = \begin{bmatrix} x_d & y_d & \theta_{1d} & \theta_{2d} & \delta_{1d} \end{bmatrix}^\top$, which satisfies the nonholonomic constraints (3.10). Such a trajectory, as well as the desired control input v_d, ω_{1d} always exists for SSTTR as it is a differentially flat system [93].

Defining $\tilde{\boldsymbol{x}} = \boldsymbol{x} - \boldsymbol{x}_d$ and $\boldsymbol{u}_d = \boldsymbol{u} - \boldsymbol{u}_d$, the linearized system around the reference trajectory is obtained as

$$\dot{\tilde{\boldsymbol{x}}} = A(t)\tilde{\boldsymbol{x}} + B(t)\tilde{\boldsymbol{u}},$$

where

$$A(t) = \sum_{i=1}^{2} u_{di}(t) \frac{\partial g_i}{\partial \boldsymbol{x}}\bigg|_{\boldsymbol{x}=\boldsymbol{x}_d(t)}, \quad B(t) = G\left(\boldsymbol{x}_d(t)\right),$$

where G is a matrix whose columns are g_1 and g_2. Computing the system matrices yields

$$A = \begin{bmatrix} 0 & 0 & -v_d \sin(\theta_{1d}) & 0 & 0 \\ 0 & 0 & v_d \cos(\theta_{1d}) & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{v_d}{\ell_1}\sec(\delta_1)^2 \\ 0 & 0 & \frac{v_d}{\ell_2}\cos(\theta_{1d}-\theta_{2d}) & -\frac{v_d}{\ell_2}\cos(\theta_{1d}-\theta_{2d}) & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}, B = \begin{bmatrix} \cos(\theta_{1d}) & 0 \\ \sin(\theta_{1d}) & 0 \\ \frac{\tan\delta_{1d}}{\ell_1} & 0 \\ \frac{\sin(\theta_{1d}-\theta_{2d})}{\ell_2} & 0 \\ 0 & 1 \end{bmatrix}.$$

The linearized system represents a time-varying system. Consequently, controllability analysis for this system is more involved and requires nonsingularity of the controllability Gramian [78]. We focus on the special case of a linear reference trajectory with constant velocity, which produces time-invariant system matrices. Thus, $u_{d1} = v_d$ and also we have $\theta_{1d} = \theta_{2d} = \theta_0$, and $\delta_1 = 0$. Then, the controllability rank condition (3.12) is satisfied, except for the case that $\theta_0 = \pm\frac{\pi}{2}$, or $v_d = 0$ (with zero velocity the trajectory would collapse to a point). In [91], it has been shown that the SSTTR system can be locally stabilized for arbitrary reference trajectory using linear feedback.

Multi-steering tractor-trailer robots

In an MSTTR, the trailer has also a steerable wheel, which enhances the maneuverability of the robot. The reference point is the midpoint of the tractor's rear axle as before and

the generalized coordinates are obtained by $\boldsymbol{x} = \begin{bmatrix} x_1 & y_1 & \theta_1 & \psi & \delta_1 & \delta_2 \end{bmatrix}^\top \in \mathbb{R}^6$, where we substituted θ_1 with ψ, which is the relative angle between two vehicle segments. Also, δ_2 is added as the steering angle of the trailer's wheel. The matrix of nonholonomic constraints is

$$C(\boldsymbol{x}) = \begin{bmatrix} \sin(\theta_1 + \delta_1) & -\cos(\theta_1 + \delta_1) & -\ell_1 \cos\delta_1 & 0 & 0 & 0 \\ \sin\theta_1 & -\cos\theta_1 & 0 & 0 & 0 & 0 \\ \sin(\theta_1 - \psi - \delta_2) & -\cos(\theta_1 - \psi - \delta_2) & \ell_2 \cos\delta_2 & -\ell_2 \cos\delta_2 & 0 & 0 \end{bmatrix}. \tag{3.14}$$

Given $S(\boldsymbol{x})$ as the null space of $C(\boldsymbol{x})$, we can write $\dot{\boldsymbol{x}} = S(\boldsymbol{x})\nu$, where the velocity vector $\boldsymbol{\nu} = \begin{bmatrix} v & \omega_1 & \omega_2 \end{bmatrix}^\top$. In this vector of the generalized velocity vector, v is the linear velocity and ω_1 and ω_2 are the turning rate of the tractor's front wheel and trailer's rear wheel, respectively. The matrix $S(\boldsymbol{x})$ can be written as

$$S(\boldsymbol{x}) = \begin{bmatrix} \cos\theta_1 & 0 & 0 \\ \sin\theta_1 & 0 & 0 \\ \frac{\tan\delta_1}{\ell_1} & 0 & 0 \\ S_{41} & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

where we have

$$S_{41} = \frac{\tan\delta_1}{\ell_1} - \frac{\tan\delta_2 \cos\psi + \sin\psi}{\ell_2}.$$

The resulting kinematic model is given by

$$\dot{\boldsymbol{x}} = \underbrace{\begin{bmatrix} \cos\theta_1 \\ \sin\theta_1 \\ \frac{\tan\delta_1}{\ell_1} \\ S_{41} \\ 0 \\ 0 \end{bmatrix}}_{g_1(\boldsymbol{x})} v + \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}}_{g_2(\boldsymbol{x})} \omega_1 + \underbrace{\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}}_{g_3(\boldsymbol{x})} \omega_2. \tag{3.15}$$

Equation (3.15) represents a nonlinear, driftless system (i.e., no motion takes place under zero input), which has fewer control inputs than the generalized coordinates. There exists model singularity at $\delta_1 = \pm\frac{\pi}{2}$ or $\delta_2 = \pm\frac{\pi}{2}$, where the first vector field has a discontinuity. This situation occurs when the tractor's (or trailer's) front wheel is perpendicular to the car's longitudinal axis, causing the car to become jammed. However, this singularity is

not highly significant in practice because the steering angle is usually restricted within a limited range.

The controllability analysis of MSTTR can be carried out the same as SSTTR. The linearized system $\dot{\boldsymbol{x}} = g_1(\boldsymbol{x}_e)v + g_2(\boldsymbol{x}_e)\omega_1 + g_3(\boldsymbol{x}_e)\omega_2$ cannot pass the controllability rank condition

$$\operatorname{rank}\left(\begin{bmatrix} B & AB & A^2B & A^3B & A^4B & A^5B \end{bmatrix}\right) = 6. \tag{3.16}$$

The Lie algebra rank condition for MSTTR system (3.15) is obtained by

$$\operatorname{rank}\left([g_1, g_2, g_3, [g_1, g_2], [g_1, g_3], [g_2, g_3], [g_1, [g_2, g_3]], [g_2, [g_1, g_3]], \ldots]\right) = 6. \tag{3.17}$$

By computing the following Lie brackets

$$[g_1, g_2] = \begin{bmatrix} 0 \\ 0 \\ -\frac{\sec(\delta_1)^2}{\ell_1} \\ -\frac{\sec(\delta_1)^2}{\ell_1} \\ 0 \\ 0 \end{bmatrix}, [g_1, g_3] = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{\cos(\psi)\sec(\delta_2)^2}{\ell_2} \\ 0 \\ 0 \end{bmatrix}, [g_1, [g_1, g_2]] = \begin{bmatrix} -\frac{\sin(\theta_1)\sec(\delta_1)^2}{\ell_1} \\ \frac{\cos(\theta_1)\sec(\delta_1)^2}{\ell_1} \\ 0 \\ \frac{(\tan(\delta_2)\sin(\psi) - \cos(\psi))\sec(\delta_1)^2}{\ell_1 \ell_2} \\ 0 \\ 0 \end{bmatrix},$$

we can verify that the MSTTR system is controllable as it passes the Lie algebra rank condition (3.17). This result was also obtained by using coordinate transformation and state feedback in [32], which makes the controllability analysis easier by transforming the system into the chain form. One important result for controllable drift-free systems is that if the driftless control system is controllable, then its dynamic extension is also controllable. We will exploit this property for MSTTR in the next chapter.

In order to analyze the controllability of MSTTR around a trajectory, we consider the reference trajectory $\boldsymbol{x}_d = \begin{bmatrix} x_d & y_d & \theta_{1d} & \psi_d & \delta_{1d} & \delta_{2d} \end{bmatrix}^\top$ and the reference inputs $v_d, \omega_{1d}, \omega_{2d}$, which are always available due to differential flatness of MSTTR [131]. Next, we obtain the time-varying system matrices for the linearized system around the trajectory

$$A = \begin{bmatrix} 0 & 0 & -v_d \sin\theta_{1d} & 0 & 0 & 0 \\ 0 & 0 & v_d \cos\theta_{1d} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{v_d \sec(\delta_{1d})^2}{\ell_1} & 0 \\ 0 & 0 & 0 & A_{44} & \frac{v_d \sec(\delta_{1d})^2}{\ell_1} & \frac{-v_d \sec(\delta_{2d})^2 \cos\psi_d}{\ell_2} \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, B = \begin{bmatrix} \cos\theta_{1d} & 0 & 0 \\ \sin\theta_{1d} & 0 & 0 \\ \frac{\tan\delta_{2d}}{\ell_1} & 0 & 0 \\ B_{41} & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

where

$$A_{44} = \frac{-v_d \tan \delta_{2d} \sin \psi_d - \cos \psi_d}{\ell_2}, \quad B_{41} = \frac{\tan \delta_{1d}}{\ell_1} - \frac{\tan \delta_{2d} \cos \psi_d + \sin \psi_d}{\ell_2}.$$

Assuming the special case of a linear trajectory with constant velocity v_d, where $\theta_{1d} = \theta_0$ and $\psi_d = \delta_{1d} = \delta_{2d} = 0$, we can verify that the linearized system passes the condition (3.16). The asymptotic stabilizability of the system around an arbitrary trajectory has been investigated in [131].

3.3 Control Background

We review the concept of stability analysis using Lyapunov theory, which builds a basis for reviewing safety analysis using control barrier functions. These concepts will be used to develop safety-critical methods in the remaining chapters of the thesis. As the original definitions for CLF and CBF are defined based on the continuous system dynamics of the form (3.1), we adhere to this formulation throughout this section.

3.3.1 Stability and control Lyapunov function

Stability is a fundamental aspect of the study of control systems. Determining the stability of a system is often the first and most crucial step, as an unstable system is typically useless and can even pose significant dangers. Lyapunov theory provides a powerful method for analyzing the stability of control systems. However, its application is not limited to verifying the stability and it can be used to synthesize a stabilizing control policy. In the following, first, we review fundamental definitions based on the Lyapunov stability theory. We concentrate on the stability analysis of autonomous systems around an equilibrium point. Based on the definitions of the Lyapunov direct method, we introduce the control Lyapunov function as a method to stabilize nonlinear systems.

Stability

Consider the control affine system (3.1). In Equation (3.2), we introduced an autonomous closed-loop system $\dot{\boldsymbol{x}} = F_{cl}(\boldsymbol{x})$ resulting from applying a time-invariant state feedback $\boldsymbol{u} = \pi(\boldsymbol{x})$ to the system. Suppose that $\boldsymbol{x}_e \in \mathcal{X}$ is an equilibrium point of the closed-loop system. For convenience, we state all definitions and theorems in this section for

the case that the equilibrium point is at the origin. It can be shown that using a change of variables, any non-zero equilibrium point can be transformed to the origin, ensuring a one-to-one correspondence between the original and transformed systems.

Intuitively, an equilibrium point is stable if all solutions can be kept arbitrarily close to the equilibrium point by starting sufficiently close to it. In the following, we have a formal definition of stability in the sense of Lyapunov.

Definition 3.1. *The equilibrium point $\boldsymbol{x}_e = \mathbf{0}$ is stable, if for each $\epsilon > 0$ there is $\delta = \delta(\epsilon) > 0$ such that*

$$\|\boldsymbol{x}(0)\|_2 < \delta \implies \|x(t)\|_2 < \epsilon, \quad \forall t > 0.$$

We say that $\boldsymbol{x}_e = \mathbf{0}$ is unstable if it is not stable.

If the origin of the closed-loop system is stable in the sense of Lyapunov, we say that the control policy π is stabilizing. However, Definition 3.1 does not imply that the trajectories started close to the origin converge to the origin as time goes to infinity. We define this form of stability below.

Definition 3.2. *The equilibrium point $\boldsymbol{x}_e = \mathbf{0}$ is asymptotically stable, if it is stable and if in addition there exists δ such that*

$$\|\boldsymbol{x}(0)\|_2 < \delta \implies \lim_{t\to\infty} \boldsymbol{x}(t) = \mathbf{0}.$$

From the control point of view, asymptotic stability constitutes a more desirable stable behavior for nonlinear systems as it ensures convergence to the equilibrium which provides greater reliability. However, in many engineering scenarios, knowing that a system will reach its equilibrium point after infinite time is not enough. The requirements demand an estimation of how fast trajectories converge to the origin. Thus, a more strict notion of stability is defined below for this purpose.

Definition 3.3. *The equilibrium point $\boldsymbol{x}_e = \mathbf{0}$ is exponentially stable if it is stable and there exist two strictly positive constants M and α such that*

$$\|\boldsymbol{x}(t)\|_2 \leq M\|\boldsymbol{x}(0)\|_2 e^{-\alpha t}, \forall t > 0,$$

for all $\boldsymbol{x}(0) = \boldsymbol{x}_0$ in some neighborhood around the origin.

The constant α is called the rate of convergence and shows that the states of an exponentially stable system converge to the origin faster than an exponential function. The

above definitions characterize the local behavior of the trajectories (i.e., how the trajectories evolve when they start near the equilibrium point). If the condition corresponding to asymptotic (exponential) stability holds for any initial states, the equilibrium point is said to be globally asymptotically (exponentially) stable. The most powerful and general approach to study the stability of nonlinear systems is the Lyapunov theory which includes the Lyapunov linearization method and the direct method for stability analysis. While the linearization method examines the local stability of the nonlinear system around its equilibrium point, the Lyapunov direct method is not limited to the local motion. In this method, the stability properties of a nonlinear system are determined by constructing a scalar energy-like function and checking the time derivation of this function.

Theorem 3.1 (Local Stability [82])**.** *Let $\boldsymbol{x}_e = \mathbf{0}$ be an equilibrium point for (3.2) and $D \in \mathbb{R}^n$ be a domain containing the origin. Let $V : D \to \mathbb{R}$ be a continuously differentiable function, such that*

$$V(\mathbf{0}) = 0 \quad \textit{and} \quad V(\boldsymbol{x}) > 0, \quad \textit{in} \quad D \setminus \{\mathbf{0}\}, \tag{3.18}$$

$$\dot{V}(\boldsymbol{x}) \leq 0 \quad \textit{in} \quad D, \tag{3.19}$$

then, $\boldsymbol{x}_e = \mathbf{0}$ is stable, Moreover, if

$$\dot{V}(\boldsymbol{x}) < 0 \quad \textit{in} \quad D \setminus \{\mathbf{0}\}, \tag{3.20}$$

then, $\boldsymbol{x}_e = \mathbf{0}$ is asymptotically stable.

A continuous scalar function satisfying (3.18) is called locally positive definite and if the condition is satisfied over the whole state space then it is globally positive definite. The scalar function V defined in Theorem 3.1, which satisfies (3.18) and (3.19) is called a Lyapunov function.

It can be shown that if the Lyapunov function satisfies conditions (3.18) and (3.20) in the whole state space instead of a domain and has an additional property of radially unboundedness (i.e., $\|\boldsymbol{x}\| \to \infty \implies \|V(\boldsymbol{x})\| \to \infty$), then the system is globally asymptotically stable (Theorem 3.2, [82]).

Control Lyapunov function

Lyapunov theory is not just a method for analyzing stability; it also provides guidelines for synthesizing a nonlinear controller method. In this section, we will review the CLF as a method for designing a stabilizing controller for control affine systems (3.1). Before stating the formal definition of CLFs, we review the definition of a special class of functions.

Definition 3.4. *A Lipschitz continuous function $\gamma : [0, a) \to [0, \infty)$, $a > 0$, is said to be a class $\mathcal{K}$ function if it is strictly increasing and $\gamma(0) = 0$. It is called class $\mathcal{K}_\infty$ function if $a = \infty$, and $\gamma(r) \to \infty$ as $r \to \infty$.*

Definition 3.5 (Control Lyapunov function [15])**.** *Let $V : \mathcal{X} \to \mathbb{R}_+$ be a positive-definite, continuously differentiable and radially unbounded function. We say V is an CLF for (3.1) if there exist a class $\mathcal{K}_\infty$ function $\gamma : \mathbb{R}_+ \to \mathbb{R}_+$ such that*

$$\inf_{\boldsymbol{u} \in \mathbb{R}^m} L_f V(\boldsymbol{x}) + L_g V(\boldsymbol{x}) \boldsymbol{u} + \gamma(V(\boldsymbol{x})) \leq 0, \tag{3.21}$$

where $L_f V(\boldsymbol{x}) = \frac{\partial V(\boldsymbol{x})}{\partial \boldsymbol{x}} f$ and $L_g V(\boldsymbol{x}) = \frac{\partial V(\boldsymbol{x})}{\partial \boldsymbol{x}} g$ are the Lie derivatives of $V(\boldsymbol{x})$ along the vector fields f and g.

If there exists a CLF for system (3.1), then we can show that the system is globally asymptotically stabilizable [15]. Thus, it is desirable to find a state feedback control law $\pi(\boldsymbol{x})$ such that it satisfies

$$L_f V(\boldsymbol{x}) + L_g V(\boldsymbol{x}) \pi(\boldsymbol{x}) + \gamma(V(\boldsymbol{x})) \leq 0, \tag{3.22}$$

for any $\boldsymbol{x} \in \mathcal{X} \setminus \{\boldsymbol{0}\}$. An important property of the CLF method is that it boils down the stabilization problem into satisfying a constraint, which is an affine constraint in control input. Then, we can solve the following constrained optimization problem to enforce the stabilizing condition (3.22):

$$\begin{aligned} \pi_{CLF}(\boldsymbol{x}) = \arg \min_{\boldsymbol{u} \in \mathbb{R}^m} \|\boldsymbol{u}\|_2^2, \\ s.t. \quad L_f V(\boldsymbol{x}) + L_g V(\boldsymbol{x}) \boldsymbol{u} + \gamma(V(\boldsymbol{x})) \leq 0, \end{aligned} \tag{3.23}$$

which finds the smallest control input that satisfies the CLF condition. In the case that a desired control policy exists, we can also modify the cost (3.23) such that it finds the closest control input satisfying the CLF condition to the desired input. It is also desirable for control design if $\pi_{CLF}(\boldsymbol{x})$ is locally Lipschitz. It has been shown in [60] that a sufficient condition for locally Lipschitz continuity of $\pi_{CLF}(\boldsymbol{x})$ is that the vector fields f, g, and the gradient of V with respect to them are each locally Lipschitz continuous [60]. Definition 3.5 characterizes a CLF in the case that the control input is unconstrained. However, there are many control applications in which the input is restricted due to the actuator limitations, i.e., $\boldsymbol{u} \in \mathcal{U} \subset \mathbb{R}^m$. In this case, the CLF condition (3.21) may not be satisfied at every $\boldsymbol{x} \in \mathcal{X} \setminus \{\boldsymbol{0}\}$, even if V is a valid CLF for (3.1). This motivates the following lemma from [15].

Lemma 3.1. *(Proposition 2.2 in [95]) Let $V : \mathcal{X} \to \mathbb{R}_+$ be a CLF for system (3.1) with $\mathcal{U} \subset \mathbb{R}^m$ as the compact set of admissible inputs. For each strictly positive constant c denote the sublevel set of V by $\Omega_c = \{\boldsymbol{x} \in \mathcal{X} \subseteq \mathbb{R}^n : V(\boldsymbol{x}) \leq c\}$. If there exists a $c_i > 0$ such that*

$$\inf_{\boldsymbol{u} \in \mathcal{U}} L_f V(\boldsymbol{x}) + L_g V(\boldsymbol{x}) \boldsymbol{u} + \gamma(V(\boldsymbol{x})) \leq 0, \tag{3.24}$$

holds for all $\boldsymbol{x} \in \Omega_{c_i} \backslash \{\boldsymbol{0}\}$, then the origin is locally asymptotically stabilizable from $\boldsymbol{x} \in \Omega_{c_i}$, and the set Ω_{c_i} is a subset of the region of attraction of the origin.

We denote the largest sub-level set of V satisfying (3.24) by $\Omega_{c_{max}}$, which is the largest set from which the system (3.1) is stabilizable. Additionally, restricting the class $\mathcal{K}_\infty$ function to be linear leads to exponential stability. If there exists a compact subset $D \subseteq \Omega_{c_{max}}$ such that the following holds for some $\lambda > 0$

$$\inf_{\boldsymbol{u} \in \mathcal{U}} L_f V(\boldsymbol{x}) + L_g V(\boldsymbol{x}) \boldsymbol{u} + \lambda V(\boldsymbol{x}) \leq 0,$$

for all $\boldsymbol{x} \in D$, then any trajectory of the system starting from $\boldsymbol{x}_0 \in \Omega_{c_{max}} \subseteq D$ will be driven to the origin with the exponential rate of convergence [11].

3.3.2 Safety and control barrier function

In this section, a notion of safety in control systems is defined and the basic ideas behind control barrier functions are reviewed. Finally, a quadratic program is introduced to enforce safety and stability to a nominal controller.

Safety

Although the definition of safety in control systems is dependent on the application, a common approach can be found in many cases. Safety is guaranteed by avoiding the hazard or reducing the chance of putting the system at risk. Therefore, a mathematical representation of these notions is needed in order to develop safety-critical control strategies. But, how can we model a hazard or a potential risk for our systems?

Consider the state space of the dynamical system (3.1) as the set of all possible variables that gives a complete description of the system at any particular time. Suppose we are able to determine the safe area in the state space. Then, we can separate this region from the whole space and obtain a set of states in which the system is guaranteed to be safe. This idea establishes a framework to verify that the system is safe by checking if the states are

evolving in the safe set. This desirable behavior is called set invariance and can be verified using Nagumo's theorem [113] (see [110] for an English translation). Then, the goal of a safety-critical controller would be enforcing a control law such that the system satisfies the conditions of the set invariance, which is the backbone of the CBF method. Before introducing the main idea of CBFs, we review the concept of set invariance. Consider the autonomous system

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x}), \tag{3.25}$$

where $\boldsymbol{x} \in \mathbb{R}^n$ is the sate vector, and the vector field $f \colon \mathbb{R}^n \to \mathbb{R}^n$ is locally Lipschitz in $\boldsymbol{x}$. Suppose this system starts from an initial state $\boldsymbol{x}_0$. Solution $\boldsymbol{x}(t)$ of (3.25) is unique on $I(\boldsymbol{x}_0)$, where $I(\boldsymbol{x}_0) = [0, I_{max})$ is the maximal interval of existence for $\boldsymbol{x}(t)$ with initial condition $\boldsymbol{x}_0$. We want to evaluate whether the solution to (3.25) starting from this initial state, stays in a predefined set or not. We define this set by the zero superlevel set of a differentiable function $h(\boldsymbol{x})$ as

$$\Omega = \{\boldsymbol{x} \in \mathbb{R}^n \mid h(\boldsymbol{x}) \geq 0\}. \tag{3.26}$$

We assume that the system starts at this set (i.e., $\boldsymbol{x}_0 \in \Omega$). According to Nagumo's invariance principle, if the derivative of $h(\boldsymbol{x})$ along with the trajectories of (3.25) satisfies

$$\dot{h}(\boldsymbol{x}) \geq 0, \quad \forall \boldsymbol{x} \in \partial\Omega, \tag{3.27}$$

where $\partial\Omega$ denotes the boundary of the set, then the system trajectories never leave this set. The condition (3.27) essentially pushes back the trajectory into the set, when it reaches the boundary of the set. As a result, if we find a function $h(\boldsymbol{x})$ that satisfies (3.27), then the zero superlevel set of the function will be invariant.

The Nagumo's invariance principle is similar to Lyapunov theorem in the sense that both verify a property of the system without solving the ODE explicitly. The main difference between them is that in Lyapunov function every positive level set is an invariant set. However, in Nagumo's theorem only the zero-level set of the function $h(\boldsymbol{x})$ (which can be considered as the flipped version of the Lyapunov function) is an invariant set.

Control barrier function

Now we want to synthesize a control input that guarantees the system's safety. While the initial results on set invariance addressed systems without inputs, the work on viability theory extended them to controlled dynamical systems [16]. Consider the control affine system (3.1), it's closed-loop system (3.2) and the maximal interval of existence $I(\boldsymbol{x}_0) =$

$[0, I_{max})$ for the solutions of the closed-loop system. We denote the safe set by $\mathcal{C}$. In consequence, as long as the states evolve in $\mathcal{C}$ over time, the system is safe. Therefore, the notion of safety can be defined as the ability of the controller to keep future states within the safe set. The problem of designing such a controller can be defined in the context of set invariance. The goal is then designing a control method, which is able to formally guarantee the forward invariance of the set $\mathcal{C}$.

Definition 3.6. *(Forward invariant set) A set $\mathcal{C}$ is said to be forward invariant, with respect to (3.2), if for any $x(0) \in \mathcal{C}$, the solution $\boldsymbol{x}(t) \in \mathcal{C}$ for all $t \in I(\boldsymbol{x}_0)$.*

We assume that the system always starts at the safe set (i.e., $\boldsymbol{x}_0 \in \mathcal{C}$). If we prove that the set $\mathcal{C}$ is forward invariant, then the system's safety is ensured. We use CBFs to certify the safety to the control system (3.1). CBFs effectively use the idea of Nagumo's theorem to ensure the forward invariance of a desired set. Consider the safe set $\mathcal{C}$ to be defined as

$$\mathcal{C} = \{\boldsymbol{x} \in \mathbb{R}^n \mid h(\boldsymbol{x}) \geq 0\}, \tag{3.28}$$

$$\partial\mathcal{C} = \{\boldsymbol{x} \in \mathbb{R}^n \mid h(\boldsymbol{x}) = 0\}, \tag{3.29}$$

$$Int(\mathcal{C}) = \{\boldsymbol{x} \in \mathbb{R}^n \mid h(\boldsymbol{x}) > 0\}, \tag{3.30}$$

where $h\colon \mathbb{R}^n \to \mathbb{R}$ is a continuously differentiable function, and is designed such that the system satisfies a safety criterion (i.e., never collides with an obstacle, never pass a certain limit, etc.). The time derivative of $h(\boldsymbol{x})$ with respect to state trajectories is

$$\dot{h}(\boldsymbol{x}) = L_f h(\boldsymbol{x}) + L_g h(\boldsymbol{x})\boldsymbol{u}, \tag{3.31}$$

where $L_f h(\boldsymbol{x}) = \frac{\partial h(\boldsymbol{x})}{\partial \boldsymbol{x}} f$ and $L_g h(\boldsymbol{x}) = \frac{\partial h(\boldsymbol{x})}{\partial \boldsymbol{x}} g$ are the Lie derivatives of $h(\boldsymbol{x})$ along the vector fields f and g. Before formulating the CBFs, we need to give some definitions.

Definition 3.7. *(Extended class $\mathcal{K}$ function) A continuous function $\alpha : (-b, a) \to \mathbb{R}$, $a, b \in \mathbb{R}_+$, is said to be an extended class $\mathcal{K}$ function if it is strictly increasing and $\alpha(0) = 0$.*

Definition 3.8. *(Relative Degree) Given a r^{th}-order continuously differentiable function $h\colon \mathcal{X} \to \mathbb{R}$, we say $h(\boldsymbol{x})$ has a relative degree of r with respect to system (3.1) in $D \subset \mathbb{R}^n$, if for all $\boldsymbol{x} \in D$, $h^{(r)}(\boldsymbol{x}, \boldsymbol{u}) = L_f^r h(\boldsymbol{x}) + L_g L_f^{r-1} h(\boldsymbol{x})\boldsymbol{u}$, with $L_g L_f^{r-1} h(\boldsymbol{x}) \neq 0$ and $L_g L_f^k h(\boldsymbol{x}) = 0$ for $k < r - 1$.*

We assume that the function $h(\boldsymbol{x})$, has relative degree one with respect to system (3.1). We now have set up the foundations to define the CBFs.

Definition 3.9. *(Control barrier function, [9]) Given the control system (3.1) and the safe set $\mathcal{C}$ as in (3.28), the continuously differentiable function $h(\boldsymbol{x})$ is called a control barrier function if there exists an extended class $\mathcal{K}$ function α such that for all $\boldsymbol{x} \in \mathcal{X}$*

$$\sup_{\boldsymbol{u}\in\mathcal{U}} L_f h(\boldsymbol{x}) + L_g h(\boldsymbol{x})\boldsymbol{u} + \alpha(h(\boldsymbol{x})) \geq 0. \tag{3.32}$$

Comparing with the previous safety condition in (3.27), the condition introduced in (3.32) is less conservative. The extended class $\mathcal{K}$ function α in (3.32) essentially allows the control system to utilize the entire safe region in its state space. However, once the trajectory reaches the boundary of the safe set (e.g., $h(\boldsymbol{x}) = 0$), the right-hand side of inequality (3.32) approaches zero, preventing the system from entering the unsafe region. The function α also maintains a smooth transition when the trajectory approaches the boundary of the safe set $\partial\mathcal{C}$ and pushes it back into the set. Thus, the behavior of the system at the boundary of the safe set depends on the choice of this function.

Consider the following set that contains all the input values that render the set $\mathcal{C}$ forward invariant

$$\mathcal{U}_s \triangleq \{\boldsymbol{u} \in \mathcal{U} \mid L_f h(\boldsymbol{x}) + L_g h(\boldsymbol{x})\boldsymbol{u} \geq -\alpha(h(\boldsymbol{x}))\},$$

which motivates stating the following theorem

Theorem 3.2. *([10]) Consider the safe set $\mathcal{C}$ in (3.28), and CBF $h(\boldsymbol{x})$, $\frac{\partial h}{\partial \boldsymbol{x}}(\boldsymbol{x}) \neq 0$, for all $\boldsymbol{x} \in \partial\mathcal{C}$, any Lipschitz continuous $\boldsymbol{u} \in \mathcal{U}_s$ guarantees that the set $\mathcal{C}$ is forward invariant for the system (3.1).*

Note that Definition 3.9 is only applicable to systems with relative degree one. However, in many applications, CBFs have relative degrees greater than one. A modification of the original theory is investigated in [114], which introduced the exponential control barrier function. Exponential CBF effectively uses the idea of virtual input-output linearization to enforce arbitrarily high relative safety constraints.

Exponential control barrier functions

Let the safe set $\mathcal{C}$ and its associated subsets be as defined in (3.28)-(3.30), where $h(\boldsymbol{x})$ is smooth. However, suppose that the relative degree of $h(\boldsymbol{x})$ is $r > 1$. Thus, we have

$$h^{(r)}(\boldsymbol{x}, \boldsymbol{u}) = L_f^r h(\boldsymbol{x}) + L_g L_f^{r-1} h(\boldsymbol{x})\boldsymbol{u}, \tag{3.33}$$

which means the input $\boldsymbol{u}$ does not appear in the time derivatives of h until the r^{th} order. We can leverage the Exponential Control Barrier Functions (ECBF) to certify the forward invariance of the set $\mathcal{C}$.

Definition 3.10. *(Exponential control barrier function, [114]) Given system (3.1) and the safe set $\mathcal{C}$ as in (3.28), the continuously differentiable function $h(\boldsymbol{x})$ with relative degree r is called an exponential control barrier function if there exists a gain vector $\boldsymbol{k} \in \mathbb{R}^{1\times r}$ such that*

$$\sup_{\boldsymbol{u}\in\mathcal{U}} \left\{ L_f^r h(\boldsymbol{x}) + L_g L_f^{r-1} h(\boldsymbol{x}) \boldsymbol{u} \right\} \geq -\boldsymbol{k}\boldsymbol{\eta}(\boldsymbol{x}), \tag{3.34}$$

where $\boldsymbol{\eta}(\boldsymbol{x}) = \begin{bmatrix} h(\boldsymbol{x}) & \dot{h}(\boldsymbol{x}) & \dots & h^{(r-1)}(\boldsymbol{x}) \end{bmatrix}^\top$.

In [114], it has been proved that if $\boldsymbol{k}$ satisfies specific properties, then forward invariance of the safe set can be also achieved for the high-order CBFs. While all of the class $\mathcal{K}$ functions are restricted to be linear in ECBFs, another definition of high-order CBFs is proposed in [169], which generalized the idea to time-variant structure, which also accepts arbitrary class $\mathcal{K}$ functions. In the next chapters, we will use these concepts to develop our safety-critical control structure for systems that have uncertainty in the model.

3.3.3 Uncertainty decomposition and robust safety verification

The first step in any control system design is system modeling. A common approach is to use physical principles to derive a model of the system, such as mobile robot modeling discussed earlier in this chapter. This typically results in a nominal model of the true underlying dynamics. Consider the nominal model

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x}, \boldsymbol{u}), \tag{3.35}$$

we can rewrite the actual system dynamics as

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x}, \boldsymbol{u}) + \underbrace{f_{true}(\boldsymbol{x}, \boldsymbol{u}) - f(\boldsymbol{x}, \boldsymbol{u})}_{=\Delta_n(\boldsymbol{x},\boldsymbol{u})}, \tag{3.36}$$

where the function $\Delta_n \colon \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ represents the error between the true dynamics and the nominal model. While it is difficult to directly characterize the set in which Δ_n evolves, assumptions about the true dynamics based on domain knowledge can be made to construct a set $\mathrm{D}_n(\boldsymbol{x}, \boldsymbol{u}) \subseteq \mathbb{R}^n$ such that $\Delta_n(\boldsymbol{x}, \boldsymbol{u}) \in \mathrm{D}_n(\boldsymbol{x}, \boldsymbol{u})$ for all $\boldsymbol{x} \in \mathcal{X}, \boldsymbol{u} \in \mathcal{U}$. This set is typically an overapproximation of the model error, leading to a conservative

robust safety design when used for safety filter construction. Data-driven techniques aim to address this issue by approximating the true dynamics or modeling the error, reducing the error term to a smaller quantity $\Delta_l \colon \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$.

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x}, \boldsymbol{u}) + f_l(\boldsymbol{x}, \boldsymbol{u}) + \underbrace{\Delta_n(\boldsymbol{x}, \boldsymbol{u}) - f_l(\boldsymbol{x}, \boldsymbol{u})}_{=\Delta_l(\boldsymbol{x}, \boldsymbol{u})}, \tag{3.37}$$

where $f_l \colon \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}^n$ represents the learned dynamics that aim to correct the nominal term. The notion of ISSf provides a framework for quantifying the effect of uncertainty on safety guarantees by characterizing changes in the set kept forward invariant. In the following, we present this notion for the case that we have a residual learning error $\Delta_l(\boldsymbol{x}, \boldsymbol{u})$ in our design.

Consider the control affine system (3.1) as the nominal model, we have

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x}) + g(\boldsymbol{x})\boldsymbol{u} + f_l(\boldsymbol{x}, \boldsymbol{u}) + \Delta_l(\boldsymbol{x}, \boldsymbol{u}). \tag{3.38}$$

Let $\mathcal{C}$ be the zero superlevel set of the CBF $h(\boldsymbol{x})$, representing the safe set for (3.38). Suppose that the safety control law $\boldsymbol{u}_s \colon \mathbb{R}^n \times \mathbb{R}^m \to \mathcal{U}$ is obtained by using the learned dynamics such that there exists $\alpha \in \mathcal{K}$

$$\nabla h(\boldsymbol{x}) \left(f(\boldsymbol{x}) + g(\boldsymbol{x})\boldsymbol{u}_s(\boldsymbol{x}, \boldsymbol{u}) + f_l(\boldsymbol{x}, \boldsymbol{u}_s(\boldsymbol{x}, \boldsymbol{u})) + \Delta_l(\boldsymbol{x}, \boldsymbol{u}) \right) \geq -\alpha(h(\boldsymbol{x})), \tag{3.39}$$

for all $\boldsymbol{x} \in \mathbb{R}^n$ and $\boldsymbol{u} \in \mathbb{R}^m$. We further assume that there exists $\bar{\delta} \in \mathbb{R}_+$ such that

$$|\nabla h(\boldsymbol{x}) \Delta_l(\boldsymbol{x}, \boldsymbol{u})| \leq \bar{\delta}, \tag{3.40}$$

for all $\boldsymbol{x} \in \mathbb{R}^n$ and $\boldsymbol{u} \in \mathbb{R}^m$. In other words, this inequality implies that the effect of the model mismatch on the time derivative of the CBF is bounded. We combine (3.39) and (3.40), which yields

$$\dot{h}(\boldsymbol{x}, \boldsymbol{u}) \geq -\alpha(h(\boldsymbol{x})) - \bar{\delta}. \tag{3.41}$$

We know that $\alpha \in \mathcal{K}$, which implies that $\alpha^{-1} \in \mathcal{K}$. We can use this property to conclude that

$$h(\boldsymbol{x}) \leq \alpha^{-1}(-\bar{\delta}_l) \implies \dot{h}(\boldsymbol{x}, \boldsymbol{u}) \geq 0. \tag{3.42}$$

Consider the superlevel set of the CBF $h(\boldsymbol{x})$ at $\alpha^{-1}(-\bar{\delta})$ as

$$\mathcal{C}_{\bar{\delta}_l} = \left\{ \boldsymbol{x} \in \mathbb{R}^n \mid h(\boldsymbol{x}) \geq \alpha^{-1}(-\bar{\delta}) \right\}. \tag{3.43}$$

Inequality (3.42) implies that the time derivative of the barrier function $h(\boldsymbol{x})$ is non-negative on the boundary of $\mathcal{C}_{\bar{\delta}_l}$ and negative at the interior of the set, therefore, we can

use Nagumo's theorem to conclude that $\mathcal{C}_{\bar{\delta}_l}$ is forward invariant. This establishes a fundamental robustness property of safety-critical methods with CBFs, since it states that the set kept forward invariant does not increase dramatically with small perturbations by the residual learning error, but rather scales proportionally. Intuitively, a more accurate learning model can obtain tighter bounds on the learning error (smaller $\bar{\delta}_l$), which in turn controls the expansion of the safe set.

Robustness analysis via the notion of ISSf is introduced in [128], which is an equivalent version of input-to-state stability [140] in safety verification of systems with uncertainty. This idea has been followed by the authors in [84], which extended the results to robust CBF design. Many works on the learning-based CBF safety filter were established based on these results.

3.3.4 Enforcing CBF and CLF

In the last two sections, we formally defined the stability and safety of the dynamical systems. CLF and CBF have been introduced as tools to systematically design a controller that leads to a stable and safe closed-loop behavior. Each method requires satisfying a constraint to verify the associated property. Now, we will focus on enforcing these constraints.

As stability was one of the primary objectives in control from the beginning, most of the methods inherently satisfy stability in various forms (stabilizing around a point or tracking a reference path or trajectory). However, the notion of safety has often been neglected at the design level and it has been gradually added to the essential performance requirements. Consequently, we often need to enforce the CBF safety constraint in a modular approach to the existing performance-driven controllers, which is equivalent to modifying the nominal control law. This intervention must be performed in a way that it has the least effect on the nominal control law. Consider system (3.1) with a nominal control policy $\boldsymbol{u}_{nom}$, Lipschitz continuous in $\boldsymbol{x}$. Suppose that a CBF design is available with constraint (3.32) to verify the safety of this system. Inequality (3.32), imposes an affine condition on the control input, which can be incorporated into a min-norm optimization. The resulting optimization is convex and can be solved in real time.

$$\boldsymbol{u}_s = \underset{\boldsymbol{u}\in\mathcal{U}}{\arg\min} \ \|\boldsymbol{u} - \boldsymbol{u}_{nom}\|_2^2 \tag{3.44a}$$

$$\text{s.t.} \quad L_f h(\boldsymbol{x}) + L_g h(\boldsymbol{x})\boldsymbol{u} + \alpha(h(\boldsymbol{x})) \geq 0. \tag{3.44b}$$

The QP (3.44) (considering that the set of admissible control inputs is compact and convex) is solved point-wise in time to apply $\boldsymbol{u}_{nom}$ to the system only if it complies with

(3.32). Note that the CBF method boils down the problem of long-term safety verification into a single time step condition (3.44b). This is a desirable property, as it removes the need to plan far ahead into the future to make sure that the system never leaves the safe set. Additionally, this framework has the capability to incorporate the stabilization property of CLF with the safety guarantees from the CBF in a unified approach.

$$\boldsymbol{u}_s = \underset{\boldsymbol{u}\in\mathbb{R}^m, d\in\mathbb{R}}{\arg\min} \ \|\boldsymbol{u}\|_2^2 + \rho d^2 \tag{3.45a}$$

$$\text{s.t.} \quad L_f V(\boldsymbol{x}) + L_g V(\boldsymbol{x})\boldsymbol{u} + \gamma(V(\boldsymbol{x})) \leq d, \tag{3.45b}$$

$$L_f h(\boldsymbol{x}) + L_g h(\boldsymbol{x})\boldsymbol{u} + \alpha(h(\boldsymbol{x})) \geq 0, \tag{3.45c}$$

where $d \in \mathbb{R}$ is a slack variable. This approach prevents infeasibility by making the CLF a soft constraint giving the CBF constraint a priority in case of conflict. This optimization requires perfect knowledge of the model as the Lie derivatives are computed using the system dynamics. In this thesis, we will eliminate the assumption that an accurate model is available and address the problem of designing a safety-critical control using this framework under model uncertainty.

3.3.5 Model predictive control

MPC is an advanced control method that effectively uses model predictions for optimization purposes. This control strategy has been successfully implemented in various industrial applications ranging from process control to autonomous driving. MPC has a simple, yet effective control design that allows the controller to take the state and input constraints into account while achieving control objectives. In this section, we give an overview of the standard MPC formulation. In the next chapter, we will use the idea to design a nominal controller for trajectory tracking for a tractor-trailer system. First, we introduce the optimal control problem that we aim to solve. The goal of this control synthesis is to achieve the desired performance in an optimal fashion. Consider a control task for the system (3.1) with the initial point $\boldsymbol{x}(0)$. The control policy should obtain the optimal input sequence $U^{opt} = \{u_1^*, u_2^*, \dots\}$ solving the following optimization problem

$$\min_{u_t} \quad \sum_{t=0}^{\infty} l(\boldsymbol{x}_t, \boldsymbol{u}_t) \tag{3.46a}$$

$$\text{s.t.} \quad \boldsymbol{x}_{t+1} = f(\boldsymbol{x}_t) + g(\boldsymbol{x}_t)\boldsymbol{u}_t, \quad \forall t \in \{0, 1, \dots\}, \tag{3.46b}$$

$$\boldsymbol{x}_t \in \mathcal{X}, \boldsymbol{u}_t \in \mathcal{U}, \quad \forall t \in \{0, 1, \dots\}, \tag{3.46c}$$

$$\boldsymbol{x}_0 = \boldsymbol{x}(0). \tag{3.46d}$$

We denote the solution of this infinite-time optimal control problem by $J^*(\boldsymbol{x}(0))$. The stage cost $l(.,.)$ is designed so that the system satisfies a predetermined control objective. There are two challenges regarding solving the optimization problem (3.46) and applying the resulting optimal control sequence to the system. First, finding the solution to the minimizing problem (3.46a) can be hard to compute in an infinite (long) horizon. Second, the model used to predict the future states (3.46b) may not be accurate enough. In consequence, the resulting cumulative prediction errors can negatively affect the output of the optimization problem. MPC has addressed these issues by reformulating the problem (3.46). First, it solves the problem in a finite horizon N, shorter than the task duration. Second, it measures the states at each sampling time and updates the optimal control sequence based on more reliable predictions with new measured information from the system. We can formulate the MPC optimization problem with a discretized form of the system dynamics as follows

$$\min_{u_t^{mpc}} \quad p(\boldsymbol{x}_N) + \sum_{t=0}^{N-1} l(\boldsymbol{x}_{t+n|t}, \boldsymbol{u}_{t+n|t}) \tag{3.47a}$$

$$\text{s.t.} \quad \boldsymbol{x}_{t+n+1|t} = f(\boldsymbol{x}_{t+n|t}) + g(\boldsymbol{x}_{t+n|t})\boldsymbol{u}_{t+n|t}, \forall n \in \{0, 1, \ldots, N-1\}, \tag{3.47b}$$

$$\boldsymbol{x}_{t+n|t} \in \mathcal{X}, \boldsymbol{u}_{t+n|t} \in \mathcal{U}, \forall n \in \{0, 1, \ldots N-1\}, \tag{3.47c}$$

$$\boldsymbol{x}_N \in \mathcal{X}_f, \tag{3.47d}$$

$$\boldsymbol{x}_0 = \boldsymbol{x}(t|t), \tag{3.47e}$$

where $u_t^{mpc} = [\boldsymbol{u}_{t|t}, \ldots, \boldsymbol{u}_{t+N-1|t}]^\top$ is the vector of optimized inputs in the prediction horizon N. The notation $\boldsymbol{x}_{t+n|t}$ is used to denote the state $\boldsymbol{x}$ at time $t+n$ predicted at the current time t. We take $\boldsymbol{u}(t) = \boldsymbol{u}^*_{t|t}(\boldsymbol{x}(t))$, from the optimal control sequence u^{mpc} and apply it to the system and measure the resulting state, which is used as the initial state for the next step as in (3.47e). The optimization problem is then repeated at time $t+1$, based on the new state $\boldsymbol{x}_{t+1|t+1} = \boldsymbol{x}(t+1)$, forming a moving or receding horizon control scheme.

The receding horizon approach alleviates the mentioned issues with infinite optimization, however, it introduces two implementation issues. First, the finite-time optimal control problem (3.47) may become infeasible after a few steps. Second, the obtained control sequence may not lead to trajectories that converge to the origin. We need to address these issues such that the short-sighted horizon leads to reliable long-term performance. Comparing with the infinite-time optimal control (3.46), the MPC problem has a terminal cost $p(\boldsymbol{x}_N)$ and terminal constraint set $\mathcal{X}_f$ which can be used to cover the mentioned shortcomings of the method. We essentially create a situation to simulate infinite-horizon by forcing the final state to be in an invariant set for which there exists an invariance-inducing controller, whose infinite-horizon cost can be expressed in closed form. We refer to [26]

for a detailed proof and procedure to ensure persistent feasibility and stability with the terminal cost and constraint set.

3.4 Gaussian Process Regression

In the last section, it has been noted that using classical robust control methods yields excessively conservative safety filters. Data-driven methods have been introduced as a solution to mitigate this issue. Let's revisit the regression problem in (3.37).

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x}, \boldsymbol{u}) + f_l(\boldsymbol{x}, \boldsymbol{u}) + \underbrace{\Delta_n(\boldsymbol{x}, \boldsymbol{u}) - f_l(\boldsymbol{x}, \boldsymbol{u})}_{=\Delta_l(\boldsymbol{x}, \boldsymbol{u})}.$$

Finding the learned dynamics $f_l(\boldsymbol{x}, \boldsymbol{u})$ can be formulated as a classical regression problem. Algorithms solving this problem are generally categorized into either parametric or nonparametric. The choice of learning method depends on the level of knowledge about the system. Parametric approaches have a fixed number of parameters, independent of the number of measurements. They are ideal when the structure of the true dynamics is known and fast predictions are a requirement of the design. On the other hand, nonparametric approaches have a variable number of parameters that grow with the size of the data, making them suitable for handling both structural and parametric uncertainties. However, they are computationally more expensive when fast predictions are needed. Gaussian process regression is a nonparametric approach that can serve as an effective framework for learning the error model. These technique provides bounds on the learning error, enabling the characterization of a state- and input-dependent set $\mathrm{D}_l(\boldsymbol{x}, \boldsymbol{u})$ such that $\Delta_l(\boldsymbol{x}, \boldsymbol{u}) \in \mathrm{D}(\boldsymbol{x}, \boldsymbol{u})$ for all $\boldsymbol{x} \in \mathcal{X}, \boldsymbol{u} \in \mathcal{U}$. In the next chapters, this techniques will be used to establish efficient learning-based safety mechanisms. In the following, we introduce the concept of Gaussian process regression.

We represent a GP approximation of $v\colon \mathcal{X} \to \mathbb{R}$ as $v(\boldsymbol{x}) \sim \mathcal{GP}\left(m(\boldsymbol{x}), k(\boldsymbol{x}, \boldsymbol{x}')\right)$, which is fully specified by its mean function $m\colon \mathcal{X} \to \mathbb{R}$ and covariance (kernel) function $k\colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. Our prior knowledge of the problem can shape the form of $m(\boldsymbol{x})$ and $k(\boldsymbol{x}, \boldsymbol{x}')$, considering that the covariance function must satisfy positive definiteness [167]. Given noisy measurements $w_j = v(\boldsymbol{x}_j) + \varepsilon_j, j \in \{1, \ldots, N\}$, which are corrupted by Gaussian noise $\varepsilon_j \sim \mathcal{N}(0, \sigma^2)$, a GP model can infer a posterior mean and variance for a test point $\boldsymbol{x}_t$ conditioned on the measurements

$$\begin{aligned} \mu(\boldsymbol{x}_t) &= \boldsymbol{w}^\top \left(K + \sigma_n^2 I\right)^{-1} \bar{K}^\top, \\ \sigma(\boldsymbol{x}_t)^2 &= k\left(x_t, x_t\right) - \bar{K}\left(K + \sigma_n^2 I\right)^{-1} \bar{K}^\top, \end{aligned} \tag{3.48}$$

where $\boldsymbol{w} \in \mathbb{R}^N$ is the vector of measurements w_j, $K \in \mathbb{R}^{N \times N}$ is the Gram matrix with elements $K_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$, and $\bar{K} = \big[k(x_t, x_1), \ldots, k(x_t, x_N)\big]^\top \in \mathbb{R}^N$. We considered zero prior mean to obtain (3.48).

Chapter 4

Multiple Control Barrier Functions

4.1 Introduction

The structure of real-world safety-critical systems (e.g. multi-body mobile robots) implies that a single CBF is not enough to guarantee the safety of the whole system and several safety criteria need to be considered for each component of the system. Despite the importance of this issue, the idea of the multiple CBFs scheme is addressed in a few studies. In [122], multivariate CBFs of relative degree two are designed to safely control a manipulator. However, the idea is not extendable to safety constraints with arbitrarily high relative degrees. Multiple safety constraints with relative degree one are also used in some applications such as obstacle avoidance in manipulator [88], constrained motion planning [136], and trajectory planning under temporal specifications [62]. In this chapter, we relax the assumption of relative degree one safety constraints for multiple CBFs structure.

Tractor-trailer systems are widely used in many applications ranging from road shipment to agricultural machinery. Although they are the most common mode of transportation in the supply chain, their poor maneuverability restricts their usage in safety-critical applications. Autonomous TTRs can be a good solution to address this issue. TTRs typically consist of a car-like tractor that tows either a passive or active trailer. Based on the trailer's articulation structure, these systems are categorized as SSTTRs or MSTTRs. Notably, MSTTRs feature an additional input mechanism for trailer navigation. Specifically, the rear axle of the trailer has independent steering, which provides the system with enhanced maneuverability. Consequently, MSTTRs offer superior mobility compared to SSTTRs.

Many tracking control methods have been introduced for SSTTRs. A nonlinear controller based on the Lyapunov direct method in [79], a robust lane-keeping control policy based on a prediction of the robot's motion in [58], and exponential convergence on the line-of-sight distance between the actual and the desired trajectories for the trailer in [77]. On the other hand, the proposed controllers for MSTTRs are not only concerned with the tracking problem but also try to overcome challenges regarding the TTR's autonomy. In [124], feedback linearization and backstepping methods have been used to address the problem of reducing the swept path width. In [101], a flatness-based trajectory planning method has been proposed. In [80], centralized, decentralized, and distributed nonlinear MPC were implemented. In [102], an MPC approach was used to solve the problem of stabilizing the system around a nominal path.

An accurate tracking control method is necessary for TTRs' autonomy but not sufficient. An obstacle avoidance control technique that guarantees safety, is essential for autonomous transportation. In the literature, obstacle avoidance of TTRs is mainly addressed in path planning methods. A multi-layered obstacle avoidance path planning approach for MSTTRs based on the particle swarm optimization method has been developed in [176]. In [178], an online trajectory planning method has been designed for the dynamic model of a tractor with an arbitrary number of trailers. In [89], collision-free paths were generated via a cooperative approach (using workspace information). As an alternative technique, obstacle avoidance was guaranteed via a reactive method (using installed sensors on the robot). In this work, we aim to develop a tracking control strategy enhanced with a reactive obstacle avoidance technique, which effectively uses the idea of a multiple CBFs scheme to generate a safe motion for both the tractor and trailer in the presence of several obstacles. To the best of our knowledge, this study is the first attempt to use CBFs to synthesize controllers enforcing safety over tractor-trailer systems.

4.2 Control of Tractor-trailer Systems

Designing a control methodology for tractor-trailer systems is divided into two categories: stabilization, and tracking. In stabilization control, the controller is designed such that the system steers from an initial state to a given terminal state (e.g., the origin of the state space). The tracking problem is concerned with following a given path or reference trajectory with the least possible error. However, there are some difficulties in designing a control methodology for tractor-trailer systems. While some of these difficulties originate from the underactuated nature of the system, others are a result of the rigid structure of the system. For instance, the space required by a truck-trailer during a turning maneuver

is an important safety aspect (which is denoted by swept-path width), especially in the narrow urban roads. Jackknifing is a type of accident that occurs when a truck with a separate cab and trailer folds at the pivot point, forming an acute angle between the two sections. Researchers tried to overcome these issues with the latest developments in the control theory over time.

The proposed controllers for MSTTRs are not only concerned with the tracking problem but also try to overcome challenges regarding the TTR's autonomy. In [124], the problem of reducing the swept path width has been formulated as a path-following problem and solved by the feedback linearization and backstepping methods. A flatness-based trajectory planning method has been proposed for a class of MSTTRs in [101]. Then, a sliding mode controller has been designed for the dynamic model to track the planned trajectory in the presence of input disturbances. MPC has been used in many MSTTRs tracking control schemes. Centralized, decentralized, and distributed structures of the nonlinear MPC were implemented on MSTTR, and a comparison study of their performances has been conducted in [80]. The path-following error model of the MSTTR has been derived, and based on that, an MPC approach was used to follow a nominal path in [102]. This work considered both the forward and backward motion control of the system.

An accurate tracking control method is necessary for TTRs' autonomy but not sufficient. An obstacle avoidance control technique that can guarantee safety is essential for autonomous transportation. In the literature, obstacle avoidance of TTRs is mainly addressed in path planning methods. A multi-layered planning approach for MSTTRs has been developed in [175, 176]. In the proposed method, first, a collision-free path was generated for the tractor by cubic splines. Then, initial knots were determined by applying the A* algorithm, and the particle swarm optimization method was used to find the shortest feasible path for the tractor. Finally, the Gaussian pseudospectral method was used to generate the trailer's path based on the one that was previously planned for the tractor. In [178], an online trajectory planning method has been developed for the dynamic model of a tractor with an arbitrary number of trailers. Dynamically feasible collision-free trajectories were generated by this planner, which can be tracked by any feedback controller. In [89], first, a trajectory has been generated for the trailer based on the desired tractor trajectory regardless of obstacles. Then redesigned collision-free paths were generated via a cooperative approach (using workspace information), or obstacle avoidance was guaranteed via a reactive approach (using installed sensors on the robot).

In the introduced motion planning methods for MSTTRs, which take obstacle avoidance into account, only the method introduced in [89] can avoid colliding with an unexpected obstacle for the tractor on the planned trajectory. In [146] reachability analysis is used to dynamically determine the safe regions. These regions then converted to state- and

input-dependent constraints in an MPC framework to generate a local collision-free path for tractor-trailers. The safe set is generated only for the tractor, thus, the safety of the trailer is not formally guaranteed. In this chapter, we aim to develop a reactive obstacle avoidance technique and tracking control strategy that can generate a safe motion for both the tractor and trailer in an MSTTR in the presence of multiple obstacles. We propose an obstacle avoidance technique based on CBFs, which provably guarantees collision-free maneuvers for not only the tractor but also the trailer. To the best of our knowledge, our study is the first attempt to use CBFs to synthesize controllers enforcing safety over the tractor-trailer systems.

4.3 MIMO Feedback Linearization

In this section, we revisit the feedback linearization technique for MIMO systems as the backbone for the proposed method in the next section. Inspired by the SISO feedback linearization, this technique has been developed to benefit from the simplicity and effectiveness of existing linear control methods on nonlinear systems. In particular, we start with the squared MIMO system (systems with the same number of inputs and outputs) and pose the question of whether there exists a state feedback control $u = \alpha(\boldsymbol{x}) + \beta(\boldsymbol{x})\nu$, and a change of variables $\boldsymbol{z} = T(\boldsymbol{x})$, that convert the nonlinear system (4.5) into a linear system. Before start developing the method we need to define some concepts from differential geometry. These definitions are mathematical results, which are independent of control systems.

Definition 4.1. *(Distribution) For vector fields $f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_k(\boldsymbol{x})$ on $D \subset \mathbb{R}^n$, let $\Delta(\boldsymbol{x}) = \{f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_k(\boldsymbol{x})\}$ be the subspace of $\mathbb{R}^n$ spanned by the set of vectors $f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_k(\boldsymbol{x})$ at any fixed $x \in D$. A distribution is the collection of all vector spaces $\Delta(\boldsymbol{x})$ for $\boldsymbol{x} \in D$ and represented by $\Delta = span\{f_1, f_2, \ldots, f_k\}$.*

The dimension of the subspace spanned by a set of vector fields is defined by

$$dim(\Delta(\boldsymbol{x})) = rank[f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_k(\boldsymbol{x})], \tag{4.1}$$

which may vary for different values of x. If we have $\Delta = span\{f_1, f_2, \ldots, f_k\}$, with the set of linearly independent vector fields $f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_k(\boldsymbol{x})$ for all $x \in D$, then $dim(\Delta(\boldsymbol{x})) = k$ for all $x \in D$. In this case, we say that Δ is a nonsingular distribution on D.

Definition 4.2. *(Lie bracket) Let f, g be two vector fields on $D \subset \mathbb{R}^n$. The Lie bracket of f and g is the third vector field defined by*

$$[f, g] = \nabla g f - \nabla f g, \tag{4.2}$$

where ∇f and ∇g are Jacobian matrices. The Lie bracket is also represented by $ad_f g$.

Definition 4.3. *(Involutive distribution) A distribution Δ is involutive if*

$$g_1 \in \Delta \text{ and } g_2 \in \Delta \rightarrow [g_1, g_2] \in \Delta. \tag{4.3}$$

Definition 4.3 essentially means, that if we have two vector fields in a distribution, the Lie bracket of them does not increase the dimension of the distribution. Thus, it defines the notion of invariant distributions. In other words, the Lie bracket does not expand the distribution. We have the following lemma for nonsingular distributions.

Lemma 4.1. *If Δ is a nonsingular distribution on D, generated by $f_1, f_2, \ldots, f_k$, then Δ is involutive if and only if*

$$[f_i, f_j] \in \Delta, 1 \leq i, j \leq k, \tag{4.4}$$

Proof. $f_1, \ldots, f_k$ are vector fields of Δ, which is the necessary condition. Let g_1 and g_2 be any two vector fields of Δ. Then, we can represent them locally by

$$g_1(\boldsymbol{x}) = \sum_{i=1}^{k} c_i(\boldsymbol{x}) f_i(\boldsymbol{x}), g_2(\boldsymbol{x}) = \sum_{i=1}^{k} d_i(\boldsymbol{x}) f_i(\boldsymbol{x}).$$

Then the Lie bracket of the vector fields $g_1(\boldsymbol{x})$ and $g_2(\boldsymbol{x})$ is

$$\begin{aligned}
[g_1, g_2] &= [\sum_{i=1}^{k} c_i f_i, \sum_{i=1}^{k} d_i f_i] \\
&= \frac{\partial}{\partial x}(\sum_{j=1}^{k} d_j f_j) \sum_{i=1}^{k} c_i f_i - \frac{\partial}{\partial x}(\sum_{i=1}^{k} c_i f_i) \sum_{j=1}^{k} d_j f_j \\
&= \sum_{j=1}^{k} d_j \frac{\partial f_j}{\partial x} \sum_{i=1}^{k} c_i f_i + \sum_{j=1}^{k} f_j \frac{\partial d_j}{\partial x} \sum_{i=1}^{k} c_i f_i - \sum_{i=1}^{k} c_i \frac{\partial f_i}{\partial x} \sum_{j=1}^{k} d_j f_j - \sum_{i=1}^{k} f_i \frac{\partial c_i}{\partial x} \sum_{j=1}^{k} d_j f_j \\
&= \sum_{i=1}^{k} \sum_{j=1}^{k} \{c_i d_j [\frac{\partial f_j}{\partial x} f_i - \frac{\partial f_i}{\partial x} f_j] + c_i (L_{f_i} d_j) f_j - d_j (L_{f_j} c_i) f_i\} \\
&= \sum_{i=1}^{k} \sum_{j=1}^{k} \{c_i d_j [f_i, f_j] + c_i (L_{f_i} d_j) f_j - d_j (L_{f_j} c_i) f_i\}.
\end{aligned}$$

Since Δ is a nonsingular distribution, $[f_i, f_j] \in \Delta$. Thus we conclude that $[g_1, g_2] \in \Delta$, which proves the sufficiency. □

Definition 4.4. *(Diffeomorphism) A continuously differentiable map $\boldsymbol{z} = T(\boldsymbol{x})$ with a continuously differentiable inverse $\boldsymbol{x} = T^{-1}(\boldsymbol{z})$, is called diffeomorphism.*

We consider the control-affine nonlinear systems as described in (3.1). However, we assume that this system has m outputs

$$\begin{aligned} \dot{\boldsymbol{x}} &= f(\boldsymbol{x}) + g(\boldsymbol{x})\boldsymbol{u}, \\ y_1 &= h_1(\boldsymbol{x}), \\ &\vdots \\ y_m &= h_m(\boldsymbol{x}), \end{aligned} \tag{4.5}$$

where $h_j(\boldsymbol{x})\colon \mathbb{R}^n \to \mathbb{R}, j = 1, \ldots, m$, are smooth functions.

Definition 4.5. *A nonlinear system of the form (4.5), where $f : D \to \mathbb{R}^n$ and $g : D \to \mathbb{R}^{n\times m}$, are sufficiently smooth on $D \subset \mathbb{R}^n$, is said to be feedback linearizable, if there exists a diffeomorphism transformation $z = T(\boldsymbol{x})$, where $T : D \to \mathbb{R}^n$ and $T(D)$ contains the origin such that it transforms (4.5) into*

$$\dot{z} = Az + B\beta(\boldsymbol{x})[u - \alpha(\boldsymbol{x})], \tag{4.6}$$

where the pair (A, B) is controllable and $\beta(\boldsymbol{x})$ is nonsingular for all $x \in D$.

4.3.1 Input-output linearization

Consider the derivative of the j^{th} output y_j of the system (4.5) with respect to time, which is represented by Lie derivative format as

$$\dot{y}_j = L_f h_j + \sum_{i=1}^{m} (L_{g_i} h_j) u_i. \tag{4.7}$$

Suppose that each of the $L_{g_i} h_j = 0, j = 1, \ldots, m$, and the inputs do not appear in (4.7). Consider r_j as the smallest integer such that at least one of the inputs appears in the derivatives of each of the outputs

$$y_j^{(r_j)} = L_f^{r_j} h_j + \sum_{i=1}^{m} (L_{g_i} L_f^{r_j - 1} h_j) u_i, \tag{4.8}$$

where at least one of the $L_{g_i}L_f^{r_j-1}h_j \neq 0$. Note that all the previous Lie derivatives of $h_j(\boldsymbol{x})$ with respect to g_i are zero. We define the matrix $E(\boldsymbol{x})$ as

$$E(\boldsymbol{x}) = \begin{bmatrix} L_{g_1}L_f^{r_1-1}h_1 & \dots & L_{g_m}L_f^{r_1-1}h_1 \\ L_{g_1}L_f^{r_2-1}h_2 & \dots & L_{g_m}L_f^{r_2-1}h_2 \\ \vdots & \ddots & \vdots \\ L_{g_1}L_f^{r_m-1}h_m & \dots & L_{g_m}L_f^{r_m-1}h_m \end{bmatrix}. \tag{4.9}$$

Based on these derivations, we define the relative degree for MIMO systems.

Definition 4.6. *(Vector Relative Degree [73]) The system (4.5) is said to have a vector relative degree $r_1, r_2, \dots, r_m$ at a point $\boldsymbol{x}^0$ if*

$$L_{g_i}L_f^k h_i(\boldsymbol{x}) = 0, \tag{4.10}$$

for all $i = 1, \dots, m$, for all $k < r_i - 1$, and the matrix $E(\boldsymbol{x})$ is nonsingular for all x in a neighborhood of $\boldsymbol{x}^0$.

If system (4.10) has well defined vector relative degree $r_1, r_2, \dots, r_m$, then (4.10) can be written in the matrix form

$$\begin{bmatrix} y_1^{(r_1)} \\ y_2^{(r_2)} \\ \vdots \\ y_m^{(r_m)} \end{bmatrix} = \begin{bmatrix} L_f^{r_1}h_1 \\ L_f^{r_2}h_2 \\ \vdots \\ L_f^{r_m}h_m \end{bmatrix} + E(\boldsymbol{x}) \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_m \end{bmatrix}. \tag{4.11}$$

Since $E(\boldsymbol{x}^0)$ is nonsingular, we can conclude that for a neighborhood $\mathcal{X}_0$ of $\boldsymbol{x}^0$, $E(\boldsymbol{x})$ is bounded away from nonsingularity for all $\boldsymbol{x} \in \mathcal{X}_0$. Thus, we can design the control law

$$\begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_m \end{bmatrix} = E^{-1}(\boldsymbol{x}) \left(\begin{bmatrix} \nu_1 \\ \nu_2 \\ \vdots \\ \nu_m \end{bmatrix} - \begin{bmatrix} L_f^{r_1}h_1 \\ L_f^{r_2}h_2 \\ \vdots \\ L_f^{r_m}h_m \end{bmatrix} \right). \tag{4.12}$$

This feedback control law is denoted by static feedback linearizing control law. Substituting (4.12) into (4.11) yields the following linear closed loop system

$$\begin{bmatrix} h_1^{(r_1)} \\ h_2^{(r_2)} \\ \vdots \\ h_m^{(r_m)} \end{bmatrix} = \begin{bmatrix} \nu_1 \\ \nu_2 \\ \vdots \\ \nu_m \end{bmatrix}. \tag{4.13}$$

Lemma 4.2. *([73]) Suppose the system (4.5) has a vector relative degree $\{r_1, \ldots, r_m\}$ at $\boldsymbol{x}_0$, then the row vectors*

$$\begin{aligned}
&\mu_1^1 = h_1(\boldsymbol{x}), \mu_2^1 = L_f h_1(\boldsymbol{x}), \ldots, \mu_{r_1}^1 = L_f^{r_1-1} h_1(\boldsymbol{x}),\\
&\mu_2^1 = h_2(\boldsymbol{x}), \mu_2^2 = L_f h_2(\boldsymbol{x}), \ldots, \mu_{r_2}^2 = L_f^{r_2-1} h_2(\boldsymbol{x}),\\
&\vdots\\
&\mu_1^m = h_m(\boldsymbol{x}), \mu_2^m = L_f h_m(\boldsymbol{x}), \ldots, \mu_{r_m}^m = L_f^{r_m-1} h_m(\boldsymbol{x}),
\end{aligned} \tag{4.14}$$

are linearly independent.

In the following Proposition, we check the conditions which guarantee the existence of a set of coordinates that transforms the system (4.5) into the normal form.

Proposition 4.1. *([73]) Suppose (4.5) has a vector relative degree $r_1, \ldots, r_m$ at a point x^0. Then $r_1 + \cdots + r_m \leq n$. Then, set*

$$\begin{aligned}
\mu_1^i(\boldsymbol{x}) &= h_i(\boldsymbol{x}),\\
\mu_2^i(\boldsymbol{x}) &= L_f h_i(\boldsymbol{x}),\\
&\vdots\\
\mu_{r_i}^i(\boldsymbol{x}) &= L_f^{r_i-1} h_i(\boldsymbol{x}),
\end{aligned} \tag{4.15}$$

if $r = r_1 + \cdots + r_m < n$, we need to complete the basis by finding $n - r$ more functions such that the mapping

$$T(\boldsymbol{x}) = [\mu_1^1(\boldsymbol{x}), \ldots, \mu_{r_1}^1(\boldsymbol{x}), \ldots, \mu_1^m(\boldsymbol{x}), \ldots, \mu_{r_m}^m(\boldsymbol{x}), \phi_{r+1}(\boldsymbol{x}), \ldots, \phi_n(\boldsymbol{x})]^\top, \tag{4.16}$$

has a nonsingular Jacobian at $\boldsymbol{x}^0$. Thus, $\Phi(\boldsymbol{x})$ qualifies as a local coordinates transformation near $\boldsymbol{x}^0$. Additionally, if the distribution $G = span\{g_1, \ldots, g_m\}$ is involutive near $\boldsymbol{x}^0$, it is always possible to choose $\phi_{r+1}(\boldsymbol{x}), \ldots, \phi_n(\boldsymbol{x})$ such that

$$L_{g_j} \phi_i(\boldsymbol{x}) = 0, \tag{4.17}$$

for all $r + 1 \leq i \leq n$, for all $1 \leq j \leq m$, and all $\boldsymbol{x}$ around $\boldsymbol{x}^0$.

In particular, we find $\phi_{r+1}, \ldots, \phi_n$ such that based on condition (4.17), the input does not appear in the additional $n - r$ dynamics. Using these variables the system (4.5) is

reduced to the normal form

$$\begin{aligned}
\dot{\mu}_1^1 &= \mu_2^1, \ldots, \dot{\mu}_{r_1}^1 = b_1(\mu,\phi) + \sum_{j=1}^{m} a_j^1(\mu,\phi)u_j, \\
\dot{\mu}_1^2 &= \mu_2^2, \ldots, \dot{\mu}_{r_2}^2 = b_2(\mu,\phi) + \sum_{j=1}^{m} a_j^2(\mu,\phi)u_j, \\
&\vdots \\
\dot{\mu}_1^m &= \mu_2^m, \dot{\mu}_{r_m}^m = b_m(\mu,\phi) + \sum_{j=1}^{m} a_j^m(\mu,\phi)u_j, \\
\dot{\phi} &= q(\mu,\phi) + p(\mu,\phi)u, \\
y_1 &= \mu_1^1, \\
&\vdots \\
y_m &= \mu_1^m.
\end{aligned} \tag{4.18}$$

Here, the transformation $\Psi : \boldsymbol{x} \to (\mu, \phi)$ is a diffeomorphism mapping from $\boldsymbol{x}$ into the normal form coordinates. If the decoupling matrix $E(\boldsymbol{x})$ is nonsingular, the zero dynamics can be obtained by solving the following problem.
Find an initial state and an input $\bar{u}$ so that the outputs $y_j(t) \equiv 0, \forall t$. From the normal form given in (4.18) we have

$$\bar{u}(t) = -E^{-1}(\mu,\phi)b(\mu,\phi). \tag{4.19}$$

This is the decoupling control law of (4.19) with $\nu = 0$. Also, the initial conditions on μ_i^j for all $i = 1, \ldots, r_j$ and $j = 1, \ldots, m$ have to be set to zero. The dynamics of ϕ can be obtained by

$$\dot{\phi} = q(0,\phi) - p(0,\phi)E^{-1}(0,\phi)b(0,\phi). \tag{4.20}$$

If $f(\boldsymbol{x}_0) = 0, h_i(\boldsymbol{x}_0) = 0$ for $i = 1, \ldots, m$ and Ψ maps $\boldsymbol{x}_0$ to the origin, then ϕ_0 is an equilibrium point of the zero dynamics of (4.20). Then, the system (4.5) is minimum phase if the equilibrium point 0 is asymptotically stable.

4.3.2 Full state feedback linearization

The main question is whether there exists a set of output functions $h_1(\boldsymbol{x}), \ldots, h_m(\boldsymbol{x})$ such that the system (4.5) has vector relative degree, which satisfies $r_1 + \cdots + r_m = n$.

Considering this, the system(4.5) can be converted into a controllable linear system by using the feedback law $u(\boldsymbol{x}) = E^{-1}(\boldsymbol{x})[-\alpha(\boldsymbol{x}) + \nu]$ and coordinate transformation $\mu = [\mu_0 \dots \mu_{r_i-1}]^\top, \mu_i = L_f^j h_i(\boldsymbol{x}), 0 \leq j \leq r_i - 1, 1 \leq i \leq m$. First, we define the distributions:

$$\begin{aligned} G_0(\boldsymbol{x}) &= span\{g_1(\boldsymbol{x}), \dots, g_m(\boldsymbol{x})\}, \\ G_1(\boldsymbol{x}) &= span\{g_1(\boldsymbol{x}), \dots, g_m(\boldsymbol{x}), ad_f g_1(\boldsymbol{x}), \dots, ad_f g_m(\boldsymbol{x})\}, \\ &\vdots \\ G_i(\boldsymbol{x}) &= span\{ad_f^k g_j(\boldsymbol{x}) | 0 \leq k \leq i, 1 \leq j \leq m\}, \end{aligned} \tag{4.21}$$

for $i = 1, \dots, n-1$. Then we have the following proposition

Proposition 4.2. *(Proposition 9.16 in [134]) Consider the system(4.5) where $g(\boldsymbol{x})$ is the control vector field. Suppose the matrix $g(\boldsymbol{x}^0)$ has rank m. Then, there exists a set of real-valued functions $h_1(\boldsymbol{x}), \dots, h_m(\boldsymbol{x})$ such that the system (4.5) has vector relative degree $\{r_1, \dots, r_m\}$, with*

$$r_1 + r_2 + \cdots + r_m = n, \tag{4.22}$$

if and only if the following conditions are satisfied

1. *for each $0 \leq i \leq n-1$, the distribution G_i has constant dimension in a neighborhood $\mathcal{X}_0$ of $\boldsymbol{x}^0$,*
2. *the distribution G_{n-1} has dimension n,*
3. *for each $0 \leq i \leq n-2$, the distribution G_i is involutive.*

The following Remarks extend the application of the MIMO feedback linearization to a more general class of control-affine nonlinear systems.

Remark 4.1. *The results established so far can be extended to a system having a different number of inputs and outputs. Note that in order to have vector relative degree, the decoupling matrix must have full row rank (equal to the number of outputs).*

Remark 4.2. *If the decoupling matrix is singular, the system does not have a vector relative degree. In this case, we can use dynamic extension which is essentially choosing new inputs as the derivative of the existing system inputs such that the corresponding decoupling matrix becomes nonsingular [73]. The control system is designed based on the additional dynamics and the new set of inputs. The original inputs will be computed by integration.*

4.4 Theoretical Analysis of Multiple CBFs

In this section, we introduce a multiple CBFs scheme which enforces several safety constraints with high relative degrees. This safety-critical control structure is essential in many challenging robotic applications that need to meet several safety criteria simultaneously. In order to illustrate the capabilities of the proposed method, we have addressed the problem of reactive obstacle avoidance for a class of tractor-trailer systems. Safety constraints can be designed such that the mobile robot keeps a predetermined distance from the obstacles and never collides with them.

Consider the following representation of the input-affine control system (3.1).

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x}) + G(\boldsymbol{x})\boldsymbol{u}, \tag{4.23}$$

where the actuation term is of the form $G(\boldsymbol{x}) = [g_1(\boldsymbol{x}), g_2(\boldsymbol{x}), \ldots, g_m(\boldsymbol{x})]$, with locally Lipschitz input vector fields $g_i \colon \mathbb{R}^n \to \mathbb{R}^n$, $i \in \{1, \ldots, m\}$. All other definitions are the same as Section 3.1.1. We begin with defining the safe sets associated with candidate CBFs with the goal of designing a controller that guarantees the forward invariance of these sets

$$\begin{aligned} \mathcal{C}_1 &= \{\boldsymbol{x} \in \mathbb{R}^n | h_1(\boldsymbol{x}) \geq 0\}, \\ \mathcal{C}_2 &= \{\boldsymbol{x} \in \mathbb{R}^n | h_2(\boldsymbol{x}) \geq 0\}, \\ &\vdots \\ \mathcal{C}_m &= \{\boldsymbol{x} \in \mathbb{R}^n | h_m(\boldsymbol{x}) \geq 0\}, \end{aligned} \tag{4.24}$$

where $h_i \colon \mathbb{R}^n \to \mathbb{R}$, $i \in \{1, \ldots, m\}$, are continuously differentiable functions that we design to satisfy several safety criteria. The sets $\partial\mathcal{C}_i$ and $Int(\mathcal{C}_i)$ are the boundary and the interior sets, respectively. We define the set of admissible states as

$$\mathcal{A} = \bigcap_{i=1}^{m} \mathcal{C}_i = \{\boldsymbol{x} \in \mathbb{R}^n | \forall 1 \leq i \leq m, h_i(\boldsymbol{x}) \geq 0\}. \tag{4.25}$$

We want to verify the forward invariance for all sets $\mathcal{C}_i$, which is equivalent to the forward invariance of their intersection. Therefore, we assume that the set of admissible states $\mathcal{A}$ is non-empty.

For a given $i \in \{1, \ldots, m\}$, suppose $h_i(\boldsymbol{x})$ has an arbitrary relative degree r_i, which is defined to be the smallest integer such that at least one of the inputs explicitly appears in the time derivatives of $h_i(\boldsymbol{x})$:

$$h_i^{(r_i)}(\boldsymbol{x}) = L_f^{r_i} h_i(\boldsymbol{x}) + \sum_{l=1}^{m} (L_{g_l} L_f^{r_i - 1} h_i(\boldsymbol{x})) u_l, \tag{4.26}$$

where at least one $L_{g_l} L_f^{r_i - 1} h_i(\boldsymbol{x})$ is non-zero for all $\boldsymbol{x} \in D \subseteq \mathbb{R}^n$. If we repeat this procedure for all $i \in \{1, \ldots, m\}$, equation (4.26) can be written in the matrix form

$$\begin{bmatrix} h_1^{(r_1)} \\ h_2^{(r_2)} \\ \vdots \\ h_m^{(r_m)} \end{bmatrix} = \begin{bmatrix} L_f^{r_1} h_1 \\ L_f^{r_2} h_2 \\ \vdots \\ L_f^{r_m} h_m \end{bmatrix} + E(\boldsymbol{x}) \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_m \end{bmatrix}, \tag{4.27}$$

where $h_i(\boldsymbol{x})$ is denoted by h_i for short. The matrix $E(\boldsymbol{x})$ is obtained by

$$E(\boldsymbol{x}) = \begin{bmatrix} L_{g_1} L_f^{r_1 - 1} h_1 & \ldots & L_{g_m} L_f^{r_1 - 1} h_1 \\ L_{g_1} L_f^{r_2 - 1} h_2 & \ldots & L_{g_m} L_f^{r_2 - 1} h_2 \\ \vdots & \ddots & \vdots \\ L_{g_1} L_f^{r_m - 1} h_m & \ldots & L_{g_m} L_f^{r_m - 1} h_m \end{bmatrix}, \tag{4.28}$$

which is called the decoupling matrix.

Suppose that the system has a vector relative degree according to Definition 4.6 and it is full-state feedback linearizable (i.e., it satisfies the conditions in Proposition 4.2). Then, for a given $\nu \in \mathbb{R}^m$, we can find $\boldsymbol{u}$ such that

$$\begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_m \end{bmatrix} = E^{-1}(\boldsymbol{x}) \left(\begin{bmatrix} \nu_1 \\ \nu_2 \\ \vdots \\ \nu_m \end{bmatrix} - \begin{bmatrix} L_f^{r_1} h_1 \\ L_f^{r_2} h_2 \\ \vdots \\ L_f^{r_m} h_m \end{bmatrix} \right). \tag{4.29}$$

This feedback linearizing control law exists since the decoupling matrix $E(\boldsymbol{x})$ is invertible. Substituting (4.29) into (4.27) yields the following expression

$$\begin{bmatrix} h_1^{(r_1)} \\ h_2^{(r_2)} \\ \vdots \\ h_m^{(r_m)} \end{bmatrix} = \begin{bmatrix} \nu_1 \\ \nu_2 \\ \vdots \\ \nu_m \end{bmatrix}. \tag{4.30}$$

Note that the system in (4.30) is also decoupled which enables us to divide it into m systems of the form

$$h_i^{(r_i)} = \nu_i, i \in \{1, \ldots, m\} \tag{4.31}$$

which can be represented in the state space format

$$\begin{aligned}\dot{\eta}_i(\boldsymbol{x}) &= F_i\eta_i(\boldsymbol{x}) + G_i\nu_i,\\ h_i(\boldsymbol{x}) &= C_i\eta_i(\boldsymbol{x}),\end{aligned} \tag{4.32}$$

where $\eta_i(\boldsymbol{x}) = \begin{bmatrix} h_i & L_f h_i & \dots & L_f^{r_i-1}h_i \end{bmatrix}^\top$, and we have $F_i \in \mathbb{R}^{r_i\times r_i}$, $G_i \in \mathbb{R}^{r_i\times 1}$, $C_i \in \mathbb{R}^{1\times r_i}$ as defined below

$$\begin{aligned} F_i &= \begin{bmatrix} 0 & 1 & 0 & \dots & 0\\ 0 & 0 & 1 & \dots & 0\\ \vdots & \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & 0 & \dots & 1\\ 0 & 0 & 0 & \dots & 0 \end{bmatrix}, G_i = \begin{bmatrix} 0\\ 0\\ \vdots\\ 0\\ 1 \end{bmatrix},\\ C_i &= \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \end{bmatrix}, \end{aligned} \tag{4.33}$$

for $i \in \{1, \dots, m\}$, which is clearly linear and controllable. Next, we can separately control m systems in (4.5). We apply the state feedback control of the form $\nu_i = -K_i\eta_i(\boldsymbol{x})$ to the system (4.30), where the elements of $K_i = \begin{bmatrix} k_i^0 & k_i^1 & \dots & k_i^{r_i-1} \end{bmatrix}$, are our design parameters. We can arbitrarily select the gain vector K_i such that all the eigenvalues of the closed-loop matrix $A_i^{cl} = F_i - G_iK_i$ are real and negative. Then, $h_i(\boldsymbol{x}) = C_i e^{A_i^{cl}t}\eta_i(\boldsymbol{x}_0)$, and by the comparison lemma [82], any $\nu_i \geq -K_i\eta_i(\boldsymbol{x})$ yields $h_i(\boldsymbol{x}) \geq C_i e^{A_i^{cl}t}\eta_i(\boldsymbol{x}_0)$.

HOCBF is a more general definition of CBFs with high relative degrees introduced in [169]. We define the notion of multiple CBFs with high relative degrees as multi-CBFs and prove the forward invariance of the safe set associated with them. Toward this end, the following definition and results will be helpful.

Definition 4.7. *(Definition 4.3 in [82]) A continuous function $\beta : [0, a) \times [0, \infty) \to [0, \infty)$ is a class $\mathcal{KL}$ function if for each fixed s, the mapping $\beta(r, s)$ belongs to class $\mathcal{K}$ functions with respect to r, and for each fixed r, $\beta(r, s)$ is decreasing with respect to s and $\beta(r, s) \to 0$ as $s \to 0$.*

Lemma 4.3. *(Lemma 4.4 in [82]) Consider the scalar autonomous differential equation $\dot{y}(t) = -\alpha(y(t))$, $y(t_0) = y_0$, where α is a class $\mathcal{K}$ function defined on $[0, a)$, for all $0 \leq y_0 < a$, this equation has a unique solution $y(t)$ defined for all $t \geq t_0$. Moreover, $y(t) = \sigma(y_0, t - t_0)$, where σ is a class $\mathcal{KL}$ function defined on $[0, a) \times [0, \infty)$.*

Lemma 4.4. *Let α be a locally Lipschitz class $\mathcal{K}$ function and $h : [0, t_f) \to \mathbb{R}$ be a continuous function. If $\dot{h}(t) \geq -\alpha(h(t))$ for all $t \in [0, t_f)$, and $h(0) \geq 0$, then there exists a class $\mathcal{KL}$ function $\beta : [0, \infty) \times [0, \infty) \to [0, \infty)$ such that $h(t) \geq \beta(h(0), t)$, and $h(t) \geq 0$, for all $t \in [0, t_f)$.*

Proof. Define the differential equality $\dot{z}(t) = -\alpha(z(t))$, $z(0) = h(0)$. Since α is a locally Lipschitz function, there exists a unique solution $z(t)$. As $z(0) \geq 0$, based on Lemma 4.3, $z(t) = \beta(z(0), t)$ for all $t \in [0, t_f)$, where β is a class $\mathcal{KL}$ function. Next, since $\dot{h}(t) \geq -\alpha(h(t))$ for all $t \in [0, t_f)$, based on comparison techniques in [87], $h(t) \geq z(t)$ for all $t \in [0, t_f]$. Thus, $h(t) \geq \beta(h(0), t)$ for all $t \in [0, t_f]$, consequently, $h(t) \geq 0$ for all $t \in [0, t_f]$. Thus, $h(t) \geq \beta(h(0), t)$ for all $t \in [0, t_f]$. □

To ensure the forward invariance of each safe set $\mathcal{C}_i$ in (4.24), we define the set of functions $m_i^j \colon \mathbb{R}^n \to \mathbb{R}$ as

$$m_i^j(\boldsymbol{x}) = \dot{m}_i^{j-1}(\boldsymbol{x}) + \alpha_i^j(m_i^{j-1}(\boldsymbol{x})), \tag{4.34}$$

where α_i^j is a class $\mathcal{K}$ function with $m_i^0(\boldsymbol{x}) = h_i(\boldsymbol{x})$ for $j \in \{1, \ldots, r_i\}$, and $i \in \{1, \ldots, m\}$. We also define their superlevel sets $\mathcal{M}_i^j$ and the interior of these sets as

$$\mathcal{M}_i^j = \{\boldsymbol{x} \in \mathbb{R}^n | m_i^{j-1}(\boldsymbol{x}) \geq 0\}, \tag{4.35}$$

$$Int(\mathcal{M}_i^j) = \{\boldsymbol{x} \in \mathbb{R}^n | m_i^{j-1}(\boldsymbol{x}) > 0\}, \tag{4.36}$$

for all $j \in \{1, \ldots, r_i\}$, and $i \in \{1, \ldots, m\}$.

Definition 4.8. *(Multi-CBFs) Let the functions $m_i^j(\boldsymbol{x})$ and sets $Int(\mathcal{M}_i^j)$ be defined by (4.34) and (4.36), respectively. The set of m continuously differentiable functions of the form $\{h_1, \ldots, h_m\}$, where $h_i \colon \mathbb{R}^n \to \mathbb{R}$, $i \in \{1, \ldots, m\}$, is called a multi-CBFs set if for each i, h_i and its derivatives up to order r_i, are locally Lipschitz continuous, and there exists a set of continuously differentiable class $\mathcal{K}$ functions α_i^j, $j \in \{1, \ldots, r_i\}$, such that $m_i^{r_i}(\boldsymbol{x}) \geq 0$. This inequality can be represented as*

$$\sup_{u \in \mathcal{U}} [L_f^{r_i} h_i(\boldsymbol{x}) + \sum_{l=1}^{m} (L_{g_l} L_f^{r_i - 1} h_i(\boldsymbol{x})) u_l + R(h) + \alpha_i^{r_i}(m_i^{r_i - 1}(\boldsymbol{x}))] > 0, \tag{4.37}$$

for each $i \in \{1, \ldots, m\}$, for all $\boldsymbol{x} \in \bigcap_{i=1}^{m} \bigcap_{j=1}^{r_i} Int(\mathcal{M}_i^j)$, where $R(h_i)$ denotes the Lie derivatives of h_i with respect to f up to order $r_i - 1$.

Based on Definition 4.8, we state the following theorem for the forward invariance of the associated sets.

Theorem 4.1. *Given system (4.23) with a set of multi-CBFs $\{h_1, \ldots, h_m\}$, any Lipschitz continuous control $\boldsymbol{u}$ that satisfies m constraints of the form (4.37) renders the set $\bigcap_{i=1}^{m} \bigcap_{j=1}^{r_i} Int(\mathcal{M}_i^j)$ forward invariant for system (4.23).*

Proof. Since $\{h_1, \ldots, h_m\}$ is a set of multi-CBFs, the derivatives of h_i are locally Lipschitz. Also, the continuous differentiability of α_i^j implies their Lipschitz continuity. Thus, $m_i^j(\boldsymbol{x})$, $j \in \{1, \ldots, r_i - 1\}$ are Lipschitz continuous. The control $\boldsymbol{u}$ appears in the r_i^{th} time derivative of h_i, thus it only exists in the expression of $m_i^{r_i}(\boldsymbol{x})$. Since $\boldsymbol{u}$ is also Lipschitz continuous, $m_i^{r_i}(\boldsymbol{x})$ is Lipschitz continuous. Since $h_i(\boldsymbol{x})$ belongs to a multi-CBFs set, then $m_i^{r_i}(\boldsymbol{x}) \geq 0$, i.e., $\dot{m}_i^{r_i-1}(\boldsymbol{x}) + \alpha_i^{r_i}(m_i^{r_i-1}(\boldsymbol{x})) \geq 0$, for all $\boldsymbol{x} \in \bigcap_{j=1}^{r_i} \mathcal{M}_i^j$. By Lemma 4.4, since $\boldsymbol{x}_0 \in \mathcal{M}_i^{r_i}$, $m_i^{r_i-1}(\boldsymbol{x}) > 0$, which implies that $\boldsymbol{x}(t) \in Int(\mathcal{M}_i^{r_i})$ for all $t \geq 0$. In consequence, we have $\dot{m}_i^{r_i-2}(\boldsymbol{x}) + \alpha_i^{r_i-1}(m_i^{r_i-2}(\boldsymbol{x})) > 0$. We use Lemma 4.4 another time. Since $\boldsymbol{x}_0 \in Int(\mathcal{M}_i^{r_i-1})$, we also have $m_i^{r_i-2}(\boldsymbol{x}) > 0$, which implies that $\boldsymbol{x}(t) \in Int(\mathcal{M}_i^{r_i-1})$ for all $t \geq 0$. Repeating this procedure, we get that $\boldsymbol{x}(t) \in Int(\mathcal{M}_i^j)$, for all $j \in \{1, \ldots, r_i\}$, for all $t \geq 0$. Therefore, the set $\bigcap_{i=1}^{m} \bigcap_{j=1}^{r_i} Int(\mathcal{M}_i^j)$ is forward invariant for system (4.23). □

We have addressed the case that the number of inputs and CBFs are the same (i.e., the decoupling matrix is squared). Remarks 4.1 and 4.2 generalize the idea to the systems with a different number of inputs and CBF.

4.4.1 Quadratic programming optimization

In this section, we demonstrate that the multiple CBF structure can be integrated with an arbitrary nominal controller to ensure safety. Specifically, we employ the safety filter introduced in Section 3.3.4. This approach yields a set of m safety constraints of the form (4.37), which are affine in the control input $\boldsymbol{u}$. The following QP optimization modifies the nominal control law to generate a safe input.

$$\boldsymbol{u}_s(\boldsymbol{x}) = \underset{\boldsymbol{u} \in \mathbb{R}^m}{\arg\min}\ \boldsymbol{u}^\top \boldsymbol{u} - 2\boldsymbol{u}_{nom}^\top \boldsymbol{u}, \tag{4.38}$$

$$\text{s.t.} \quad A_{mcbf}(\boldsymbol{x})\boldsymbol{u} \leq b_{mcbf}(\boldsymbol{x}), \tag{4.39}$$

where $\boldsymbol{u}_{nom}$ is a locally Lipschitz continuous nominal control law. The constraint (4.39) is the matrix form of (4.37), where

$$\begin{aligned} A_{mcbf}(\boldsymbol{x}) &= -E(\boldsymbol{x}), \\ b_{mcbf}(\boldsymbol{x}) &= \begin{bmatrix} L_f^{r_1} h_1 + K_1 \eta_1(\boldsymbol{x}) \\ L_f^{r_2} h_2 + K_2 \eta_2(\boldsymbol{x}) \\ \vdots \\ L_f^{r_m} h_m + K_m \eta_m(\boldsymbol{x}) \end{bmatrix}. \end{aligned} \tag{4.40}$$

Note that the non-singularity of the decoupling matrix $E(\boldsymbol{x})$ is essential for the existence of a control input $\boldsymbol{u}$ that satisfies (4.39).

Remark 4.3. *In [1], we previously stated that the control law $\boldsymbol{u}_s$, obtained by solving the QP optimization problem (4.38), is Lipschitz continuous and provided a proof in [2] based on the conditions established in [111]. However, a later publication [112] identified a counterexample demonstrating that the original conditions in [111] do not always guarantee Lipschitz continuity, as illustrated in [126]. Since CBF-based safety filters fundamentally rely on the assumption that the solution to the QP optimization, which defines the safe control input, is Lipschitz continuous, this property is critical for ensuring the Lipschitz continuity of the closed-loop system. Addressing this limitation remains an open research challenge, particularly in the context of the multi-CBF structure proposed in this chapter.*

A recent trend in this topic is to guarantee the feasibility under the presence of input constraints [177] or multiple CBFs [172]. The latter derived the necessary and sufficient conditions for the feasibility of the QP optimization for two safety constraints and a system with a single input.

4.5 An Application to Autonomous Tractor-trailer

In this section, the problem of obstacle avoidance while tracking a given reference trajectory is addressed for autonomous tractor-trailer systems. The reference trajectory is not necessarily safe, and we assume that there might be some obstacles on the reference trajectory. Thus, reactive obstacle avoidance based on CBFs is required. The structure of the MSTTR enables us to deploy the multi-CBFs method. The trajectory tracking objective is achieved using an Linear Time-Varying Model Predictive Control (LTV)-MPC approach as the nominal controller, and a multiple CBFs scheme is designed for the tractor and trailer to ensure the system's safety.

4.5.1 Kinematic model of the tractor-trailer robot

The MSTTR mobile robot comprises a car-like tractor and a trailer equipped with actively steerable wheels. Each body has a reference point located on the midpoint of its rear axle. Let $p_i = [x_i, y_i]^\top, i = 1, 2$, be the coordinates of the reference points on the Cartesian plane. The steering angles of the tractor and trailer are δ_1 and δ_2, respectively. The angles θ and ψ denote the orientation of the tractor and the relative angle between two vehicle segments, respectively. v is the linear velocity, and a is the acceleration of the tractor. We consider low-speed turning maneuvers in this work. Thus, the assumption of pure rolling of wheels holds and wheel slip dynamical effects can be neglected. Figure 4.1

shows an on-axle tractor-trailer mobile robot. The state vector of the system denoted by $\boldsymbol{x} = [x_1, y_1, v, a, \theta, \psi, \delta_1, \delta_2]^\top$ and the kinematic model of the system can be described by

$$\begin{aligned}
\dot{x}_1 &= v\cos\theta, \\
\dot{y}_1 &= v\sin\theta, \\
\dot{v} &= a, \\
\dot{a} &= J, \\
\dot{\theta} &= \frac{v}{l_1}\tan\delta_1, \\
\dot{\psi} &= \frac{v}{l_1}\tan\delta_1 - \frac{v}{l_2}(\tan\delta_2\cos\psi + \sin\psi), \\
\dot{\delta}_1 &= \omega_1, \\
\dot{\delta}_2 &= \omega_2,
\end{aligned} \tag{4.41}$$

where l_1 and l_2 are the distances between the front and rear axle (wheelbase) of the bodies. The coordinates of the trailer's reference point are given by

$$\begin{aligned}
x_2 &= x_1 - l_2\cos(\theta - \psi), \\
y_2 &= y_1 - l_2\sin(\theta - \psi).
\end{aligned} \tag{4.42}$$

The control input is represented by $\boldsymbol{u} = [u_1, u_2, u_3]^\top = [J, \omega_1, \omega_2]^\top$, where J is the rate of change of the acceleration and ω_1 is the angular velocity of the tractor, and ω_2 is the angular velocity of the trailer.

4.5.2 Construction of CBFs

Position-based barrier functions are used to construct safe sets for both the tractor and trailer to ensure collision-free maneuvers for the whole system. This type of safety constraint constructs a circular area with a user-defined radius (denoted by the safety distance) that keeps the system away from the unsafe area. We choose reference points $p_i, i = 1, 2$, of two vehicles in the system (4.41) as the critical positions.

Suppose that we know the exact position of the robot at each time step. The position of the obstacles are denoted by $p_k^o = [x_k^o, y_k^o]^\top, k \in \{1, \ldots, N_o\}$, where N_o is the number of obstacles in the workspace. We assume that obstacles are distributed in the workspace such that there exists a collision-free path around the reference trajectory.

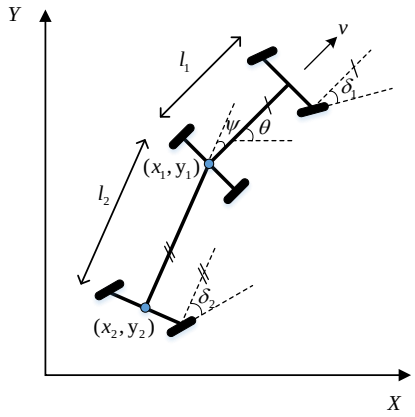


Figure 4.1: A multi-steering tractor-tractor system.

We use multiple CBFs scheme to construct the CBFs for this system. The CBFs $h_1(\boldsymbol{x})$ and $h_2(\boldsymbol{x})$ are the CBFs associated with the tractor and trailer, respectively. Suppose that we have N_o obstacles in the workspace, and for each of them, we design $h_{ik}(\boldsymbol{x})$, $i = 1, 2$, $k \in \{1, \ldots, N_o\}$. These CBFs are obtained based on the position of the bodies' reference points p_i, and the position of each obstacle p_k^o. The safe sets and their associated CBFs for the tractor and trailer are given by

$$\mathcal{C}_{ik} = \{\boldsymbol{x} \in \mathbb{R}^n | h_{ik}(\boldsymbol{x}) \geq 0\}, \tag{4.43}$$

$$h_{ik}(\boldsymbol{x}) = (x_i - x_k^o)^2 + (y_i - y_k^o)^2 - {d_i}^2, \tag{4.44}$$

for all $i = 1, 2$, $k \in \{1, \ldots, N_o\}$, where d_i is a design parameter and denotes the safety distance for each vehicle segment. The CBFs $h_{1k}(\boldsymbol{x})$ and $h_{2k}(\boldsymbol{x})$ have relative degrees three and two, respectively, for all $k \in \{1, \ldots, N_o\}$.

Remark 4.4. *The conventional kinematic model of the MSTTR has input vector* $\boldsymbol{u} = [v, \omega_1, \omega_2]^\top$. *However, considering the designed CBFs, the system does not have a vector relative degree since the decoupling matrix has rank one for this system. We have executed two steps of dynamic extension to achieve the decoupling matrix of rank 2 and vector relative degree* $r_1 = 3, r_2 = 2$ *for the system (4.41).*

We choose linear functions with positive coefficients k_i^j as the set of locally Lipschitz class $\mathcal{K}$ functions $\alpha_i^j(\boldsymbol{x})$, $i = 1, 2$, $j = \{1, \ldots, r_i\}$, which are defined the same for all obstacles

$k \in \{1, \ldots, N_o\}$. Therefore, the constraints in (4.37) can be identically written as

$$\dddot{h}_{1k}(\boldsymbol{x}, \boldsymbol{u}) + K_1[h_{1k}(\boldsymbol{x}), \dot{h}_{1k}(\boldsymbol{x}), \ddot{h}_{1k}(\boldsymbol{x})]^\top \geq 0, \tag{4.45}$$

$$\ddot{h}_{2k}(\boldsymbol{x}, \boldsymbol{u}) + K_2[h_{2k}(\boldsymbol{x}), \dot{h}_{2k}(\boldsymbol{x})]^\top \geq 0, \tag{4.46}$$

where $K_1 = [k_1^0, k_1^1, k_1^2]$, $K_2 = [k_2^0, k_2^1]$. Also, $\dddot{h}_{1k}(\boldsymbol{x}, \boldsymbol{u})$ and $\ddot{h}_{2k}(\boldsymbol{x}, \boldsymbol{u})$ with $k = \{1, \ldots, N_o\}$ are affine in $\boldsymbol{u}$. We refer to the Appendix A for the derivation of (4.45) and (4.46).

4.5.3 Controller design

This section addresses the tracking control design problem of the MSTTR with multiple CBFs. An LTV-MPC approach is utilized for trajectory tracking. Thus, we first derive a linear approximation of the system and then discretize the obtained linearized time-varying system. Next, we formulate the discrete LTV-MPC controller, which focuses on the tracking of the reference trajectory. Then, we enforce the safety constraints (4.45) and (4.46) using the QP optimization introduced in (4.38).

First, we need to linearize the system (4.41) around a given reference trajectory. Assume we are given the reference state $\boldsymbol{x}_r$ and input $\boldsymbol{u}_r \in \mathcal{U}$. We denote the reference state path by a finite sequence of states $X_{ref} = \{\boldsymbol{x}_r(t+i)\}_i$ obtained by applying a finite input sequence $U_{ref} = \{\boldsymbol{u}^r(t+i)\}_i$ (which is given by a planner) to the system (4.41). Then, the linearization of the system described in (4.41) around X_{ref} and U_{ref} results in the following LTV system

$$\dot{\boldsymbol{x}}' = A'(\boldsymbol{z}_t)\boldsymbol{x}' + B'(\boldsymbol{z}_t)\boldsymbol{u}', \tag{4.47}$$

where $\boldsymbol{x}' = \boldsymbol{x} - \boldsymbol{x}_r$ and $\boldsymbol{u}' = \boldsymbol{u} - \boldsymbol{u}_r$ at each time step and $A(\boldsymbol{z}_t) \in \mathbb{R}^{n \times n}$, $B(\boldsymbol{z}_t) \in \mathbb{R}^{n \times m}$, where $\boldsymbol{z}_t = [\boldsymbol{x}_r(t), \boldsymbol{u}_r(t)]^\top$ is a parameter vector that is known at each time step. Then we can discretize (4.47) with sampling time T_s using the forward Euler discretization method. The discretized approximation can be identically written as

$$\boldsymbol{x}_{k+1} = A(\boldsymbol{z}_k)\boldsymbol{x}_k + B(\boldsymbol{z}_k)\boldsymbol{u}_k + C(\boldsymbol{z}_k), \tag{4.48}$$

where $A(\boldsymbol{z}_k) \in \mathbb{R}^{n \times n}, B(\boldsymbol{z}_k) \in \mathbb{R}^{n \times m}$, and $C(\boldsymbol{z}_k) \in \mathbb{R}^n$ are matrices with known elements at current time step k.

$$
\begin{aligned}
\boldsymbol{x}_{k+1} &= A_k \boldsymbol{x}_k + B_k \boldsymbol{u}_k + C_k, \\
A_k &= I + \frac{\partial F}{\partial \boldsymbol{x}}(\boldsymbol{x}^r, \boldsymbol{u}^r) T_s, \\
B_k &= \frac{\partial F}{\partial \boldsymbol{u}}(\boldsymbol{x}^r, \boldsymbol{u}^r) T_s, \\
C_k &= (F(\boldsymbol{x}^r, \boldsymbol{u}^r) - \frac{\partial F}{\partial \boldsymbol{x}}(\boldsymbol{x}^r, \boldsymbol{u}^r)\boldsymbol{x}^r - \frac{\partial F}{\partial \boldsymbol{u}}(\boldsymbol{x}^r, \boldsymbol{u}^r)\boldsymbol{u}^r) T_s.
\end{aligned} \tag{4.49}
$$

We refer to the Appendix A, for a derivation of the system matrices in (4.49).

Discrete-time LTV-MPC formulation

Consider we have a full measurement of the state $\boldsymbol{x}_k$ at the current time step k. The following optimization problem formulates the discrete LTV-MPC design:

$$
\begin{aligned}
\min_{X_t^{opt}, U_t^{opt}} & \|\boldsymbol{x}^r_{t+N|t} - \boldsymbol{x}_{t+N|t}\|_P^2 + \sum_{k=t}^{t+N-1} \|\boldsymbol{x}^r_{k|t} - \boldsymbol{x}_{k|t}\|_Q^2 + \|\boldsymbol{u}^r_{k|t} - \boldsymbol{u}_{k|t}\|_R^2 \\
\text{s.t.} \quad & \boldsymbol{x}_{k+1|t} = A_k \boldsymbol{x}_{k|t} + B_k \boldsymbol{u}_{k|t} + C_k, \quad k = t, \dots, t+N-1, \\
& \boldsymbol{x}_{t|t} = \boldsymbol{x}(t), \\
& \boldsymbol{x}_{k|t} \in \mathcal{X}, \\
& \boldsymbol{u}_{k|t} \in \mathcal{U},
\end{aligned} \tag{4.50}
$$

where $X_t^{opt} = [\boldsymbol{x}_{t|t}, \dots, \boldsymbol{x}_{t+N-1|t}]^\top$ is the vector of optimized states, and the vector $U_t^{opt} = [\boldsymbol{u}_{t|t}, \dots, \boldsymbol{u}_{t+N-1|t}]^\top$ is the vector of optimized inputs in the prediction horizon N. The notation $\boldsymbol{x}_{t+k|t}$ is used to denote the state $\boldsymbol{x}$ at time step $t+k$ predicted at the current time t. The weighting matrices P and Q are positive semidefinite and R is positive definite and penalizes the difference between the final state, state, and the input with their reference values, respectively. The nominal control law is denoted by $\boldsymbol{u}_{nom}$, which is obtained by picking the first element of U_t^{opt}.

4.6 Simulation Results

In this section, simulation studies highlight the effectiveness of the proposed method. A comparison is made between the multiple CBFs scheme and ECBF method. Then, the

Table 4.1: Maximum values for the MSTTR and SSTTR models

MSTTR Parameters		SSTTR Parameters	
v_{max}	$20\ m/s$	v_{max}	$20\ m/s$
a_{max}	$1\ m/s^2$	$\delta_{1_{max}}$	$0.784\ rad$
ψ_{max}	$0.784\,rad$	a_{max}	$1\ m/s^2$
$\delta_{1_{max}}$	$0.784\ rad$	$\omega_{1_{max}}$	$1.5\ rad/s$
$\delta_{2_{max}}$	$0.784\ rad$		
J_{max}	$2.5\ m/s^3$		
$\omega_{1_{max}}$	$1.5\ rad/s$		
$\omega_{2_{max}}$	$0.5\ rad/s$		

accuracy of the proposed control strategy is investigated in a cluttered environment. Simulations are run in the Python environment, and the optimization problems are solved using CVXPY [51] package.

Consider the system described in (4.41), the tractor and trailer wheelbases are $l_1 = 2.5\,m$ and $l_2 = 5.5\,m$, respectively. We have compact sets

$$\mathcal{X} = \{\boldsymbol{x} \in \mathbb{R}^8 : |\boldsymbol{x}_i| \leq \boldsymbol{x}_{i_{max}}\}, i = 1, \ldots, 8, \tag{4.51}$$

$$\mathcal{U} = \{\boldsymbol{u} \in \mathbb{R}^3 : |\boldsymbol{u}_i| \leq \boldsymbol{u}_{i_{max}}\}, i = 1, 2, 3, \tag{4.52}$$

where $\boldsymbol{x}_i$ and $\boldsymbol{u}_i$ are the constrained elements of the state and input vectors, respectively. $\boldsymbol{x}_{i_{max}}$ and $\boldsymbol{u}_{i_{max}}$ are the maximum values which can be obtained from Table 4.1.

The multi-CBFs design parameters are safety distances $d_1 = 4.6\,m$, $d_2 = 3\,m$, and gain vectors $K_1 = \begin{bmatrix}1 & 2 & 2\end{bmatrix}^\top$, $K_2 = \begin{bmatrix}4 & 4\end{bmatrix}^\top$. The LTV-MPC parameters are the prediction horizon $N = 5$, the sampling time $T_s = 0.2\,s$, the weighting matrices $Q = diag(1, 1, 0.5, 0, 0.05, 0.1, 0, 0)$, $P = Q$, and $R = diag(0.01, 0.05, 0.05)$.

4.6.1 Obstacle avoidance in a 90° turning maneuver

The task is to follow a 90° turning maneuver path while avoiding two obstacles. In this simulation, the performance of the multi-CBFs structure application to MSTTR has been compared with the application of ECBF method to the SSTTR (an SSTTR model can be obtained by substituting $\delta_2 = 0$ in (4.41)). As the position-based CBF has relative degree two with respect to SSTTR system, we have applied ECBF method [114] to this system. The ECBF gain vector is $K_1' = \begin{bmatrix}1 & 2\end{bmatrix}^\top$ and the weighting matrices are $Q' =$

$P' = diag(1, 1, 0.5, 0.5, 0.5)$, and $R' = diag(0.01, 0.05)$. Other parameters are the same in both cases. Identical parameters have been used in LTV-MPC and CBF design for both systems.

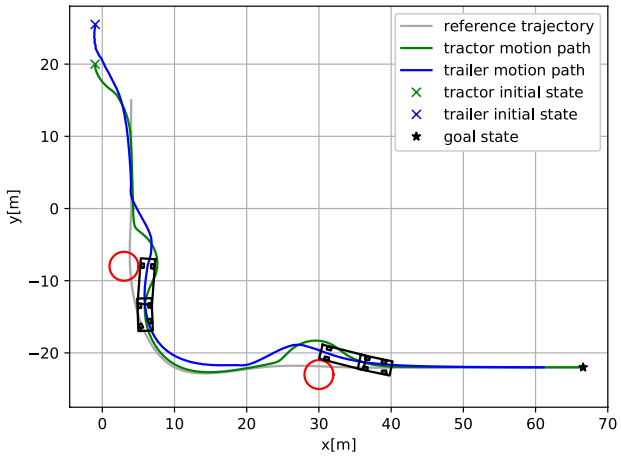


Figure 4.2: SSTTR with ECBF in a 90° turning maneuver in the presence of two obstacles. See this link for the simulation video.

Figure 4.2 and 4.4 illustrate the performance of the ECBF and multiple CBFs, respectively. The transit paths of the tractor and trailer are represented, and the reference trajectory is shown in dark grey. The system started at an arbitrary initial position, where it is not in collision with any obstacle. The results show that the LTV-MPC was able to track the reference trajectory in a short amount of time. In Figures 4.2 and 4.4, the tractor-trailer system is illustrated in two snapshots when it passes each obstacle. As shown in Figure 4.2, the SSTTR with ECBF had a collision with obstacles at the corner of the trailer. In contrast, Figure 4.4 shows that the MSTTR with multiple CBFs scheme successfully passed the obstacles without any collision. The reason is that the multiple CBFs approach has a dedicated CBF for each body. It effectively benefits from the MSTTR's additional input to navigate the trailer such that it also passes the obstacle without any collision. The input signals for the SSTTR and MSTTR systems are illustrated in Figures 4.3 and 4.5, respectively. It can be verified that the input constraints are satisfied in both cases using the LTV-MPC structure. The additional input in the MSTTR (the angular velocity ω_2) has been represented in Figure 4.5, which shows that the multi-CBFs scheme avoids any collision by changing this input at two time instances when the trailer is passing the obstacles. A video of these results is available in this link.

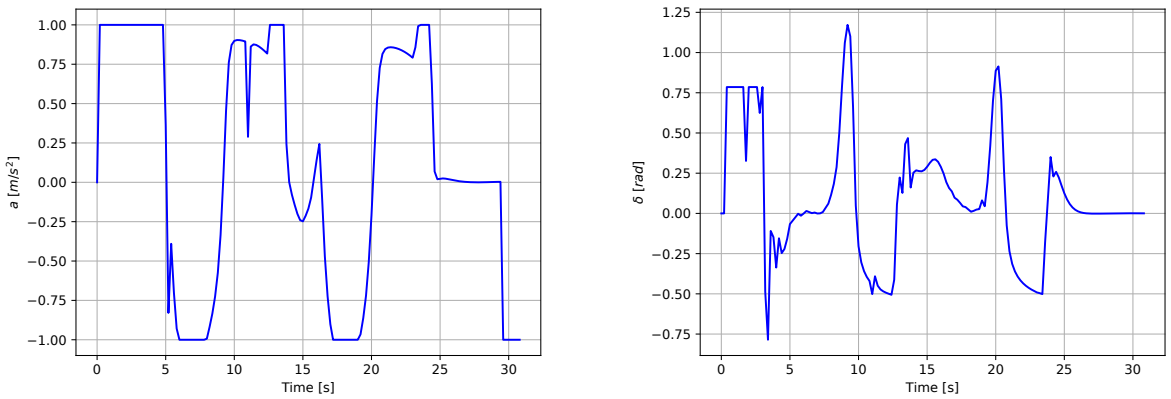


Figure 4.3: The SSTTR system inputs. Left: the tractor acceleration. Right: the tractor steering angle.

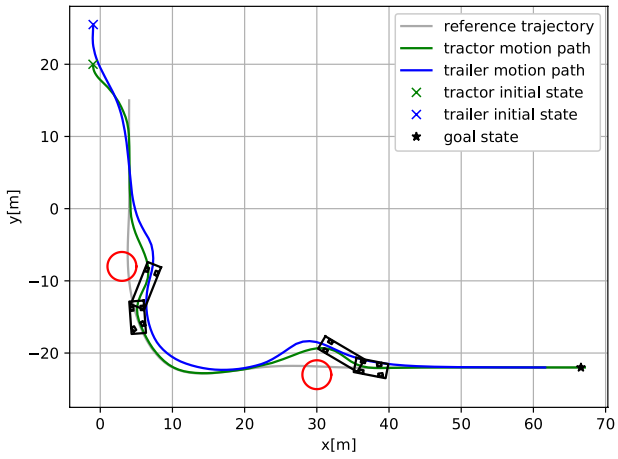


Figure 4.4: MSTTR with multiple CBFs in a 90° turning maneuver in the presence of two obstacles. See this link for the simulation video.

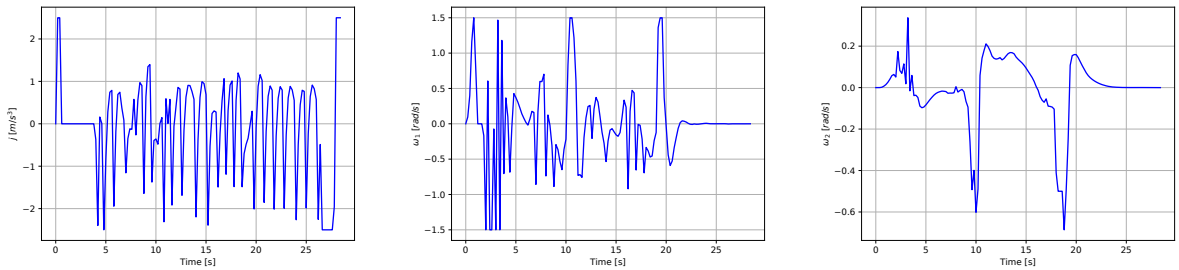


Figure 4.5: The MSTTR system inputs. Left: the tractor jerk signal. Middle: the tractor angular velocity. Right: the trailer angular velocity.

It is important to note that this simulation is specifically designed to illustrate how the multi-CBF structure ensures safety while fully utilizing the system's capabilities. To this end, two types of TTRs are considered, where the multi-CBF approach leverages the additional control input from the MSTTR to facilitate obstacle avoidance for both the tractor and the trailer. In contrast, the ECBF design fails to guarantee safety, as it lacks direct control over the rear axle of the trailer. Ensuring safety for both the tractor and trailer using ECBF alone is only feasible by imposing a large safety threshold. However, this leads to a conservative approach, which becomes problematic in scenarios where the available free space between obstacles is narrow.

4.6.2 Obstacle avoidance in a cluttered workspace

In this section, we investigate the accuracy of the proposed tracking and obstacle avoidance in a complicated scenario. A cluttered workspace is considered where eight obstacles are presented on a reference trajectory. The LTV-MPC and multiple CBFs scheme parameters are the same as previously stated. In order to show that the system passed the obstacles without any collision, a footprint plot is provided in Figure 4.6, which shows the space covered by the whole system (tractor and trailer) during the simulation. The input signals associated with the MSTTR system have also been represented in Figure 4.7.

The evaluation process in this environment can be divided into two phases. In the first phase, the reactive obstacle avoidance of the proposed method is tested. Particularly, the location of the obstacles left a narrow space for the tractor and trailer to pass without collision. However, the safety module could successfully ensure a collision-free pass while effectively enforcing the safe control input via the QP framework. In time instances that the system passes the obstacles, the input ω_2 changes such that the trailer does not have any

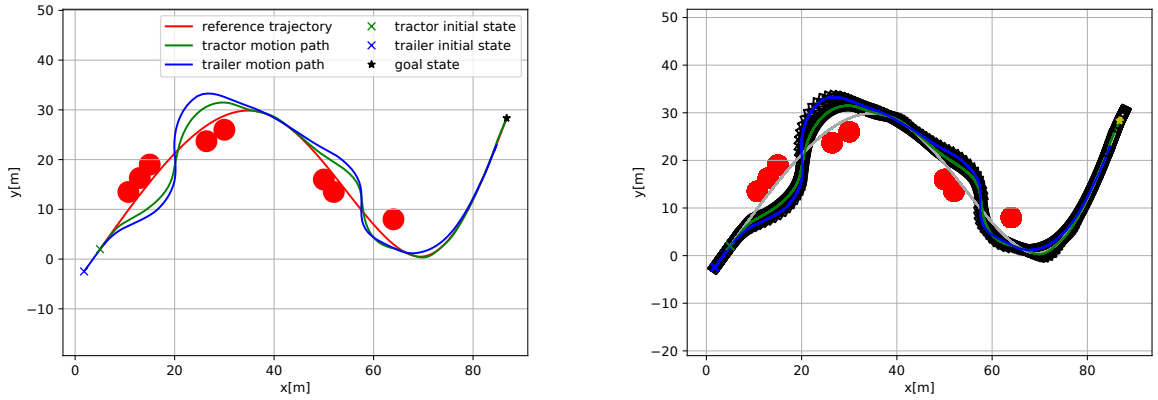


Figure 4.6: MSTTR with multiple CBFs in a cluttered environment. Left: the trajectories for the tractor and trailer. Right: footprint plot that shows the space covered by the whole system (tractor and trailer) during the simulation. See this link for the simulation video.

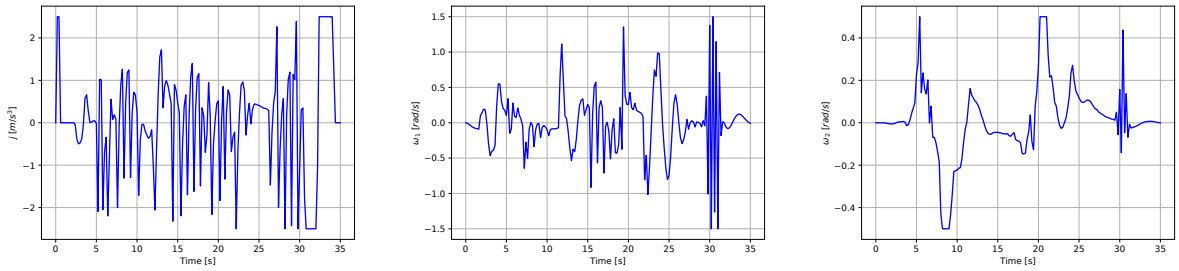


Figure 4.7: The MSTTR system inputs in a cluttered environment simulation. Left: the tractor jerk signal. Middle: the tractor angular velocity. Right: the trailer angular velocity.

collision with the obstacles (see Figure 4.7). In the second phase, the tracking objective is evaluated. As shown in Figure 4.6, when the reference is free of obstacles, both the tractor and trailer accurately follow the reference path, which is illustrated in the final part of the simulation. A video of these results is available in this link.

4.7 Chapter Summary

Robots are usually exposed to various safety hazards in real-world industrial applications. We must consider these hazards' risks in the control design to achieve a fully autonomous robotic system. In this work, we have introduced a novel multiple CBFs structure, which effectively uses the idea of high-order CBFs to ensure the forward invariance of the associated safe sets. A reactive obstacle avoidance technique is designed for MSTTRs based on the proposed multiple CBFs scheme. Simulation results show that the multi-CBFs method ensures collision avoidance for both the tractor and trailer.

Chapter 5

Learning High-Order CBFs for Safety-Critical Control with GP

5.1 Introduction

In recent years, the requirements of modern control applications have extended beyond mere stability assurance to include safety verification. The notion of safety holds such major importance that we deploy an autonomous system in the real world only if it is verified to be safe or achieves a high level of probabilistic safety. CBFs have recently provided a powerful theoretical tool for synthesizing controllers that ensure the safety of dynamical systems. They were initially applied to autonomous driving and bipedal walking [10]. While this method has shown significant promise in addressing safety verification challenges, it had an initial limitation: it was only suitable for systems of relative degree one with respect to the CBF. The introduction of exponential CBFs [114] eliminated this limitation and expanded the application of this approach to systems of arbitrary relative degree. Consequently, the use of this safety method has extended to a wide range of robotic systems. A more general definition of HOCBFs is proposed in [169]. More recently, the assumptions of forward completeness and uniform relative degree have been relaxed for HOCBFs [145].

However, the CBFs method initially advanced on the theoretical assumption that a precise dynamical model of the system is available. In practice, model uncertainty compromises the safety guarantees provided by CBFs. Data-driven methods offer powerful tools to address these challenges with adaptive or probabilistic solutions. Among machine learning techniques, GPs have gained significant attention in control systems. Beyond pre-

diction, GPs provide probabilistic confidence intervals, which are invaluable in data-driven safety-critical control methods where uncertainty quantification is crucial.

Data-driven methods have been extensively used to ensure safety in the presence of model uncertainty with CBFs of relative degree one. In [164], a GP is employed to learn the residual control-independent dynamics for the safe navigation of quadrotors with CBFs. A GP-based systematic controller synthesis method is introduced in [75], offering an inherently safe control strategy. In [54], the authors employ a Bayesian approach to learn model uncertainty while ensuring safety during the learning process. All the aforementioned works assume that the actuation term is known.

Data-driven methods for HOCBFs can be divided into two approaches. The first approach learns the unknown (residual) dynamics and uses the approximated model to derive safety certificates. The impact of the residual dynamics on ECBFs is addressed in [157], where system dynamics are learned using a neural network. Methods employing neural networks must achieve precise approximations of the dynamics to formally ensure safety. In general, this requirement is challenging to meet, but GPs can provide a solution by offering confidence bounds. In [83], completely unknown control-affine system dynamics are approximated with Matrix Variate Gaussian Process (MVGP), resulting in a probabilistic high-order safety constraint. This method ensures that the expectation and variance of the resulting probabilistic safety constraint remain linear and quadratic in the control input, leading to convex optimization. However, approximating the entire system using GPs is computationally expensive. A prevalent assumption, therefore, is to learn each element of the vector fields individually and overlook their correlations.

These limitations lead us to the second approach, which approximates the projection of the residual of system dynamics onto the CBF. In [147], an episodic learning method is employed to approximate the impact of unmodeled dynamics on system safety, eliminating the need for exhaustive data collection. In [33], a GP-based min-norm controller stabilizing an unknown control affine system using control Lyapunov functions is introduced. The effect of the uncertainty in the stability constraint is approximated by a GP model. The same method is applied to a relative degree one CBF design in [34]. This technique results in the approximation of a real-valued function representing the influence of uncertainty on the safety certificate.

In this chapter, we present a data-driven approach based on GP that learns the impact of model uncertainty on high-order safety certificates. We then incorporate the uncertainty-aware high-order safety constraint with an arbitrary robust controller in a min-norm quadratic program. The main contributions of this method relative to prior works are as follows:

- We characterize the impact of uncertainty on the HOCBF. It is proven that with a proper design of HOCBFs, the residual term in the high-order safety certificate is control affine, which facilitates obtaining the GP-based high-order safety constraint.
- We demonstrate that by selecting a particular form of the covariance function in the GP structure, the resulting min-norm QP with an uncertainty-aware safety constraint can be converted into an SOCP.
- We analyze the feasibility of the resulting constrained optimization problem and determine the necessary and sufficient conditions for feasibility.

5.2 Background and Preliminaries

Consider the nonlinear control affine system (3.1) without input constraints (i.e., $\mathcal{U} = \mathbb{R}^m$). Suppose that we have a barrier function $h\colon \mathcal{X} \to \mathbb{R}$ associated with this system for safety purposes which has a high relative degree $r > 1$ with respect to the system (3.1). In Chapter 3, we reviewed the definition of ECBF. Here, we recall a more general approach known as HOCBFs has been developed in [169] and develop our methodology based on this definition.

In this approach, we define a series of continuously differentiable functions $\zeta\colon \mathbb{R}^n \to \mathbb{R}$ as

$$\begin{aligned} \zeta_0(\boldsymbol{x}) &= h(\boldsymbol{x}), \\ \zeta_i(\boldsymbol{x}) &= \dot{\zeta}_{i-1}(\boldsymbol{x}) + \alpha_i(\zeta_{i-1}(\boldsymbol{x})),\ i \in \{1, \dots, r\}, \end{aligned} \tag{5.1}$$

where $\alpha_i(\cdot)$, $i \in \{1, \dots, r\}$ are class $\mathcal{K}$ functions. We also define their zero superlevel sets $\mathcal{Z}_i$ and their interior sets for $i \in \{1, \dots, r\}$ as

$$\mathcal{Z}_i = \{\boldsymbol{x} \in \mathbb{R}^n \mid \zeta_{i-1}(\boldsymbol{x}) \geq 0\}, \tag{5.2}$$

$$Int(\mathcal{Z}_i) = \{\boldsymbol{x} \in \mathbb{R}^n \mid \zeta_{i-1}(\boldsymbol{x}) > 0\}. \tag{5.3}$$

Definition 5.1. *Let the functions $\zeta_i(\boldsymbol{x})$ and sets $Int(\mathcal{Z}_i)$ be defined by (5.1) and (5.3), respectively. The r^{th} order continuously differentiable function $h(\boldsymbol{x})$ with relative degree $r > 1$ is called an HOCBF if h and its derivatives up to order r, are locally Lipschitz continuous, and there exists a set of sufficiently smooth class $\mathcal{K}$ functions α_i, such that*

$$\sup_{u \in \mathbb{R}^m} [L_f^r h + L_g L_f^{r-1} h\, \boldsymbol{u} + \sum_{i=1}^{r-1} L_f^i(\alpha_{r-i} \circ \zeta_{r-i-1}) + \alpha_r(\zeta_{r-1})] \geq 0, \tag{5.4}$$

for all $\boldsymbol{x} \in \bigcap_{i=1}^{r} Int(\mathcal{Z}_i)$*, where the dependence on* $\boldsymbol{x}$ *is removed for simplicity.*

Note that (5.4) is obtained by substituting the expression $\zeta_0(\boldsymbol{x}) = h(\boldsymbol{x})$ into the recursive formula (5.1) r times. Therefore, it is equivalent to $\zeta_r(\boldsymbol{x}, \boldsymbol{u}) \geq 0$, which will be used in the remainder of the chapter.

Proposition 5.1. *Given system (3.1) with a HOCBF* h *with relative degree* r*, any Lipschitz continuous control* $\boldsymbol{u}$ *that satisfies (5.4) renders the set* $\bigcap_{i=1}^{r} Int(\mathcal{Z}_i)$ *forward invariant for system (3.1).*

Proof. A proof of this result can be found in Chapter 4 of this thesis. □

Inequality (5.4) imposes an affine condition on the control values which can be used to ensure safety. Consider a nominal control policy $\boldsymbol{u}_{nom}$, Lipschitz continuous in $\boldsymbol{x}$, designed primarily to fulfill control objectives. Our goal is to apply this control law to the system only if it complies with (5.4). As indicated in Section 3.3.4, this is accomplished by solving the following quadratic programming optimization:

$$\begin{aligned} \boldsymbol{u}_s = \underset{\boldsymbol{u} \in \mathcal{R}^m}{\arg\min}\ & \|\boldsymbol{u} - \boldsymbol{u}_{nom}\|_2^2 \\ \text{s.t.}\quad & \zeta_r(\boldsymbol{x}, \boldsymbol{u}) \geq 0. \end{aligned} \tag{5.5}$$

The resulting minimally invasive point-wise controller prioritizes the safety over control objectives by satisfying $\zeta_r(\boldsymbol{x}, \boldsymbol{u}) \geq 0$ as a hard constraint.

5.3 Impact of Model Uncertainty on HOCBFs

In this section, we focus on reformulating (5.4) such that it takes the model uncertainty into account. We refer to the system (3.1) as the true model, which is partially unknown, and consider a nominal model

$$\dot{\boldsymbol{x}} = \hat{f}(\boldsymbol{x}) + \hat{g}(\boldsymbol{x})\boldsymbol{u}, \tag{5.6}$$

where $\hat{f}\colon \mathbb{R}^n \to \mathbb{R}^n$ and $\hat{g}\colon \mathbb{R}^n \to \mathbb{R}^{n \times m}$ are known functions, locally Lipschitz in $\boldsymbol{x}$. We design a HOCBF based on the known nominal model (5.6). To apply it to the true system, we make the following assumption.

Assumption 5.1. *If an* r^{th} *order differentiable function* $h\colon \mathbb{R}^n \to \mathbb{R}$ *is an HOCBF for the nominal model (5.6), it will be also a valid HOCBF for the true system (3.1).*

Intuitively, this assumption requires the true and nominal model to share the same degree of actuation, which is met by feedback linearizable systems [82].

Due to the inherent model mismatch between the true system and the nominal model, the Lie derivatives of $h(\boldsymbol{x})$ (of order r) calculated based on the nominal model differ from their true values. We represent the resulting error by

$$\Delta_i(\boldsymbol{x}) = L_f^i h(\boldsymbol{x}) - L_{\hat{f}}^i h(\boldsymbol{x}), i \in \{1, \dots, r\}, \tag{5.7}$$

$$\Delta_g(\boldsymbol{x}) = L_g L_f^{r-1} h(\boldsymbol{x}) - L_{\hat{g}} L_{\hat{f}}^{r-1} h(\boldsymbol{x}). \tag{5.8}$$

Similarly, the errors in the time derivatives of h up to order $r-1$ are denoted by $\Delta_i(\boldsymbol{x})$, where $i \in \{1, \dots, r-1\}$, while the error in the r^{th} order time derivative is obtained by $\Delta_r(\boldsymbol{x}) + \Delta_g(\boldsymbol{x})\,\boldsymbol{u}$.

Consequently, these discrepancies propagate into the auxiliary functions (5.1), potentially compromising the safety of the system. In the proposed method, our goal is to approximate the adverse effect of uncertainty on the high-order safety certificate (5.4) by adopting a supervised learning approach. This approach makes our HOCBF design robust to model uncertainty. To achieve this, we must separate the propagated error in (5.4) from the approximations based on the nominal model. This goal depends on the design of the HOCBF, which is addressed in the following result.

Theorem 5.1. *The true auxiliary functions in the recursive formula (5.1) can be written as $\zeta_i = \hat{\zeta}_i + R_i(\Delta_i, \dots, \Delta_1)$, for $i \in \{1, \dots, r-1\}$, where $\hat{\zeta}_i$ calculated based on the nominal model (5.6) and $R_i(\Delta_i, \dots, \Delta_1)$ is a linear function of its argument, if and only if class $\mathcal{K}$ functions $\alpha_i(x) = k_i x$, where $k_i > 0$. Furthermore, $R_i(\Delta_i, \dots, \Delta_1), i \in \{1, \dots, r-1\}$ is obtained by*

$$\begin{aligned} R_i(\Delta_i, \dots, \Delta_1) = \Delta_i + &\sum_{1 \le j_1 \le i} k_{j_1} \Delta_{i-1} + \\ &\sum_{1 \le j_1 < j_2 \le i} k_{j_1} k_{j_2} \Delta_{i-2} + \cdots + \\ &\sum_{1 \le j_1 < \cdots < j_{i-1} \le i} k_{j_1} k_{j_2} \dots k_{j_{i-1}} \Delta_1. \end{aligned} \tag{5.9}$$

Proof. For the forward direction, we are given that $\zeta_i = \hat{\zeta}_i + R_i$, for $i \in \{1, \dots, r-1\}$.

Based on this equation and (5.1), we have

$$\begin{aligned}\zeta_i &= \frac{\partial}{\partial t}(\hat{\zeta}_{i-1} + R_{i-1}) + \alpha_i(\hat{\zeta}_{i-1} + R_{i-1}) \\ &= \frac{\partial}{\partial t}\hat{\zeta}_{i-1} + \frac{\partial}{\partial t}R_{i-1} + \alpha_i(\hat{\zeta}_{i-1} + R_{i-1}) \\ &= \hat{\zeta}_i + \underbrace{\frac{\partial}{\partial t}R_{i-1} + \alpha_i(\hat{\zeta}_{i-1} + R_{i-1}) - \alpha_i(\hat{\zeta}_{i-1})}_{=R_i}.\end{aligned}$$

Note that for the time derivative of residuals in (5.7), we have the following by the definition of high-order Lie derivatives

$$\frac{\partial}{\partial t}\Delta_i = \frac{\partial}{\partial t}L_f^i h - \frac{\partial}{\partial t}L_{\hat{f}}^i h = L_f^{i+1}h - L_{\hat{f}}^{i+1}h = \Delta_{i+1},$$

for $i \in \{1, \ldots, r-2\}$. Thus, taking time derivative of R_{i-1} and multiplying by k_i, we have $R_i = \frac{\partial}{\partial t}R_{i-1} + k_i R_{i-1}$. Substituting this equation into the resulting expression for R_i, yields

$$\alpha_i(\hat{\zeta}_{i-1} + R_{i-1}) - \alpha_i(\hat{\zeta}_{i-1}) = k_i R_{i-1},$$

where we are looking for functions α_i. This functional equation can be solved if the class $\mathcal{K}$ functions α_i satisfy Cauchy's additive functional equation

$$\alpha_i(x+y) = \alpha_i(x) + \alpha_i(y), \tag{5.10}$$

for all x, y in their domain. Next, we will show that this equation only holds for $\alpha_i = k_i x$. Since α_i satisfy (5.10), by induction and simple calculation we can show that $\alpha_i(qx) = q\alpha_i(x)$, for all $q \in \mathbb{Q}, q > 0$, where $\mathbb{Q}$ is the set of all rational numbers. Since α_i is strictly increasing the following limit exists

$$\lim_{x \to 0^+} \alpha_i(x) = \lim_{n \to +\infty} \alpha_i(\frac{1}{n}) = \lim_{n \to +\infty} \frac{1}{n}\alpha_i(1) = 0.$$

Then, pick any x from the domain and let $\{q_n\} \in \mathbb{Q}$ be any decreasing sequence converging to x. From the above results, we have that $\alpha_i(q_n) = q_n\alpha_i(1) = x\alpha_i(1)$. On the other hand, from (5.10), we can write $\alpha_i(q_n) = \alpha_i(x) + \alpha_i(q_n - x)$. We know that $q_n - x \to 0^+$ as $n \to +\infty$, thus $\alpha_i(q_n) = \alpha_i(x)$. Thus, we can conclude $\alpha_i(x) = \alpha_i(1)x = k_i x$.

For the reverse direction, suppose that we have $\alpha_i(x) = k_i x$, for $i \in \{1, \ldots, r\}$. We use mathematical induction to prove that $\zeta_i = \hat{\zeta}_i + R_i(\Delta_i, \ldots, \Delta_1)$, where R_i is given by (5.9). We start from $i = 1$:

$$\zeta_1 = \dot{\zeta}_0 + \alpha_1(\zeta_0) = \dot{h} + k_1 h = \dot{\hat{h}} + \Delta_1 + k_1 h = \hat{\zeta}_1 + \Delta_1.$$

Thus, $R_1(\Delta_1) = \Delta_1$. Now, assume that $\zeta_p = \hat{\zeta}_p + R_p(\Delta_p, \ldots, \Delta_1)$ holds for $i = p$, where $R_p(\Delta_p, \ldots, \Delta_1)$ is given by (5.9). Then for $i = p+1$, from (5.1) we have

$$\begin{aligned}\zeta_{p+1} &= \dot{\zeta}_p + \alpha_{p+1}(\zeta_p) = \frac{\partial}{\partial t}\left(\hat{\zeta}_p + R_p\right) + k_{p+1}\left(\hat{\zeta}_p + R_p\right) \\ &= \hat{\zeta}_{p+1} + \frac{\partial}{\partial t}R_p + k_{p+1}R_p.\end{aligned}$$

Using the obtained relationship for the time derivative of residuals, we have

$$\begin{aligned}\zeta_{p+1} &= \hat{\zeta}_{p+1} + \frac{\partial}{\partial t}R_p + k_{p+1}R_p \\ &= \hat{\zeta}_{p+1} + \Delta_{p+1} + \sum_{1\le j_1\le p} k_{j_1}\Delta_p + \cdots + \\ &\quad \sum_{1\le j_1<\cdots<j_{p-1}\le p} k_{j_1}k_{j_2}\ldots k_{j_{p-1}}\Delta_2 \\ &\quad + k_{p+1}(\Delta_p + \sum_{1\le j_1\le p} k_{j_1}\Delta_{p-1} + \cdots + \\ &\quad \sum_{1\le j_1<\cdots<j_{p-1}\le p} k_{j_1}k_{j_2}\ldots k_{j_{p-1}}\Delta_1).\end{aligned} \tag{5.11}$$

By factorization, we have

$$\begin{aligned}\zeta_{p+1} = \hat{\zeta}_{p+1} + \Delta_{p+1} + \sum_{1\le j_1\le p+1} k_{j_1}\Delta_p + \cdots + \\ \sum_{1\le j_1<\cdots<j_p\le p} k_{j_1}k_{j_2}\ldots k_{j_p}\Delta_1,\end{aligned}$$

which proves our claim for $i = p+1$. □

5.4 GP-based HOCBF Design

Theorem 5.1 shows that we can approximate the effect of uncertainty on the high-order safety certificate by supervised learning only if we choose linear class $\mathcal{K}$ functions in our HOCBF design. It also sets the basis for reducing the complexity of our proposed learning method.

By substituting the results from Theorem 5.1 in (5.1) recursively and using (5.7) and (5.8), we can rewrite $\zeta_r(\boldsymbol{x}, \boldsymbol{u})$ as

$$\zeta_r(\boldsymbol{x}, \boldsymbol{u}) = \hat{\zeta}_r(\boldsymbol{x}, \boldsymbol{u}) + \underbrace{R_r(\Delta_r, \ldots, \Delta_1) + \Delta_g(\boldsymbol{x})\boldsymbol{u}}_{=\Delta_{hocbf}(\boldsymbol{x}, \boldsymbol{u})},$$

where $\Delta_i \colon \mathbb{R}^n \to \mathbb{R}, i \in \{1, \ldots, r\}$ and $\Delta_g \colon \mathbb{R}^n \to \mathbb{R}^{1\times m}$, which gives us the high-order safety certificate of the form

$$\hat{\zeta}_r(\boldsymbol{x}, \boldsymbol{u}) + \Delta_{hocbf}(\boldsymbol{x}, \boldsymbol{u}) \geq 0. \tag{5.12}$$

The result in (5.12) enables us to compensate for the adverse effect of the uncertainty in the safety of the true system by learning the residual term $\Delta_{hocbf}(\boldsymbol{x}, \boldsymbol{u})$.

Remark 5.1. *In many data-driven safety-critical control methods, the target functions are the unknown dynamics f, g in (3.1). However, Δ_{hocbf} is a scalar value. This fact substantially reduces the complexity of the learning process by addressing a lower-dimensional problem compared to the latter approach.*

Substituting $i = r$ in (5.9) and expanding $\Delta_{hocbf}(\boldsymbol{x}, \boldsymbol{u})$ yields

$$\Delta_{hocbf}(\boldsymbol{x}, \boldsymbol{u}) = \gamma_r\Delta_r + \gamma_{r-1}\Delta_{r-1} + \cdots + \gamma_1\Delta_1 + \Delta_g(\boldsymbol{x})\boldsymbol{u}, \tag{5.13}$$

where we denote coefficients of Δ_i's in (5.9) by $\gamma_i, i \in \{1, \ldots, r\}, \gamma_r = 1$, to simplify notations. We can approximately measure Δ_i using finite difference methods and collect a dataset for our learning method:

$$\begin{aligned}
\tilde{\Delta}_j &= FD^j(\boldsymbol{x}(t), \Delta t) - \hat{h}^j(\boldsymbol{x}(t)), j \in \{1, \ldots, r-1\}, \\
\tilde{\Delta}_r &= FD^r(\boldsymbol{x}(t), \boldsymbol{u}(t), \Delta t) - \hat{h}^r(\boldsymbol{x}(t), \boldsymbol{u}(t)),
\end{aligned} \tag{5.14}$$

where Δt is the sampling interval. Then, the approximated measurement can be derived by substituting (5.14) into (5.13).

5.4.1 GP for uncertainty-aware HOCBF

Going back to the original problem, our objective is to approximate Δ_{hocbf} using GP. From (5.13), we can verify that Δ_{hocbf} is control-affine and can be characterized by

$$\Delta_{hocbf} = \begin{bmatrix}\Delta_1 & \ldots & \Delta_r & \Delta_g\end{bmatrix} \begin{bmatrix}\boldsymbol{\gamma} \\ \boldsymbol{u}\end{bmatrix} \coloneqq \begin{bmatrix}\Delta_1 & \ldots & \Delta_r & \Delta_g\end{bmatrix} \boldsymbol{y},$$

where $\boldsymbol{\gamma} = \begin{bmatrix}\gamma_1 & \dots & \gamma_r\end{bmatrix}^\top \in \mathbb{R}^r$, also we denote the concatenation of $\boldsymbol{\gamma}, \boldsymbol{u}$ by $\boldsymbol{y} \in \mathbb{R}^{m+r}$. This prior knowledge of the problem can be embedded into the structure of our kernel function, which facilitates a more accurate regression.

Proposition 5.2. *Let $\boldsymbol{x} \in \mathbb{R}^n$, $\boldsymbol{y} \in \mathbb{R}^{m+r}$, and define the input domain $\mathcal{X} = \mathbb{R}^{n+m+r}$. Then consider the real-valued function $k_c : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$, defined by*

$$k_c\left(\begin{bmatrix}\boldsymbol{x}\\ \boldsymbol{y}\end{bmatrix}, \begin{bmatrix}\boldsymbol{x}'\\ \boldsymbol{y}'\end{bmatrix}\right) = \boldsymbol{y}^\top \Lambda(\boldsymbol{x}, \boldsymbol{x}')\boldsymbol{y}', \tag{5.15}$$

where $\Lambda(\boldsymbol{x}, \boldsymbol{x}') = \operatorname{diag}(\begin{bmatrix}k_1(\boldsymbol{x}, \boldsymbol{x}'), \dots, k_{m+r}(\boldsymbol{x}, \boldsymbol{x}')\end{bmatrix})$ and $k_i : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}, i \in \{1, \dots, m + r\}$. If k_i's are positive-definite kernels, then k_c is a positive-definite kernel.

Proof. Since $k_i(\boldsymbol{x}, \boldsymbol{x}')$'s are positive definite (valid) kernels, then by the definition, there is a feature map $\varphi(\boldsymbol{x})$ such that $k_i(\boldsymbol{x}, \boldsymbol{x}') = \varphi_i^\top(\boldsymbol{x})\varphi_i(\boldsymbol{x}')$ for $i \in \{1, \dots, m + r\}$. Now, Let y^i and y'^i be the i^{th} element of the corresponding vectors. We have

$$\begin{aligned}
k_c &= \boldsymbol{y}^\top \operatorname{diag}\left([\varphi_1^\top(\boldsymbol{x})\varphi_1(\boldsymbol{x}'), \dots, \varphi_{m+r}^\top(\boldsymbol{x})\varphi_{m+r}(\boldsymbol{x}')]\right)\boldsymbol{y}' \\
&= \sum_{i=1}^{m+r} y^i \varphi_i^\top(\boldsymbol{x})\varphi_i(\boldsymbol{x}')y'^i \\
&= \begin{bmatrix}y^1\varphi_1^\top(\boldsymbol{x}) & \dots & y^{m+r}\varphi_{m+r}^\top(\boldsymbol{x})\end{bmatrix} \begin{bmatrix}y'^1\varphi_1(\boldsymbol{x}')\\ \vdots \\ y'^{m+r}\varphi_{m+r}(\boldsymbol{x}')\end{bmatrix} \\
&= \psi^\top\left(\begin{bmatrix}\boldsymbol{x}\\ \boldsymbol{y}\end{bmatrix}\right)\psi\left(\begin{bmatrix}\boldsymbol{x}'\\ \boldsymbol{y}'\end{bmatrix}\right),
\end{aligned}$$

which again by the definition of kernels, proves that $k_c(\cdot, \cdot)$ is a positive-definite (valid) kernel. □

An immediate consequence of Proposition 5.2 is that k_c is the reproducing kernel of a Reproducing Kernel Hilbert Space (RKHS) $\mathcal{H}_{k_c}(\mathcal{X})$.

A dataset can be generated by collecting trajectories of the true system. We denote it by $\mathcal{D} = \{((\boldsymbol{x}_j, \boldsymbol{y}_j), z_j)\}_{j=1}^N$, where $(\boldsymbol{x}_j, \boldsymbol{y}_j) \in \mathcal{X} \subset \mathbb{R}^n \times \mathbb{R}^{m+r}$ is the input data and outputs z_j are obtained from (5.13) and (5.14). Also, we define $X = \begin{bmatrix}\boldsymbol{x}_1, \dots, \boldsymbol{x}_N\end{bmatrix} \in \mathbb{R}^{n\times N}$, and $Y = \begin{bmatrix}\boldsymbol{y}_1, \dots, \boldsymbol{y}_N\end{bmatrix} \in \mathbb{R}^{(m+r)\times N}$. Based on the collected dataset, the GP model which is

equipped with the kernels of the form k_c in Proposition 5.2, gives the following expression for the posterior distribution of a test point $(\boldsymbol{x}_*, \boldsymbol{y}_*)$:

$$m(\boldsymbol{x}_*, \boldsymbol{y}_*) = \boldsymbol{z}^\top \left(K_c + \sigma_n^2 I\right)^{-1} \bar{K}^\top \boldsymbol{y}_*, \tag{5.16}$$

$$\sigma^2(\boldsymbol{x}_*, \boldsymbol{y}_*) = \boldsymbol{y}_*^\top \left(\Lambda(\boldsymbol{x}_*, \boldsymbol{x}_*) - \bar{K} \left(K_c + \sigma_n^2 I\right)^{-1} \bar{K}^\top\right) \boldsymbol{y}_*, \tag{5.17}$$

where K_c is the Gram matrix of $k_c(\cdot, \cdot)$ for the input data pair (X, Y), and $\bar{K} \in \mathbb{R}^{(m+r)\times N}$ is given by

$$\bar{K} = \begin{bmatrix} \bar{\boldsymbol{k}}_{\boldsymbol{1}} & \bar{\boldsymbol{k}}_{\boldsymbol{2}} & \dots & \bar{\boldsymbol{k}}_N \end{bmatrix} \circ Y,$$
$$\bar{\boldsymbol{k}}_{\boldsymbol{i}} = \begin{bmatrix} k_1(x_*, x_i) & \dots & k_{m+r}(x_*, x_i) \end{bmatrix}^\top, i \in \{1, \dots, N\},$$

where $\circ$ denotes the element-wise multiplication (Hadamard product) of two matrices with identical dimensions. This result highlights a key advantage of using k_c kernel. The resulting expression for the posterior mean $m(\boldsymbol{x}_*, \boldsymbol{y}_*) = \boldsymbol{\mu}(\boldsymbol{x}_*)\boldsymbol{y}_*$ and the variance $\sigma^2(\boldsymbol{x}_*, \boldsymbol{y}_*) = \boldsymbol{y}_*^\top \Sigma(\boldsymbol{x}_*)\boldsymbol{y}_*$ are linear and quadratic in $\boldsymbol{y}_*$ (and the control input), respectively. We will exploit this feature in the next section to establish a convex GP-based safety filter based on HOCBFs.

5.4.2 Confidence bounds for the estimation of the effect of uncertainty

Although GP is inherently a probabilistic model, a high probability error bound can be derived for the distance between the true value and the GP prediction. This requires an additional assumption on Δ_{hocbf}.

Assumption 5.2. *Each $\Delta_i, i \in \{1, \dots, r\}$ and each i^{th} element of Δ_g for $i \in \{1, \dots, m\}$ is a member of $\mathcal{H}_{k_i}$ with bounded RKHS norm $\|\Delta_i\|_{k_i} \leq \eta$ for $i \in \{1, \dots, m+r\}$.*

Lemma 5.1. *([141])*

Let Assumption 5.2 hold. Then, with a probability of at least $1-\delta$, the following holds

$$|m(\boldsymbol{x}, \boldsymbol{y}) - \Delta_{hocbf}(\boldsymbol{x}, \boldsymbol{u})| \leq \beta\sigma(\boldsymbol{x}, \boldsymbol{y}), \tag{5.18}$$

on a compact set $\mathcal{D} \subset \mathcal{X}, \delta \in (0, 1)$ and $\beta = \sqrt{2\eta^2 + 300\kappa_{N+1}\log^3((N+1)/\delta)}$, where κ_{N+1} is the maximum mutual information that can be obtained after getting $N+1$ data, and η is the upper bound of the corresponding RKHS norm, and $m(\boldsymbol{x}, \boldsymbol{y})$ and $\sigma(\boldsymbol{x}, \boldsymbol{y})$ are the posterior mean and standard deviation of a test point $(\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{X}$.

Based on the probabilistic bounds on (5.18) and the high-order safety certificate (5.12), we can conclude that the following hold with a probability of at least $1-\delta$

$$\zeta_r(\boldsymbol{x},\boldsymbol{u}) \geq \hat{\zeta}_r(\boldsymbol{x},\boldsymbol{u}) + m(\boldsymbol{x},\boldsymbol{y}) - \beta\sigma(\boldsymbol{x},\boldsymbol{y}), \tag{5.19}$$

where $m(\boldsymbol{x},\boldsymbol{y})$ and $\sigma(\boldsymbol{x},\boldsymbol{y})$ are the mean and standard deviation of a query point derived from (5.16), (5.17). Incorporating (5.19) into the optimization problem (5.5), we have

$$\begin{aligned} \boldsymbol{u}_s = \underset{\boldsymbol{u}\in\mathcal{R}^m}{\arg\min}\ & \|\boldsymbol{u}-\boldsymbol{u}_{nom}\|_2^2 \\ \text{s.t.}\quad & \hat{\zeta}_r(\boldsymbol{x},\boldsymbol{u}) + m(\boldsymbol{x},\boldsymbol{y}) - \beta\sigma(\boldsymbol{x},\boldsymbol{y}) \geq 0. \end{aligned} \tag{5.20}$$

Note that the constraint in (5.20) is constructed regardless of the underlying true dynamics.

Theorem 5.2. *The optimization problem (5.20) is convex and can be converted into the standard second-order cone program of the form (5.21), if $m(\boldsymbol{x},\boldsymbol{y})$ and $\sigma(\boldsymbol{x},\boldsymbol{y})$ satisfy (5.16) and (5.17).*

$$\begin{aligned} \begin{bmatrix}\boldsymbol{u}_{socp}^\top & t_*\end{bmatrix}^\top = \underset{\boldsymbol{\omega}}{\arg\min}\ & \boldsymbol{f}^\top\boldsymbol{\omega} \\ s.t.\ \ & \|M_1\boldsymbol{\omega}+\boldsymbol{n}_1\|_2 \leq \boldsymbol{p}_1^\top\boldsymbol{\omega}, \\ & \|M_2\boldsymbol{\omega}+\boldsymbol{n}_2\|_2 \leq \boldsymbol{p}_2^\top\boldsymbol{\omega}+q_2, \end{aligned} \tag{5.21}$$

where $\boldsymbol{\omega} = \begin{bmatrix}\boldsymbol{u}^\top & t\end{bmatrix}^\top \in \mathbb{R}^{m+1}$ *and* $\boldsymbol{f} = \begin{bmatrix}\boldsymbol{0}_m^\top & 1\end{bmatrix} \in \mathbb{R}^{m+1}$, *and*

$$\begin{aligned} &M_1 = \begin{bmatrix}I_m & \boldsymbol{0}_m\end{bmatrix}, \boldsymbol{n}_1 = -\boldsymbol{u}_{nom}, \boldsymbol{p}_1 = \begin{bmatrix}\boldsymbol{0}_m^\top & 1\end{bmatrix}^\top, \\ &M_2 = \begin{bmatrix}\beta L^m & \boldsymbol{0}_{m+r}\end{bmatrix}, \boldsymbol{n}_2 = \beta L^r\boldsymbol{\gamma}, \boldsymbol{p}_2 = \begin{bmatrix}\hat{\boldsymbol{\zeta}}_g+\boldsymbol{\mu}_m & 0\end{bmatrix}^\top, q_2 = (\hat{\boldsymbol{\zeta}}_f+\boldsymbol{\mu}_r)\boldsymbol{\gamma}, \end{aligned}$$

have appropriate dimensions.

Proof. Let's denote the objective function of (5.20) by $J = \|\boldsymbol{u}-\boldsymbol{u}_{nom}\|_2^2$. Since the Euclidean norm is always positive, we can consider the equivalent objective function $J_1 = \|\boldsymbol{u}-\boldsymbol{u}_{nom}\|_2$. We set $\|\boldsymbol{u}-\boldsymbol{u}_{nom}\|_2 \leq t$, where $t\in\mathbb{R}$ is an auxiliary variable and convert it to a second-order cone constraint. Now, we need to solve a new minimization problem with a new augmented variable $\boldsymbol{\omega} = \begin{bmatrix}\boldsymbol{u}^\top & t\end{bmatrix}^\top \in \mathbb{R}^{m+1}$ as

$$\begin{aligned} &\min \begin{bmatrix}\boldsymbol{0}_m^\top & 1\end{bmatrix}\boldsymbol{\omega} \\ &\text{s.t. } \| \underbrace{\begin{bmatrix}I_m & \boldsymbol{0}_m\end{bmatrix}}_{:=M_1}\boldsymbol{\omega} \underbrace{-\boldsymbol{u}_{nom}}_{:=\boldsymbol{n}_1} \| \leqslant \underbrace{\begin{bmatrix}\boldsymbol{0}_m^\top & 1\end{bmatrix}}_{:=\boldsymbol{p}_1^\top}\boldsymbol{\omega}, \end{aligned} \tag{5.22}$$

where I_m is a $m \times m$ identity matrix and $\mathbf{0}_m \in \mathbb{R}^m$ is a vector of zeros, and $\boldsymbol{f} = \begin{bmatrix}\mathbf{0}_m^\top & 1\end{bmatrix}^\top$.

Next, we need to show that the constraint in (5.20) is a second-order cone constraint. Note that based on (5.16), we have $m(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{\mu}(\boldsymbol{x})\boldsymbol{y}$, which can be written as

$$m(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{\mu}_r \boldsymbol{\gamma} + \boldsymbol{\mu}_m \boldsymbol{u},$$

where the notation $\boldsymbol{\mu}_r$ refers the first r elements, and $\boldsymbol{\mu}_m$ refers the last m elements of the row vector $\boldsymbol{\mu} \in \mathbb{R}^{1\times(m+r)}$. Also, from (5.4), we know that $\hat{\zeta}_r$ is control affine, i.e. $\hat{\zeta}_r = \hat{\boldsymbol{\zeta}}_f \boldsymbol{\gamma} + \hat{\boldsymbol{\zeta}}_g \boldsymbol{u}$, where the row vectors $\hat{\boldsymbol{\zeta}}_f \in \mathbb{R}^{1\times r}$ and $\hat{\boldsymbol{\zeta}}_g \in \mathbb{R}^{1\times m}$. Thus, the first two terms of the inequality in (5.20) are affine in $\boldsymbol{u}$ as desired.

Based on (5.17), we have that $\sigma^2(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{y}^\top \Sigma(\boldsymbol{x}) \boldsymbol{y}$, where $\Sigma \in \mathbb{R}^{(m+r)\times(m+r)}$. Since k_c is a valid kernel and $\sigma_n > 0$, Σ is positive definite. Then, we have

$$\sigma(\boldsymbol{x}, \boldsymbol{y}) = \sqrt{\boldsymbol{y}^\top \Sigma \boldsymbol{y}} = \sqrt{\boldsymbol{y}^\top L^\top L \boldsymbol{y}} = \|L\boldsymbol{y}\|_2,$$

where $L \in \mathbb{R}^{(m+r)\times(m+r)}$ is the matrix square root of Σ. By definition $\boldsymbol{y} = \begin{bmatrix}\boldsymbol{\gamma}^\top & \boldsymbol{u}^\top\end{bmatrix}^\top$, thus we have $\sigma(\boldsymbol{x}, \boldsymbol{y}) = \|L^r \boldsymbol{\gamma} + L^m \boldsymbol{u}\|_2$, where the notation L^r and L^m refers to the first r columns, and the last m columns of the matrix L, respectively. Now, we can rewrite the high-order safety certificate (5.20) as a second-order cone constraint of the form

$$\|A(\boldsymbol{x})\boldsymbol{u} + \boldsymbol{b}(\boldsymbol{x})\|_2 \leq \boldsymbol{c}(\boldsymbol{x})\boldsymbol{u} + d(\boldsymbol{x}), \tag{5.23}$$

where

$$\begin{aligned} A(\boldsymbol{x}) &= \beta L^m \in \mathbb{R}^{(m+r)\times m}, \\ \boldsymbol{b}(\boldsymbol{x}) &= \beta L^r \boldsymbol{\gamma} \in \mathbb{R}^{m+r}, \\ \boldsymbol{c}(\boldsymbol{x}) &= \hat{\boldsymbol{\zeta}}_g + \boldsymbol{\mu}_m \in \mathbb{R}^{1\times m}, \\ d(\boldsymbol{x}) &= (\hat{\boldsymbol{\zeta}}_f + \boldsymbol{\mu}_r)\boldsymbol{\gamma} \in \mathbb{R}. \end{aligned}$$

It can be expressed in terms of the new variable $\boldsymbol{\omega}$ as

$$\| \underbrace{\begin{bmatrix}A(\boldsymbol{x}) & \mathbf{0}_{m+r}\end{bmatrix}}_{:=M_2} \boldsymbol{\omega} + \underbrace{\boldsymbol{b}(\boldsymbol{x})}_{:=\boldsymbol{n}_2} \|_2 \leq \underbrace{\begin{bmatrix}\boldsymbol{c}(\boldsymbol{x}) & 0\end{bmatrix}}_{:=\boldsymbol{p}_2^\top} \boldsymbol{\omega} + \underbrace{d(\boldsymbol{x})}_{:=q_2}.$$

Adding the resulting constraint to (5.22) leads to (5.21), which concludes the proof. □

5.4.3 Feasibility analysis

In this section, we will analyze the point-wise feasibility of the SOCP (5.20). Intuitively, if the GP prediction is less accurate, the resulting standard deviation $\sigma(\boldsymbol{x}, \boldsymbol{y})$ will be large, leading to a more conservative approach. In particular, it restricts the space on which a safe control input can be selected. In consequence, it may make the SOCP infeasible. In the following, we theoretically analyze the conditions for feasibility.

Theorem 5.3. *Given a point $\boldsymbol{x} \in \mathbb{R}^n$ and a HOCBF design with the coefficient vector $\boldsymbol{\gamma} \in \mathbb{R}^r$, the SOCP (5.20) is feasible, if and only if there exists a control input $\boldsymbol{u} \in \mathbb{R}^m$ such that it satisfies the following conditions*

$$\begin{bmatrix} \hat{\boldsymbol{\zeta}}_f + \boldsymbol{\mu}_r & \boldsymbol{c}(\boldsymbol{x}) \end{bmatrix} \begin{bmatrix} \boldsymbol{\gamma} \\ \boldsymbol{u} \end{bmatrix} \geq 0, \tag{5.24}$$

$$\begin{bmatrix} \boldsymbol{\gamma}^\top & \boldsymbol{u}^\top \end{bmatrix} S(\boldsymbol{x}) \begin{bmatrix} \boldsymbol{\gamma} \\ \boldsymbol{u} \end{bmatrix} \leq 0, \tag{5.25}$$

where $S \in \mathbb{R}^{(m+r)\times(m+r)}$ is of the form

$$\begin{aligned} S(\boldsymbol{x}) &= \begin{bmatrix} S_1(\boldsymbol{x}) & S_2(\boldsymbol{x}) \\ S_2^\top(\boldsymbol{x}) & S_3(\boldsymbol{x}) \end{bmatrix}, \\ S_1(\boldsymbol{x}) &= \beta^2 {L^r}^\top L^r - (\hat{\boldsymbol{\zeta}}_f + \boldsymbol{\mu}_r)^\top (\hat{\boldsymbol{\zeta}}_f + \boldsymbol{\mu}_r), \\ S_2(\boldsymbol{x}) &= \beta {L^r}^\top A(\boldsymbol{x}) - (\hat{\boldsymbol{\zeta}}_f + \boldsymbol{\mu}_r)^\top \boldsymbol{c}(\boldsymbol{x}), \\ S_3(\boldsymbol{x}) &= A^\top(\boldsymbol{x}) A(\boldsymbol{x}) - \boldsymbol{c}^\top(\boldsymbol{x}) \boldsymbol{c}(\boldsymbol{x}). \end{aligned} \tag{5.26}$$

Proof. In order to analyze the feasibility of the SOCP (5.20), we need to verify that the safety second-order cone constraint (5.23) is feasible. Since the left-hand side is always non-negative, the right-hand side must be also non-negative, which leads to the first condition.

Now, since both sides of (5.23) are non-negative, we can square both sides of the equation and then gather all the terms on the left-hand side, Factorizing the similar terms gives the second condition. □

Based on this result, we obtain the necessary condition for point-wise feasibility as stated in the following:

Lemma 5.2. *Given $\boldsymbol{x} \in \mathbb{R}^n$, if the SOCP (5.20) is feasible, then the following condition must be satisfied*

$$1 - \frac{1}{\beta^2} \boldsymbol{\phi} \Sigma^{-1} \boldsymbol{\phi}^\top \leq 0, \tag{5.27}$$

where $\boldsymbol{\phi} = \begin{bmatrix} \hat{\boldsymbol{\zeta}}_f + \boldsymbol{\mu}_r & \boldsymbol{c}(\boldsymbol{x}) \end{bmatrix} \in \mathbb{R}^{1\times(m+r)}$.

Proof. We prove this by contradiction. Let's assume that there exists a solution $\boldsymbol{u} \in \mathbb{R}^m$ of the SOCP (5.20), which satisfies $1 - \frac{1}{\beta^2}\boldsymbol{\phi}\Sigma^{-1}\boldsymbol{\phi}^\top > 0$. Let's define the matrix M partitioned as

$$M = \begin{bmatrix} 1 & \boldsymbol{\phi} \\ \boldsymbol{\phi}^\top & \beta^2\Sigma(\boldsymbol{x}) \end{bmatrix} \in \mathbb{R}^{(m+r+1)\times(m+r+1)}.$$

The Schur complements of M is obtained by

$$\begin{cases} M/1 = \beta^2\Sigma(\boldsymbol{x}) - \boldsymbol{\phi}^\top\boldsymbol{\phi}, \\ M/\beta^2\Sigma(\boldsymbol{x}) = 1 - \frac{1}{\beta^2}\boldsymbol{\phi}\Sigma^{-1}\boldsymbol{\phi}^\top. \end{cases}$$

We can easily verify that $\beta^2\Sigma = \begin{bmatrix} \beta L^r & A(\boldsymbol{x}) \end{bmatrix}^\top \begin{bmatrix} \beta L^r & A(\boldsymbol{x}) \end{bmatrix}$ and $M/1 = S(\boldsymbol{x})$. We know that $\Sigma(\boldsymbol{x})$ is Symmetric Positive Definite (SPD). So, M is also symmetric and we can characterize its definiteness by the Schur complement theorem. Since its lower right block is positive definite and its corresponding Schur complement $M/\beta^2\Sigma(\boldsymbol{x}) > 0$ by our assumption, we can conclude that M is also positive definite.

Now, given that M is SPD, and its upper left block is always positive definite, we can conclude that $M/1 = S(\boldsymbol{x})$ must be positive definite, which is a contradiction to the condition (5.25). Thus, $1 - \frac{1}{\beta^2}\boldsymbol{\phi}\Sigma^{-1}\boldsymbol{\phi}^\top \leq 0$ holds. □

Theorem 5.3 also provides the theoretical background to obtain a sufficient condition of feasibility.

Lemma 5.3. *Given $\boldsymbol{x} \in \mathbb{R}^n$, the SOCP (5.20) is feasible at $\boldsymbol{x}$, if $S_3(\boldsymbol{x})$ is negative definite.*

Proof. From (5.26), we know that $S_3(\boldsymbol{x})$ is a symmetric matrix, thus it has real eigenvalues. Let's denote its maximum eigenvalue and the corresponding eigenvector by λ_m, $\boldsymbol{e}_m$. Then, by the assumption that $S_3(\boldsymbol{x})$ is negative definite, we can conclude that $\lambda_m < 0$ and $\boldsymbol{e}_m^\top S_3(\boldsymbol{x})\boldsymbol{e}_m < 0$. Substituting (5.26) into the former, we have

$$\boldsymbol{e}_m^\top A^\top(\boldsymbol{x})A(\boldsymbol{x})\boldsymbol{e}_m - (\boldsymbol{c}(\boldsymbol{x})\boldsymbol{e}_m)^\top(\boldsymbol{c}(\boldsymbol{x})\boldsymbol{e}_m) < 0.$$

Since $A(\boldsymbol{x})^\top A(\boldsymbol{x})$ is positive definite, $\boldsymbol{c}(\boldsymbol{x})\boldsymbol{e}_m \neq 0$ must hold. We take a control input in the direction of this eigenvector as $\boldsymbol{u}_e = \alpha\,\mathrm{sgn}(\boldsymbol{c}(\boldsymbol{x})\boldsymbol{e}_m)\cdot\boldsymbol{e}_m, \alpha > 0$. Next, we need to check if the resulting control input can satisfy the conditions of Theorem 5.3. Substituting $\boldsymbol{u}_e$ in (5.25), we have

$$\boldsymbol{\gamma}^\top S_1\boldsymbol{\gamma} + 2\alpha\gamma^\top S_2((\boldsymbol{c}(\boldsymbol{x})\boldsymbol{e}_m)\cdot\boldsymbol{e}_m) + \alpha^2\boldsymbol{e}_m^\top S_3(\boldsymbol{x})\boldsymbol{e}_m.$$

By choosing large enough α, the above statement can be made negative since $\boldsymbol{e}_m^\top S_3(\boldsymbol{x})\boldsymbol{e}_m < 0$. Also, substituting $\boldsymbol{u}_e$ into (5.24), we have

$$\boldsymbol{c}(\boldsymbol{x})\boldsymbol{u}_e + (\hat{\boldsymbol{\zeta}}_f + \boldsymbol{\mu}_r)\boldsymbol{\gamma} = \alpha|\boldsymbol{c}(\boldsymbol{x})\boldsymbol{e}_m| + (\hat{\boldsymbol{\zeta}}_f + \boldsymbol{\mu}_r)\boldsymbol{\gamma}.$$

Again, by choosing a large enough α, we can make the above expression positive, which concludes the proof. □

5.5 Bayesian Learning of System Dynamics

In this section, we solve the problem of ensuring safety for systems with uncertainty from another perspective. In [83, 50], a Bayesian approach is employed to learn the unknown system dynamics. Then, the learned dynamics are integrated into the safety filter based on CBFs. We review the proposed Bayesian learning method with the goal of comparing this approach with our design. This comparison will help us to highlight the pros and cons of each approach.

5.5.1 Problem formulation

We assume the same control-affine system as defined in (3.1) and consider the safe set (3.28) along with its associated CBF design. Initially, we focus on a relative degree one barrier function $h(\boldsymbol{x})$ with respect to the system. Extension to the HOCBF will be addressed after developing the method for the base case. For notational convenience, we rewrite the system (3.1) in the following form

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x}) + g(\boldsymbol{x})\boldsymbol{u} = \underbrace{\begin{bmatrix} f(\boldsymbol{x}) & g(\boldsymbol{x}) \end{bmatrix} \begin{bmatrix} 1 \\ \boldsymbol{u} \end{bmatrix}}_{:=\mathcal{F}(\boldsymbol{x})\bar{\boldsymbol{u}}}, \tag{5.28}$$

where the state and input vectors, together with drift and input gain vector fields are defined as in (3.1). Additionally, we introduce $\text{vec}(\mathcal{F}(\boldsymbol{x})) \in \mathbb{R}^{n(m+1)}$, which is a vector obtained by stacking the columns of the matrix $\mathcal{F}(\boldsymbol{x})$.

Similar to Section 5.3, we assume that f and g are unknown, but the system states and input measurements are available, enabling us to collect a training set of the form $X_{1:k} := [\boldsymbol{x}(t_1), \ldots, \boldsymbol{x}(t_k)]$, $U_{1:k} := [\boldsymbol{u}(t_1), \ldots, \boldsymbol{u}(t_k)]$, and $\dot{X}_{1:k} = [\dot{\boldsymbol{x}}(t_1), \ldots, \dot{\boldsymbol{x}}(t_k)]$. The objective is to approximate the matrix $\mathcal{F} : \mathbb{R}^n \to \mathbb{R}^{n\times(m+1)}$ using GP regression. The

first step is to design a GP structure for $\text{vec}(\mathcal{F}(\mathbf{x}))$ and initialize it with a mean function $\text{vec}(M_0(\mathbf{x}))$ and covariance function $K_0(\mathbf{x}, \mathbf{x}')$. Using the prior distribution of the unknown dynamics and the collected training data, we compute the posterior distribution $\mathcal{GP}(\text{vec}(M_k(\mathbf{x})), K_k(\mathbf{x}, \mathbf{x}'))$ of $\text{vec}(\mathcal{F}(\mathbf{x}))$ conditioned on the training set $(X_{1:k}, U_{1:k}, \dot{X}_{1:k})$.

We aim to integrate this approximation into our CBF-based safety filter design. Suppose the system starts within the safe set $\mathcal{C} = \{\boldsymbol{x} \in \mathbb{R}^n \mid h(\boldsymbol{x}) \geq 0\}$ (i.e. $\mathbf{x}_k := \mathbf{x}(t_k) \in \mathcal{C}$). Given the distribution $\mathcal{GP}(\text{vec}(M_k(\mathbf{x})), K_k(\mathbf{x}, \mathbf{x}'))$ of $\text{vec}(\mathcal{F}(\mathbf{x}))$ at time t_k, the objective is to find the safe control law $\boldsymbol{u}_s$ by solving the following optimization

$$\begin{aligned} \boldsymbol{u}_s = \underset{\boldsymbol{u} \in \mathcal{R}^m}{\arg\min} \ & \|\boldsymbol{u} - \boldsymbol{u}_{nom}\|_2^2 \\ \text{s.t.} \quad & \Pr\left(L_f h(\boldsymbol{x}) + L_g h(\boldsymbol{x})\boldsymbol{u} + \alpha(h(\boldsymbol{x})) \geq 0\right) \geq p_k, \end{aligned} \tag{5.29}$$

where $p_k \in (0, 1)$ is an adjustable risk tolerance parameter. The structure of this optimization is the same as (5.5). However, the safety constraint is converted to a chance constraint due to the GP approximation of $\mathcal{F}$.

There are different approaches to infer a posterior over $\mathcal{F}(\boldsymbol{x})$ and convert the chance constraint into a deterministic one. In the following, we will introduce MVGP to develop the Bayesian safety filter design.

5.5.2 MVGP-based HOCBF design

MVGP regression is an extension of the standard GP regression, designed to handle multi-dimensional outputs more efficiently by leveraging the inherent correlations between different output dimensions. In many applications, it is crucial to account for the dependencies between the outputs rather than treating them independently. In MVGP regression, instead of modeling each output separately, the outputs are represented as a matrix, and a matrix-variate distribution is applied to capture the correlations across both input dimensions and output dimensions. This approach generalizes the GP framework by introducing a covariance structure not only for the input space but also for the output space.

In the problem we want to solve, we aim to approximate the unknown drift f and input gain g using the training dataset $(X_{1:k}, U_{1:k}, \dot{X}_{1:k})$. To simplify the derivation, we assume that the state and control samples $X_{1:k}, U_{1:k}$, are collected without noise, but the measurements $\dot{X}_{1:k}$ are corrupted with zero-mean Gaussian noise $\mathcal{N}(\mathbf{0}, \Sigma)$ which is independent across time steps. Consider the matrix-valued function $\mathcal{F}(\boldsymbol{x})$ in (5.28). Knowing the fact that each measurement $\dot{\boldsymbol{x}}(t_k)$ is obtained by the control-affine combination of $f(\boldsymbol{x}) + g(\boldsymbol{x})\boldsymbol{u}$ directs us to integrate this affine structure into the learning process. This approach not

only increases the learning efficiency but also ensures that the safety condition associated with CBF remains linear in $\boldsymbol{u}$. Kojasteh et al. [83] proposed a factorization of the covariance matrix $K_k(\mathbf{x}, \mathbf{x}')$ that efficiently models the correlation among the elements of $\mathcal{F}(\boldsymbol{x})$ and preserves the control-affine structure in the posterior of the GP distribution. In the following, we review a definition of Matrix Variate Gaussian (MVG) distribution and list some of its properties, which are important for the derivation of the results.

Definition 5.2. *The MVG distribution is a three-parameter distribution $\mathcal{MN}(M, A, B)$ describing a random matrix $X \in \mathbb{R}^{n\times m}$ with probability density function:*

$$Pr(X; M, A, B) := \frac{\exp\left(-\frac{1}{2} tr\left[B^{-1}(X-M)^\top A^{-1}(X-M)\right]\right)}{(2\pi)^{nm/2}\det(A)^{m/2}\det(B)^{n/2}}, \tag{5.30}$$

where $M \in \mathbb{R}^{n\times m}$ is the mean of the distribution, and the invertible matrices $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{m\times m}$ encode the covariance matrix of the rows and columns of X, respectively.

Although the matrix variate Gaussian is a straightforward generalization of the multivariate case, it offers a simple and efficient way to separate the correlations between the rows and columns of the matrix, which implicitly affects the correlations among the input and output hidden units.

Lemma 5.4. *([69]) Let X follow a MVG distribution $\mathcal{MN}(M, A, B)$. Then,*

$$\mathrm{vec}(X) \sim \mathcal{N}\left(\mathrm{vec}(M), B \otimes A\right), \tag{5.31}$$

where $\mathrm{vec}(M)$ is the column-wise vectorization of the matrix M and $\otimes$ is the standard Kronecker product.

Lemma 5.5. *Let X follow a MVG distribution $\mathcal{MN}(M, A, B)$ and let $C \in \mathbb{R}^{d\times n}$ and $D \in \mathbb{R}^{m\times d}$. Then,*

$$CX \sim \mathcal{MN}(CM, CAC^\top, B), \quad XD \sim \mathcal{MN}(MD, A, D^\top BD), \tag{5.32}$$

and

$$cov(\mathrm{vec}(CX), \mathrm{vec}(X)) = B \otimes CA \qquad cov(\mathrm{vec}(XD), \mathrm{vec}(X)) = D^\top B \otimes A \tag{5.33}$$

Proof. For the first part, a proof is provided in [69]. For the second part, a proof based on [83] is provided here. Let $Y := X - M$, such that $Y \sim \mathcal{MVG}(0, A, B)$. Let $Y = [\mathbf{y}_1, \ldots, \mathbf{y}_m]$ where $\mathbf{y}_i \in \mathbb{R}^n$ are column vectors of Y.

$$E\left[\mathrm{vec}(Y)\,\mathrm{vec}(Y)^\top\right] = E\left[\begin{bmatrix} \mathbf{y}_1\mathbf{y}_1^\top & \cdots & \mathbf{y}_1\mathbf{y}_m^\top \\ \vdots & & \vdots \\ \mathbf{y}_m\mathbf{y}_1^\top & \cdots & \mathbf{y}_m\mathbf{y}_m^\top \end{bmatrix}\right] = B \otimes A = \begin{bmatrix} b_{11}A & \cdots & b_{1m}A \\ \vdots & & \vdots \\ b_{m1}A & \cdots & b_{mm}A \end{bmatrix}$$

Note that $CY = [C\mathbf{y}_1, \ldots, C\mathbf{y}_m]$.

$$\operatorname{cov}(\operatorname{vec}(CY), \operatorname{vec}(Y)) = E\left[\operatorname{vec}(CY)\operatorname{vec}(Y)^\top\right] = E\left[\begin{bmatrix} C\mathbf{y}_1\mathbf{y}_1^\top & \ldots & C\mathbf{y}_1\mathbf{y}_m^\top \\ \vdots & & \vdots \\ C\mathbf{y}_m\mathbf{y}_1^\top & \ldots & C\mathbf{y}_m\mathbf{y}_m^\top \end{bmatrix}\right]$$
$$= \begin{bmatrix} b_{11}CA & \ldots & b_{1m}CA \\ \vdots & & \vdots \\ b_{m1}CA & \ldots & b_{mm}CA \end{bmatrix} = B \otimes CA.$$

The second part can be proved by noticing that,

$$\begin{aligned} X \sim \mathcal{MVG}(M, A, B) &\implies X^\top \sim \mathcal{MVG}(M, B, A) \\ &\implies \operatorname{cov}(\operatorname{vec}(D^\top X^\top), \operatorname{vec}(X^\top)) = A \otimes D^\top B \\ &\implies \operatorname{cov}(\operatorname{vec}(XD), \operatorname{vec}(X)) = D^\top B \otimes A. \end{aligned}$$

□

Lemma 5.4 indicates that an MVG distribution is a structured Gaussian distribution with a covariance matrix of the form $V \otimes U$ while Lemma 5.5 reveals the distribution properties of a linearly transformed MVG random variable.

Based on Lemma 5.4, the following GP parametrization can be proposed for $\operatorname{vec}(\mathcal{F}(\boldsymbol{x}))$

$$\operatorname{vec}(\mathcal{F}(\mathbf{x})) \sim \mathcal{GP}(\operatorname{vec}(M_0(\mathbf{x})), B_0(\mathbf{x}, \mathbf{x}') \otimes A), \tag{5.34}$$

The above parametrization is more efficient compared to learning the full covariance $K_0(.,.) \in \mathbb{R}^{n(m+1)\times(m+1)n}$, as it requires learning smaller matrices $B_0(\mathbf{x}, \mathbf{x}') \in \mathbb{R}^{(m+1)\times(m+1)}$ and $A \in \mathbb{R}^{n\times n}$. Another significant advantage of this approach is that it preserves the structure after inference. However, this requires the following assumption: the measurement noise covariance matrix must take the form $\sigma^2 A$, where σ is a positive coefficient. Under this assumption, the GP posterior maintains the Kronecker product structure. Consider the training set $(X_{1:k}, U_{1:k}, \dot{X}_{1:k})$, and a query point $\boldsymbol{x}_*$. The train and test data are jointly Gaussian, thus the mean and variance of the posterior $\operatorname{vec}(\mathcal{F}(\boldsymbol{x}_*))$ can be obtained as below. To simplify notation, let $\boldsymbol{\mathcal{B}}_{1:k}^{1:k} \in \mathbb{R}^{k(m+1)\times k(m+1)}$ be a matrix with $k \times k$ block elements $\left[\boldsymbol{\mathcal{B}}_{1:k}^{1:k}\right]_{i,j} \triangleq B_0(\boldsymbol{x}_i, \boldsymbol{x}_j)$ and define $\boldsymbol{\mathcal{M}}_{1:k} \triangleq \begin{bmatrix} M_0(\boldsymbol{x}_1) & \cdots & M_0(\boldsymbol{x}_k) \end{bmatrix} \in \mathbb{R}^{n\times k(m+1)}$, $\boldsymbol{\mathcal{B}}_{1:k}(\boldsymbol{x}) \triangleq \begin{bmatrix} B_0(\boldsymbol{x}, \boldsymbol{x}_1), \ldots, B_0(\boldsymbol{x}, \boldsymbol{x}_k) \end{bmatrix} \in \mathbb{R}^{(m+1)\times k(m+1)}$ and $\bar{\boldsymbol{\mathcal{U}}}_{1:k} \triangleq \operatorname{diag}(\bar{\boldsymbol{u}}_1, \ldots, \bar{\boldsymbol{u}}_k) \in \mathbb{R}^{k(m+1)\times k}$.

$$\begin{bmatrix} \operatorname{vec}(\dot{X}_{1:k}) \\ \operatorname{vec}(\mathcal{F}(\boldsymbol{x}_*)) \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \operatorname{vec}\left(\mathcal{M}_{1:k}\mathcal{U}_{1:k}\right) \\ \operatorname{vec}\left(M_0\left(\boldsymbol{x}_*\right)\right) \end{bmatrix}, \begin{bmatrix} \bar{\mathcal{U}}_{1:k}^\top \mathcal{B}_{1:k}^{1:k} \bar{\mathcal{U}}_{1:k} + \sigma^2 \boldsymbol{I}_k & \bar{\mathcal{U}}_{1:k}^\top \mathcal{B}_{1:k}^\top\left(\boldsymbol{x}_*\right) \\ \mathcal{B}_{1:k}\left(\boldsymbol{x}_*\right) \bar{\mathcal{U}}_{1:k} & B_0\left(\boldsymbol{x}_*, \boldsymbol{x}_*\right) \end{bmatrix} \otimes \boldsymbol{A} \right), \tag{5.35}$$

where

$$\bar{\mathcal{U}}_{1:k}^\top \mathcal{B}_{1:k}^{1:k} \bar{\mathcal{U}}_{1:k} = \begin{bmatrix} \bar{\boldsymbol{u}}_1^\top B_0\left(\boldsymbol{x}_1, \boldsymbol{x}_1\right) \bar{\boldsymbol{u}}_1 & \cdots & \bar{\boldsymbol{u}}_1^\top B_0\left(\boldsymbol{x}_1, \boldsymbol{x}_k\right) \bar{\boldsymbol{u}}_k \\ \vdots & \ddots & \vdots \\ \bar{\boldsymbol{u}}_k^\top B_0\left(\boldsymbol{x}_k, \boldsymbol{x}_1\right) \bar{\boldsymbol{u}}_1 & \cdots & \bar{\boldsymbol{u}}_k^\top B_0\left(\boldsymbol{x}_k, \boldsymbol{x}_k\right) \bar{\boldsymbol{u}}_k \end{bmatrix},$$
$$\mathcal{B}_{1:k}\left(\boldsymbol{x}_*\right) \bar{\mathcal{U}}_{1:k} = \begin{bmatrix} B_0\left(\boldsymbol{x}_1, \boldsymbol{x}_*\right) \bar{\boldsymbol{u}}_1 & \cdots & B_0\left(\boldsymbol{x}_k, \boldsymbol{x}_*\right) \bar{\boldsymbol{u}}_k \end{bmatrix}.$$

This formulation enables us to obtain the posterior mean and covariance function, which is provided below.

Proposition 5.3. *([83]) Consider the training set $(X_{1:k}, U_{1:k}, \dot{X}_{1:k})$ obtained from the trajectories of the system (5.28). The posterior distribution of $\operatorname{vec}(\mathcal{F}(\boldsymbol{x}))$ conditioned on the training set is a Gaussian process $\mathcal{GP}(\operatorname{vec}(M_k(\mathbf{x}_*)), B_k(\mathbf{x}_*, \mathbf{x}'_*) \otimes A)$ where*

$$M_k(\mathbf{x}_*) := M_0(\mathbf{x}_*) + \left(\dot{X}_{1:k} - \mathcal{M}_{1:k}\bar{\mathcal{U}}_{1:k}\right) \left(\bar{\mathcal{U}}_{1:k}^\top B_0(X_{1:k}, X_{1:k})\bar{\mathcal{U}}_{1:k}\right)^{-1} \bar{\mathcal{U}}_{1:k}^\top B_0(X_{1:k}, \mathbf{x}_*),$$
$$B_k(\mathbf{x}_*, \mathbf{x}'_*) := B_0(\mathbf{x}_*, \mathbf{x}'_*) + B_0(\mathbf{x}_*, X_{1:k})\bar{\mathcal{U}}_{1:k} \left(\bar{\mathcal{U}}_{1:k}^\top B_0(X_{1:k}, X_{1:k})\bar{\mathcal{U}}_{1:k}\right)^{-1} \bar{\mathcal{U}}_{1:k}^\top B_0(X_{1:k}, \mathbf{x}'_*),$$

are the mean and covariance functions, respectively.

Note that in the following posterior distribution, the mean function is independent of the A matrix and the covariance function $K_k(\boldsymbol{x}, \boldsymbol{x}')$ is obtained by the Kronecker product $B_k(\mathbf{x}, \mathbf{x}') \otimes A$. We use the properties of the MVG distribution as stated in Lemma 5.5 to obtain the MVGP posterior for a given control input $\boldsymbol{u}$ as below

$$\mathcal{F}(\mathbf{x})\bar{\mathbf{u}} = f(\mathbf{x}) + g(\mathbf{x})\mathbf{u} \sim \mathcal{GP}(M_k(\mathbf{x})\bar{\mathbf{u}}, \bar{\mathbf{u}}^\top B_k(\mathbf{x}, \mathbf{x}')\bar{\mathbf{u}} \otimes A). \tag{5.36}$$

Lemma 5.6. *Consider the dynamics in (5.28) with the posterior distribution (5.36). Given $\mathbf{x}_k$ and $\mathbf{u}_k$, the CBF constraint is*

$$\underbrace{L_f h(\boldsymbol{x}_k) + L_g h(\boldsymbol{x}_k)\boldsymbol{u}_k + \alpha(h(\boldsymbol{x}_k))}_{:=\mathfrak{C}(\boldsymbol{x}_k, \boldsymbol{u}_k)} \geq 0,$$

which is a Gaussian random variable with the mean and variance

$$\mathbb{E}[\mathfrak{C}(\boldsymbol{x}_k, \boldsymbol{u}_k)] = \nabla_{\mathbf{x}} h(\mathbf{x}_k)^\top M_k(\mathbf{x}_k)\bar{\mathbf{u}}_k + \alpha(h(\mathbf{x}_k)), \tag{5.37}$$
$$\operatorname{Var}[\mathfrak{C}(\boldsymbol{x}_k, \boldsymbol{u}_k)] = \left(\bar{\mathbf{u}}_k^\top B_k(\mathbf{x}_k, \mathbf{x}_k)\bar{\mathbf{u}}_k\right) \left(\nabla_{\mathbf{x}} h(\mathbf{x}_k)^\top A \nabla_{\mathbf{x}} h(\mathbf{x}_k)\right). \tag{5.38}$$

Proof. We start by rewriting the CBF constraint as

$$\begin{aligned}\mathfrak{C}(\boldsymbol{x}_k, \boldsymbol{u}_k) &= L_f h(\boldsymbol{x}_k) + L_g h(\boldsymbol{x}_k)\boldsymbol{u}_k + \alpha(h(\boldsymbol{x}_k)) \\ &= \nabla_{\mathbf{x}} h(\mathbf{x}_k)^\top \begin{bmatrix} f(\mathbf{x}_k) & g(\mathbf{x}_k) \end{bmatrix} \begin{bmatrix} 1 \\ \mathbf{u}_k \end{bmatrix} + \alpha(h(\mathbf{x}_k)).\end{aligned}$$

Note that from the posterior distribution, we have the mean and variance of $\mathcal{F}(\boldsymbol{x}_k)\bar{\boldsymbol{u}}_k$. Thus, conditioned on $\mathbf{x}_k$ and $\mathbf{u}_k$ we have

$$\begin{aligned}\mathbb{E}\left[\mathfrak{C}(\boldsymbol{x}_k, \boldsymbol{u}_k)\right] &= \nabla_{\boldsymbol{x}} h\left(\boldsymbol{x}_k\right)^\top \mathbb{E}\left[\mathcal{F}\left(\boldsymbol{x}_k\right) \bar{\boldsymbol{u}}_k\right] + \alpha\left(h\left(\boldsymbol{x}_k\right)\right), \\ \operatorname{Var}\left[\mathfrak{C}(\boldsymbol{x}_k, \boldsymbol{u}_k)\right] &= \nabla_{\boldsymbol{x}} h\left(\boldsymbol{x}_k\right)^\top \operatorname{Var}\left[\mathcal{F}\left(\boldsymbol{x}_k\right) \bar{\boldsymbol{u}}_k\right] \nabla_{\boldsymbol{x}} h\left(\boldsymbol{x}_k\right).\end{aligned}$$

The mean is obtained by substituting the expression in (5.36). For the variance, one extra step is to use the Kronecker product definition to simplify the resulting expression which yields (5.37). □

We want to calculate the probability that $\mathfrak{C}(\boldsymbol{x}_k, \boldsymbol{u}_k)$ is positive. To do this, we standardize the random variable

$$Z = \frac{\mathfrak{C} - \mathbb{E}[\mathfrak{C}(\boldsymbol{x}_k, \boldsymbol{u}_k)]}{\sqrt{\operatorname{Var}\left[\mathfrak{C}(\boldsymbol{x}_k, \boldsymbol{u}_k)\right]}} \sim \mathcal{N}(0, 1),$$

where Z follows a standard normal distribution

$$\Pr(\mathfrak{C}(\boldsymbol{x}_k, \boldsymbol{u}_k) \geq 0) = \Pr\left(Z \geq \frac{-\mathbb{E}[\mathfrak{C}(\boldsymbol{x}_k, \boldsymbol{u}_k)]}{\sqrt{\operatorname{Var}\left[\mathfrak{C}(\boldsymbol{x}_k, \boldsymbol{u}_k)\right]}}\right).$$

Using the symmetry property of the Gaussian distribution[1], the chance constraint in (5.29) can be written as

$$\Pr(\mathfrak{C}(\boldsymbol{x}_k, \boldsymbol{u}_k) \geq 0) = 1 - \Phi\left(\frac{-\mathbb{E}[\mathfrak{C}(\boldsymbol{x}_k, \boldsymbol{u}_k)]}{\sqrt{\operatorname{Var}\left[\mathfrak{C}(\boldsymbol{x}_k, \boldsymbol{u}_k)\right]}}\right) \geq p_k, \tag{5.39}$$

where $\Phi(\cdot)$ represents the Cumulative Distribution Function (CDF) of the standard Gaussian. Equation (5.39) implies that if the uncertainty about the system dynamics decreases to near zero, the probability $\Pr(\mathfrak{C}(\boldsymbol{x}_k, \boldsymbol{u}_k) \geq 0)$ tends to one, which is our desire. We use

[1] $\Pr(Z \geq a) = 1 - \Phi(a)$

the property of the inverse CDF, $\Phi^{-1}(1-p_k) = -\sqrt{2}\text{erf}^{-1}(2p_k - 1)$ and reformulate the chance constraint (5.29) as follows

$$\begin{aligned} \boldsymbol{u}_s = \underset{\boldsymbol{u}\in\mathbb{R}^m}{\arg\min}\ & \|\boldsymbol{u} - \boldsymbol{u}_{nom}\|_2^2 \\ \text{s.t.}\quad & \mathbb{E}[\mathfrak{C}(\boldsymbol{x}_k, \boldsymbol{u}_k)] \geq \sqrt{2}\text{erf}^{-1}(2p_k - 1)\sqrt{\text{Var}\left[\mathfrak{C}(\boldsymbol{x}_k, \boldsymbol{u}_k)\right]}. \end{aligned} \tag{5.40}$$

We can show that the optimization problem (5.40) is convex and can be solved in real time.

Proposition 5.4. *The optimization problem (5.29) with chance constraint which has expectation (5.37) and variance (5.38) is convex and can be converted into the standard second-order cone program of the form*

$$\begin{aligned} \boldsymbol{u}_s = \underset{z\in\mathbb{R},\boldsymbol{u}\in\mathbb{R}^m}{\arg\min}\ & z \\ s.t.\ \ & \|\boldsymbol{u} - \boldsymbol{u}_{nom}\|_2 \leq z, \\ & \|\boldsymbol{A}_k(\boldsymbol{x}_k)\,\bar{\boldsymbol{u}}\|_2 \leq c_k^{-1}\boldsymbol{b}_k(\boldsymbol{x}_k)^\top\,\bar{\boldsymbol{u}}, \end{aligned} \tag{5.41}$$

where

$$\begin{aligned} \boldsymbol{A}_k(\boldsymbol{x}_k) &\triangleq \sqrt{\nabla_{\boldsymbol{x}} h(\boldsymbol{x}_k)^\top A \nabla_{\boldsymbol{x}} h(\boldsymbol{x}_k)}\, B_k^{\frac{1}{2}}(\boldsymbol{x}_k, \boldsymbol{x}_k), \\ \boldsymbol{b}_k(\boldsymbol{x}_k) &\triangleq M_k(\boldsymbol{x}_k)^\top \nabla_{\boldsymbol{x}} h(\boldsymbol{x}_k) + \begin{bmatrix} \alpha(h(\boldsymbol{x}_k)) \\ \boldsymbol{0}_m \end{bmatrix}, \end{aligned}$$

and $c_k \triangleq \sqrt{2}erf^{-1}(2p_k - 1)$ *and* $B_k^{\frac{1}{2}}(\boldsymbol{x}_k, \boldsymbol{x}_k)$ *represents the Cholesky decomposition of* $B_k(\mathbf{x}_*, \mathbf{x}'_*)$.

Proof. The first constraint converts the objective function of the original problem (5.29) into a second-order cone constraint. The second constraint is obtained by substituting (5.37), (5.38) into the safety constraint in the original problem. We can also express the optimization problem (5.41) in standard SOCP form by following the same approach as Theorem 3.2. □

Proposition 5.4 implies that the original optimization problem in (5.29) remains convex when MVGP is used to approximate the system dynamics. However, this formulation is based on CBF of relative degree one. The extension to higher-order relative degrees is done by Khojasteh et al. [83] based on the ECBF method. Recall the ECBF safety constraint from Definition 3.10:

$$\underbrace{h^{(r)}(\boldsymbol{x},\boldsymbol{u})+\boldsymbol{k}^\top\boldsymbol{\eta}(\boldsymbol{x})}_{:=\mathfrak{C}^r(\boldsymbol{x}_k,\boldsymbol{u}_k)}\geq 0, \tag{5.42}$$

where $\eta(\boldsymbol{x})=\begin{bmatrix}h(\boldsymbol{x}) & \dot{h}(\boldsymbol{x}) & \dots, h^{(r-1)}(\boldsymbol{x})\end{bmatrix}^\top$ and $\boldsymbol{k}\in\mathbb{R}^r$ is the row vector of gains. Note that based on the definition of the relative degree, $h^{(i)}(\boldsymbol{x})=L_f^i h(\boldsymbol{x})$ for $i=1,\dots,r-1$. The following lemma shows a property of ECBF design.

Lemma 5.7. *Consider the dynamics in (5.28) which has relative degree $r\geq 1$ with respect to the barrier function $h(\boldsymbol{x})$. Also, consider the posterior distribution (5.36) of the system dynamics approximated by MVGP. Then, $\mathfrak{C}^r(\boldsymbol{x}_k,\boldsymbol{u}_k)$ is a Gaussian random variable with mean and variance which are linear and quadratic in $\boldsymbol{u}$, respectively.*

Proof. First, we rewrite $\mathfrak{C}^r(\boldsymbol{x}_k,\boldsymbol{u}_k)$ in the form of (5.28):

$$\begin{aligned}
&\mathfrak{C}^r(\boldsymbol{x}_k,\boldsymbol{u}_k)=L_f^r h(\boldsymbol{x})+L_gL_f^{r-1}h(\boldsymbol{x})\boldsymbol{u}+\boldsymbol{k}^\top\boldsymbol{\eta}(\boldsymbol{x})\\
&=L_fL_f^{r-1}h(\boldsymbol{x})+L_gL_f^{r-1}h(\boldsymbol{x})\boldsymbol{u}+\boldsymbol{k}^\top\boldsymbol{\eta}(\boldsymbol{x})\\
&=\left(\nabla_{\boldsymbol{x}}h^{(r-1)}(\boldsymbol{x})\right)^\top f(\boldsymbol{x})+\left(\nabla_{\boldsymbol{x}}h^{(r-1)}(\boldsymbol{x})\right)^\top g(\boldsymbol{x})\boldsymbol{u}+\boldsymbol{k}^\top\boldsymbol{\eta}(\boldsymbol{x})\\
&=\left(\nabla_{\boldsymbol{x}}h^{(r-1)}(\boldsymbol{x})\right)^\top \mathcal{F}(\boldsymbol{x})\bar{\boldsymbol{u}}+\boldsymbol{k}^\top\boldsymbol{\eta}(\boldsymbol{x}).
\end{aligned}$$

We rewrite the above equation in the following form

$$\mathfrak{C}^r(\boldsymbol{x}_k,\boldsymbol{u}_k)=\left((\nabla_{\boldsymbol{x}}h^{(r-1)}(\boldsymbol{x}))^\top\mathcal{F}(\boldsymbol{x})+\begin{bmatrix}\boldsymbol{k}^\top\boldsymbol{\eta}(\boldsymbol{x})\\ \boldsymbol{0}_m\end{bmatrix}^\top\right)\bar{\boldsymbol{u}}.$$

Next, we use the linearity of the expectation operator to obtain

$$\mathbb{E}(\mathfrak{C}^r(\boldsymbol{x}_k,\boldsymbol{u}_k))=\mathbb{E}(\mathcal{F}(\boldsymbol{x})^\top\nabla_{\boldsymbol{x}}h^{(r-1)}(\boldsymbol{x})+\begin{bmatrix}\boldsymbol{k}^\top\boldsymbol{\eta}(\boldsymbol{x})\\ \boldsymbol{0}_m\end{bmatrix})^\top\bar{\boldsymbol{u}},$$

and using the multiplication properties of the variance, we have

$$\mathrm{Var}(\mathfrak{C}^r(\boldsymbol{x}_k,\boldsymbol{u}_k))=\bar{\boldsymbol{u}}^\top\,\mathrm{Var}((\nabla_{\boldsymbol{x}}h^{(r-1)}(\boldsymbol{x}))^\top\mathcal{F}(\boldsymbol{x})+\begin{bmatrix}\boldsymbol{k}^\top\boldsymbol{\eta}(\boldsymbol{x})\\ \boldsymbol{0}_m\end{bmatrix}^\top)\bar{\boldsymbol{u}},$$

which concludes the proof. □

We know that the covariance matrix is symmetric and semi-positive definite, thus we can express the variance of the safety constraint by $\text{Var}(\mathfrak{C}^r(\boldsymbol{x}_k, \boldsymbol{u}_k)) = \bar{\boldsymbol{u}}^\top A_k^{(r)} {A_k^{(r)}}^\top$. We also denote the expectation by $\mathbb{E}(\mathfrak{C}^r(\boldsymbol{x}_k, \boldsymbol{u}_k)) = {\boldsymbol{b}_k^{(r)}}^\top$. Next, we have the following result for the safety filter with high-order CBF.

Proposition 5.5. *The optimization problem of the form (5.29) with high-order chance constraint (5.42) is convex and can be converted into the standard second-order cone program of the form*

$$\begin{aligned} \boldsymbol{u}_s = \; & \underset{z\in\mathbb{R}, \boldsymbol{u}\in\mathbb{R}^m}{\arg\min} \; z \\ s.t. \; & \|\boldsymbol{u} - \boldsymbol{u}_{nom}\|_2 \leq z, \\ & \left\| A_k^{(r)}(\boldsymbol{x}_k)\, \bar{\boldsymbol{u}}_k \right\|_2 \leq c_k^{(r)} \boldsymbol{b}_k^{(r)}(\boldsymbol{x}_k)^\top \underline{\boldsymbol{u}}, \end{aligned} \tag{5.43}$$

where $c_k^{(r)} = \sqrt{\frac{1}{p_k} - 1}$ *and* p_k *is a predefined probability margin for safety.*

Proof. Since $\mathfrak{C}^r(\boldsymbol{x}_k, \boldsymbol{u}_k)$ includes the derivatives of vector fields f and g, we cannot use CDF to convert the chance constraint to a constraint based on expectation and variance (similar to relative degree one case). Instead, in [83] the Cantelli's inequality is used to obtain a lower bound on the chance constraint based on the variance which states that for any $\alpha > 0$:

$$\Pr\left(\mathfrak{C}^r(\boldsymbol{x}_k, \boldsymbol{u}_k) \geq \mathbb{E}\left[\mathfrak{C}^r(\boldsymbol{x}_k, \boldsymbol{u}_k)\right] - \alpha\right) \geq 1 - \frac{\text{Var}\left[\mathfrak{C}^r(\boldsymbol{x}_k, \boldsymbol{u}_k)\right]}{\text{Var}\left[\mathfrak{C}^r(\boldsymbol{x}_k, \boldsymbol{u}_k)\right] + \alpha^2}.$$

The probability must be greater than the probability margin for safety, we have

$$1 - \frac{\text{Var}\left[\mathfrak{C}^r(\boldsymbol{x}_k, \boldsymbol{u}_k)\right]}{\text{Var}\left[\mathfrak{C}^r(\boldsymbol{x}_k, \boldsymbol{u}_k)\right] + \alpha^2} \geq p_k.$$

Considering the quadratic form for the variance and substituting $\alpha = \mathbb{E}\left[\mathfrak{C}^r(\boldsymbol{x}_k, \boldsymbol{u}_k)\right]$, a relationship between the variance and expectation is obtained which leads to the constraint in (5.43). The remainder of the proof proceeds similarly as before. □

5.5.3 Comparison of the SOCP-HOCBF and MVGP

In Section 5.4, we introduced a GP-based safety controller designed to approximate the impact of uncertainty on safety constraints for HOCBFs. Additionally, we reviewed the

MVGP methodology in Section 5.5, which models the system dynamics vector field and incorporates this approximation within the safety constraints. Although both approaches result in a SOCP formulation, key differences exist between them. This section provides a comparative analysis of the effectiveness and practical implications of each approach.

Structural differences: A primary difference between the SOCP-HOCBF and MVGP approaches lies in the structure of the high-order CBFs that is being employed by each method. In the SOCP-HOCBF approach, high-order CBFs [169] are directly incorporated, with system uncertainties explicitly propagated within the HOCBF framework. By contrast, the MVGP approach adopts the exponential CBF structure [114] and leverages approximations of the system dynamics to mitigate the adverse effects of system uncertainty on safety. Since HOCBFs provide a more general formulation than ECBF design, the SOCP-HOCBF method can be viewed as a more generalized approach than MVGP in handling system uncertainty with high relative degrees in safety-critical control applications.

Use of approximation: Both the MVGP and GP-HOCBF approaches rely on approximations at different stages of the process. In the MVGP approach, dataset is collected directly from system trajectories to fit a MVGP model. Once the hyperparameters controlling the covariance function are optimized, the model can be used to infer the posterior distribution of the system dynamics at any given state. However, for high-order CBFs, the distribution is no longer a GP due to the presence of high-order time derivatives of the barrier function. To address this, a lower bound for the probabilistic safety constraint was introduced, and the required mean and variance of the high-order safety constraint can be estimated by Monte Carlo sampling for arbitrary relative degrees. On the other hand, the GP-HOCBF approach requires finite difference methods to collect the dataset, meaning the dataset is not directly derived from system trajectories and instead requires some level of approximation. But, once the hyperparameters for the covariance matrix are optimized and the GP model is obtained the posterior distribution can be directly applied in the SOCP optimization, allowing for efficient handling of uncertainty within the HOCBF framework.

Computational complexity: The learning complexity of the MVGP approach is highly dependent on the dimension of the system. To reduce the number of learning parameters, one approach is to fix the kernel to a single choice. In the special case of $B_0(\boldsymbol{x}, \boldsymbol{x}') = Bk_0(\boldsymbol{x}, \boldsymbol{x}')$, where $k_0 : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ is a valid kernel, in addition to the kernel parameters, MVGP requires learning the elements of $B \in \mathbb{R}^{(m+1)\times(m+1)}$, $A \in \mathbb{R}^{n\times n}$, and $\sigma \in \mathbb{R}$. This is a key property of MVGP design that the posterior preserves the kernel decomposition, distinguishing it from similar methods like CoGP [103], which involve learning more than $(1+m)^2n^2$ parameters. Additionally, for a dataset of N samples, MVGP

requires an $N \times N$ matrix inversion to compute the posterior variance, resulting in a training complexity of $O(N^3)$. In GP-HOCBF design, the learning complexity depends on both the number of system inputs and the relative degree. We need to learn the parameters of at most $m + r$ kernels. While GP-HOCBF may involve more parameters, it enables a more adaptable kernel selection compared to MVGP's single kernel design. The computational complexity of this method is also $O(N^3)$ for a dataset of N samples.

5.6 Simulation Results

In this section, two simulation studies highlight the effectiveness of the proposed method. Simulations are run in the Python environment, and the optimization problems are solved using the CVXPY [51] package.

5.6.1 Adaptive cruise control with collision avoidance

Consider an adaptive cruise control modeled by

$$\dot{\boldsymbol{x}} = \begin{bmatrix} -\frac{f_0 + f_1 v + f_2 v^2}{m} \\ v_0 - v \end{bmatrix} + \begin{bmatrix} \frac{1}{m} \\ 0 \end{bmatrix} u, \tag{5.44}$$

where the system state $\boldsymbol{x} = \begin{bmatrix} v & z \end{bmatrix}^\top \in \mathbb{R}^2$ containing the ego car's forward velocity v, and the distance between the ego car and the preceding car, denoted by z. The wheel force $u \in \mathbb{R}$ is the control input of the ego vehicle. The model assumes a constant velocity v_0 for the preceding car; however, there exists uncertainty in the estimation of v_0. The mass of the ego car is denoted by m and it experiences rolling resistance modeled as $f_0 + f_1 v + f_2 v^2$.

The control objective is to reach a target speed v_d while maintaining a safe distance D to the vehicle in front. We are provided with a nominal model characterized with $m = 825\,kg$, and $f_0 = 0.1, f_1 = 5, f_2 = 0.25$, and $v_0 = 16\,m/s$. The true system parameters, which are unknown, are $m = 3300\,kg$, $f_0 = 0.2, f_1 = 10, f_2 = 0.5$, and $v_0 = 14\,m/s$.

We consider CLF $V(\boldsymbol{x}) = (v - v_d)^2$, and use GP-based CLF approach [33] to stabilize the system near the desired velocity $v_d = 24\,m/s$. GP-based CLF can be converted to a second-order cone constraint and incorporated to the (5.20) as a soft constraint. We have $h(\boldsymbol{x}) = z - D$, where $D = 30\,m$, as the CBF of the system, which has relative degree two, with respect to the system (5.44). The vector of coefficients is $\boldsymbol{\gamma} = \begin{bmatrix} 4 & 3.75 \end{bmatrix}^\top$ in the HOCBF design. Figure 5.1 illustrates the performance of the proposed method in

comparison to the nominal safety-critical controller. It also includes the result of the oracle QP-HOCBF that is designed by the true dynamics as a reference.

The simulation is started from the initial point $\boldsymbol{x}_0 = \begin{bmatrix}20, 100\end{bmatrix}^\top$. First, the controller increases the velocity to converge to the desired speed, which decreases z. The distance between the ego car and the preceding car reaches the limit after 7 seconds of the simulation. Due to the adverse effect of uncertainty, the nominal safety controller violates the safe distance. However, the GP-based controller could avoid unsafe behavior. It can be verified that the proposed method could recover the true system's performance.

We used the episodic data collection method. For the first episode, we run the system using nominal QP-HOCBF design and collect the system trajectory, until it becomes unsafe. Then, we use the proposed GP model with a high probability bound $(1 - \delta = 0.95)$ and apply the SOCP controller, which contains an uncertainty-aware safety constraint. We add the collected data to the dataset at each episode, until the system passes the simulation time without any unsafe interruption. A total number of 119 samples are collected for this simulation. We used infinitely differentiable squared exponential kernel [167], for $k_i, i = 1, 2, 3$ individual kernels in the structure of k_c.

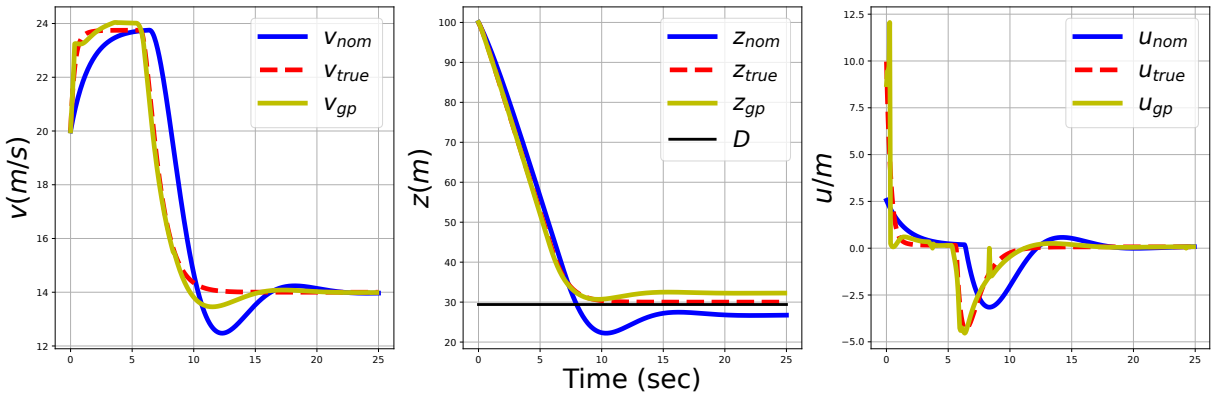


Figure 5.1: The state variables and control input in the presence of uncertainty when applying the GP-based SOCP-HOCBF (yellow) compared to the nominal QP-HOCBF (blue) and the QP-HOCBF design based on the true system (red dashed).

5.6.2 Active suspension system

Consider two degrees of freedom, active suspension system of a quarter car (see Figure 5.2 for an illustration), modeled in the form of $\dot{\boldsymbol{x}} = f(\boldsymbol{x}) + g(\boldsymbol{x})u + g_d(\boldsymbol{x})d$, as follows:

$$\dot{\boldsymbol{x}} = \begin{bmatrix} x_3 \\ x_4 \\ \frac{k_1(x_2-x_1)+b(x_4-x_3)}{m_1} \\ \frac{k_1(x_1-x_2)-k_2x_2+b(x_3-x_4)}{m_2} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \frac{1}{m_1} \\ -\frac{1}{m_2} \end{bmatrix} u + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{k_2}{m_2} \end{bmatrix} d.$$

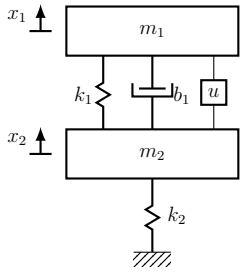


Figure 5.2: Active suspension system.

The state vector denoted by $\boldsymbol{x} = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 \end{bmatrix}^\top \in \mathbb{R}^4$, where x_1 and x_2 are the vertical displacements of the body and wheel from their equilibrium position, respectively. x_3 and x_4 represent the vertical velocity of the body and wheel, respectively, indicating how fast the body and wheel are moving vertically. The control input $u \in \mathbb{R}$ is the force applied to the system, enabling the vehicle to respond to external forces such as road irregularities, which are modeled by $d \in \mathbb{R}$. By controlling the forces applied to the suspension components, the system can achieve improved ride comfort and handling characteristics. Therefore, the control objective is regulating x_1 and x_2 to zero displacement. However, large displacements for x_1 must be prohibited to avoid excessive motion that could impact the safety of the passengers.

Safety in this problem is defined as avoiding vertical displacements larger than a threshold D. We consider $D = 6\,cm$, and the CBF is $h(\boldsymbol{x}) = D - x_1$, which has relative degree two, with respect to the active suspension system. The nominal model parameters are $m_1 = 300\,kg, m_1 = 60\,kg, k_1 = 16\,kN/m, k_2 = 190\,kN/m, b = 1\,kN.s/m$. The true model is obtained by $m_1 = 675\,kg, m_1 = 135\,kg, k_1 = 36\,kN/m, k_2 = 427.5\,kN/m, b = 2.25\,kN.s/m$. We use Linear Quadratic Regulator (LQR) as the stabilization controller with weighting

matrices $Q = 10I$ and $R = 1$, and the vector of coefficients in the HOCBF design is $\boldsymbol{\gamma} = \begin{bmatrix} 41 & 395 \end{bmatrix}^\top$. The system is started from its equilibrium point $\boldsymbol{x}_0 = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}^\top$.

We used the same data collecting approach, which collected 174 data points in total. The same kernel structure as the previous example is selected. Figure 5.3 shows the states x_1 and x_3 and the control input, together with the road disturbance profile. It highlights that the nominal QP-HOCBF controller violates the threshold, while the proposed GP-based SOCP controller successfully recovers the performance of the true system.

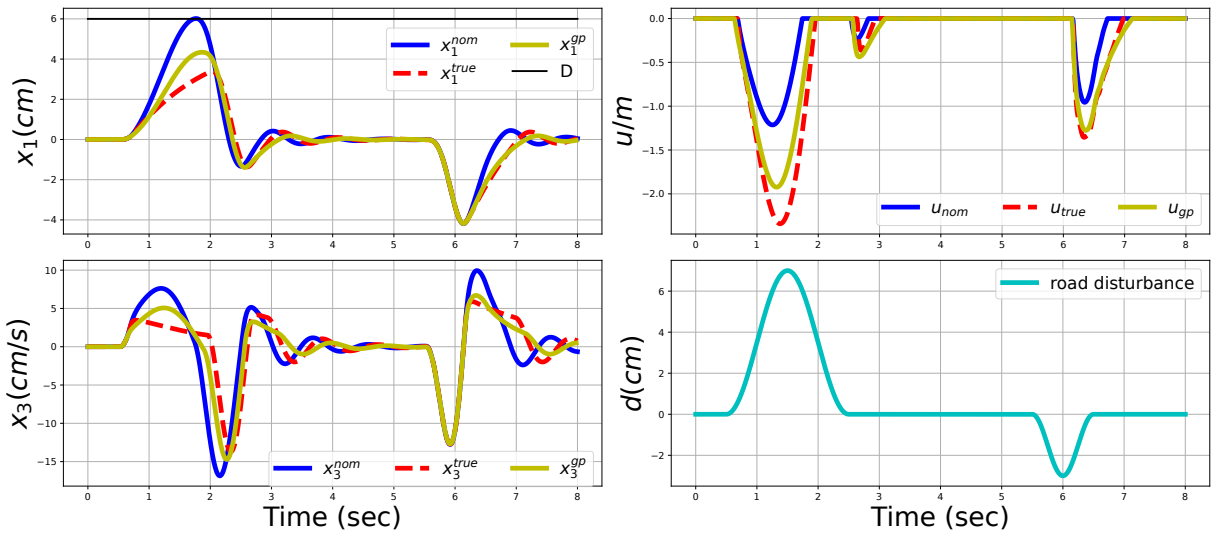


Figure 5.3: Comparison of the GP-based SOCP-HOCBF (yellow), nominal QP-HOCBF (blue), and true QP-HOCBF (dashed red). States x_1 and x_3 of the active suspension system (left column). Control input and the road disturbance profile (right column).

5.7 Chapter Summary

In this chapter, A GP framework is developed to approximate the effect of model uncertainty on the high-order safety certificate. By choosing linear class $\mathcal{K}$ functions in HOCBF design, the residual term in the high-order safety certificate is fully characterized. Then, the resulting constrained optimization problem with the uncertainty-aware chance constraint is converted into an SOCP which can be solved in real time. Finally, the feasibility of the SOCP is addressed.

Chapter 6

Learning Piecewise Residuals of CBFs for Safety of Switching Systems

6.1 Introduction

Many real-world control problems are modeled as hybrid dynamical systems, which involve a coupling between continuous dynamics and discrete events. Research in the stability analysis of such systems has grown significantly in recent decades [28, 47, 99]. More recently, GPs are used to approximate hybrid residual dynamics model which can be integrated into a stabilizing model predictive control design ensuring probabilistic constraint satisfaction [52]. However, in modern control applications, the focus has expanded from mere stability assurance to encompass safety verification. CBFs have recently provided a powerful theoretical tool for synthesizing controllers that ensure the safety of dynamical systems. They were initially applied to autonomous driving and bipedal walking [10]. Stability guarantees are then added via a unified Control Lyapunov Function-Control Barrier Function (CLF-CBF) framework by solving a state-dependent QP problem at each time step to compute the control input [129]. In the context of hybrid systems, barrier functions were initially used as safety certificates in [118]. After the introduction of CBFs, CLF-CBF design has been extensively used in safety-critical methods for hybrid systems [10, 106, 109]. However, there are very few studies on learning-based safety verification of switching systems via CBFs in the presence of uncertainty.

Learning-based CBFs for uncertain systems can be divided into two approaches. The first approach learns the unknown dynamics and uses the approximated model to derive safety certificates. In [164], a GP is employed to learn the unmodeled dynamics for the safe

navigation of quadrotors with CBFs. A GP-based systematic controller synthesis method is introduced in [75], offering an inherently safe control strategy. In [54], the authors employ a Bayesian approach to learn model uncertainty while ensuring safety during the learning process. Learning the complete system dynamics with GPs is computationally demanding. Therefore, a common approach is to learn each component of the vector fields individually, disregarding their correlations. Furthermore, existing studies typically neglect uncertainty in the actuation term to simplify the learning-based model.

These limitations lead us to the second approach, which approximates the projection of the residual onto the CLF and CBF. In [46], an episodic learning method using neural networks is employed to approximate the impact of unmodeled dynamics of hybrid bi-pedal robots on system safety, eliminating the need for exhaustive data collection. However, methods employing neural networks must achieve precise approximations of the dynamics to formally ensure safety. In general, this requirement is challenging to meet, but GPs can provide uncertainty quantification with analytical confidence bounds. In [34], a GP-based min-norm controller stabilizing an unknown control affine system while guaranteeing safety using CLF-CBF method is introduced. The effect of the uncertainty in the stability and safety constraints is approximated by a GP model. This technique results in the approximation of a real-valued function instead of a vector field, leading to a significant reduction in the model complexity for high-dimensional systems. In [3], the effect of uncertainty on high-order CBFs is quantified using GPs.

In this chapter, we develop a batch MOGP model to learn the piecewise residuals in each switching surface. We use the multi-output design to learn the piecewise residuals corresponding to CLF and CBF simultaneously. The main contributions of this chapter relative to prior works are presented as follows:

- We show that the switching system imposes piecewise residuals on CLF and CBF constraints. We develop a batch MOGP model to efficiently approximate residuals in each switching surface.
- We demonstrate that by selecting a particular form of the covariance function in the batch MOGP structure, the resulting min-norm controller with an uncertainty-aware chance constraint can be converted into an SOCP.
- We analyze the feasibility of the resulting constrained optimization problem and determine the necessary and sufficient conditions for feasibility.

6.2 Problem Setup and Statement

Consider the control-affine switching system of the form

$$\dot{\boldsymbol{x}} = \sum_{r=1}^{R} \delta_r \left(f_r(\boldsymbol{x}) + g_r(\boldsymbol{x})\boldsymbol{u} \right), \tag{6.1}$$
$$\delta_r = \begin{cases} 1 & \text{if} \quad \boldsymbol{x} \in \mathcal{R}_r, \\ 0 & \text{otherwise}, \end{cases}$$

where $\boldsymbol{x} \in \mathcal{X} \subset \mathbb{R}^n$ is the state, and $\boldsymbol{u} \in \mathbb{R}^m$ is the input, and $\mathcal{X}$ is a compact and convex set. The function $\delta_r \colon \mathcal{X} \to \{0, 1\}, r \in \mathcal{I}$ is a state-dependent switching signal with the finite index set $\mathcal{I} = \{1, \ldots, R\}$ which indicates the index of the active switching surface $\mathcal{R}_r \subset \mathcal{X}$. The vector fields $f_r \colon \mathbb{R}^n \to \mathbb{R}^n$ and $g_r \colon \mathbb{R}^n \to \mathbb{R}^{m \times n}, r \in \mathcal{I}$, are the unknown dynamics, which are assumed to be locally Lipschitz in their arguments. Let $\pi \colon \mathbb{R}^n \to \mathbb{R}^m$ be a locally Lipschitz continuous state feedback control law. We assume that the impulse effects are absent, i.e., the reset map is the identity. Furthermore, we assume that the closed-loop system

$$\dot{\boldsymbol{x}} = F_{cl}(\boldsymbol{x}) \triangleq \sum_{r=1}^{R} \delta_r \left(f_r(\boldsymbol{x}) + g_r(\boldsymbol{x})\pi(\boldsymbol{x}) \right), \tag{6.2}$$

satisfies Caratheodory condition [71]. Then, for any initial state $\boldsymbol{x}(t_0) = \boldsymbol{x}_0$, system (6.2) admits a solution $\boldsymbol{x}(t)$ in the sense of Caratheodory defined on a maximal interval of existence $I(\boldsymbol{x}_0) = [t_0, I_{max})$. In this research, we consider complete solutions, thus $I_{max} = \infty$.

Assumption 6.1. *We assume that the state space is partitioned into R non-overlapping regions labeled as $\mathcal{R}_r \subset \mathcal{X}$, such that they cover the whole state space, i.e., $\cup_{r=1}^{R} \mathcal{R}_r = \mathcal{X}$, and $\mathcal{R}_i \cap \mathcal{R}_j = \emptyset$ for all $i, j \in \{1, \ldots, R\}, i \neq j$.*

Assumption 6.1 imposes that system (6.1) needs to be well-posed, i.e., for all $\boldsymbol{x} \in \mathcal{X}$ there exists only one active index $r \in \mathcal{I}$ satisfying the membership condition.

6.2.1 Control Lyapunov functions for switching systems

Definition 6.1 (Control Lyapunov functions)**.** *A continuously differentiable function of the form $V \colon \mathcal{X} \to \mathbb{R}$ is a CLF for system (6.1) if there exist positive constants $\lambda, c_1, c_2 > 0$*

such that for all $\boldsymbol{x} \in \mathcal{X}$ *and* $r \in \mathcal{I}$,

$$
\begin{aligned}
& c_1\|\boldsymbol{x}\|^2 \leq V(\boldsymbol{x}) \leq c_2\|x\|^2, \\
& \sum_{r=1}^{R} \delta_r \left(L_{f_r} V(\boldsymbol{x}) + L_{g_r} V(\boldsymbol{x})\boldsymbol{u}\right) \leq -\lambda V(\boldsymbol{x}),
\end{aligned} \tag{6.3}
$$

where $L_{f_r}V(\boldsymbol{x})$ *and* $L_{g_r}V(\boldsymbol{x})$ *are the Lie derivatives of* $V(\boldsymbol{x})$ *with respect to the corresponding vector fields.*

Therefore, if the subsystems in (6.1) share a CLF, the rate of decrease of V along the vector fields, given by (6.3), is not affected by the switching, thus exponential stability is uniform with respect to δ_r. Also, if there exists a compact subset $D \subseteq \mathcal{X}$ including the origin such that (6.3) holds for some $\lambda > 0$, and there exists a sublevel set $\Omega = \{\boldsymbol{x} \in D \mid V(\boldsymbol{x}) \leq c\}$, where $c > 0$, then the origin is locally exponentially stabilizable from Ω.

6.2.2 Control barrier functions

CBF method defines safety based on the notion of set invariance, where a subset of the state space is specified as the safe set. This set is characterized by the zero superlevel set of a continuously differentiable function $h\colon \mathbb{R}^n \to \mathbb{R}$ as

$$
\begin{aligned}
\mathcal{C} &= \{\boldsymbol{x} \in \mathbb{R}^n \mid h(\boldsymbol{x}) \geq 0\}, \\
\partial\mathcal{C} &= \{\boldsymbol{x} \in \mathbb{R}^n \mid h(\boldsymbol{x}) = 0\}.
\end{aligned} \tag{6.4}
$$

CBFs provide a constructive tool for achieving forward invariance of set $\mathcal{C}$. We define a CBF as follows:

Definition 6.2 (Control barrier function [10])**.** *Given the set* $\mathcal{C}$ *as defined in (6.4), the continuously differentiable function* $h(\boldsymbol{x})$ *is called a CBF on a domain* $\mathcal{D}$ *with* $\mathcal{C} \subset \mathcal{D} \subset \mathbb{R}^n$*, if there exists a class* $\mathcal{K}$ *function* α *such that the following holds for all* $\boldsymbol{x} \in \mathcal{D}$ *and* $r \in \mathcal{I}$

$$
\sum_{r=1}^{R} \delta_r \left(L_{f_r} h(\boldsymbol{x}) + L_{g_r} h(\boldsymbol{x})\boldsymbol{u}\right) \geq -\alpha(h(\boldsymbol{x})). \tag{6.5}
$$

We can derive the following Corollary based on Nagumo's theorem [25] for the forward invariance of the safe set $\mathcal{C}$:

Corollary 6.1. *Given CBF $h\colon \mathbb{R}^n \to \mathbb{R}$ with the associated set $\mathcal{C}$ in (6.3), if $\nabla h(\boldsymbol{x}) \neq 0$ for all $\boldsymbol{x} \in \partial\mathcal{C}$, any Lipschitz continuous controller $\boldsymbol{u}(\boldsymbol{x})$ satisfying (6.5), guarantees that the set $\mathcal{C}$ is forward invariant for the system (6.1) and thus safe.*

Inequalities (6.3) and (6.5) impose an affine condition on the control values which can be used to ensure stability and safety. Consider the CLF constraint (6.3), primarily designed to fulfill control objectives. Our goal is to apply the control law that satisfies (6.3) to the system only if it complies with (6.5). In practice, this is accomplished by solving the following QP optimization:

$$\boldsymbol{u}_s = \underset{(\boldsymbol{u},d)\in\mathbb{R}^{m+1}}{\arg\min} \ \|\boldsymbol{u}\|_2^2 + \rho d^2 \tag{6.6a}$$

$$\text{s.t.} \quad \sum_{r=1}^{R} \delta_r \left(L_{f_r} V(\boldsymbol{x}) + L_{g_r} V(\boldsymbol{x})\boldsymbol{u}\right) + \lambda V(\boldsymbol{x}) \leq d, \tag{6.6b}$$

$$\sum_{r=1}^{R} \delta_r \left(L_{f_r} h(\boldsymbol{x}) + L_{g_r} h(\boldsymbol{x})\boldsymbol{u}\right) + \alpha(h(\boldsymbol{x})) \geq 0, \tag{6.6c}$$

where ρ is a positive coefficient and $d \in \mathbb{R}$ is a slack variable. The resulting minimally invasive point-wise controller prioritizes safety over control objectives by satisfying (6.5) as a hard constraint.

6.2.3 Impact of uncertainty on the switching systems

In this section, we focus on reformulating CLF and CBF constraints such that they take the model uncertainty into account. We refer to the system (6.1) as the true system, which is unknown and modeled by a single nominal model in all regions as

$$\dot{\boldsymbol{x}} = \hat{f}(\boldsymbol{x}) + \hat{g}(\boldsymbol{x})\boldsymbol{u}, \tag{6.7}$$

where $\hat{f}\colon \mathbb{R}^n \to \mathbb{R}^n$ and $\hat{g}\colon \mathbb{R}^n \to \mathbb{R}^{n\times m}$ are locally Lipschitz in $\boldsymbol{x}$. We design a locally exponentially stabilizing CLF and CBF based on the nominal system and assume they are also valid for the true system.

Due to the inherent model mismatch between the true system and the nominal model, the time derivatives of $V(\boldsymbol{x})$ and $h(\boldsymbol{x})$ based on the nominal model differ from their true

values. The resulting error can be represented as

$$d^V(\boldsymbol{x},\boldsymbol{u}) = \dot{V}(\boldsymbol{x},\boldsymbol{u}) - \hat{\dot{V}}(\boldsymbol{x},\boldsymbol{u}) = \sum_{r=1}^{R} \delta_r d_r^V(\boldsymbol{x},\boldsymbol{u}),$$
$$d^h(\boldsymbol{x},\boldsymbol{u}) = \dot{h}(\boldsymbol{x},\boldsymbol{u}) - \hat{\dot{h}}(\boldsymbol{x},\boldsymbol{u}) = \sum_{r=1}^{R} \delta_r d_r^h(\boldsymbol{x},\boldsymbol{u}),$$

where for $r \in \mathcal{I}$, we have

$$\begin{aligned} d_r^V &= L_{f_r}V(\boldsymbol{x}) - L_{\hat{f}}V(\boldsymbol{x}) + \left(L_{g_r}V(\boldsymbol{x}) - L_{\hat{g}}V(\boldsymbol{x})\right)\boldsymbol{u}, \\ d_r^h &= L_{f_r}h(\boldsymbol{x}) - L_{\hat{f}}h(\boldsymbol{x}) + \left(L_{g_r}h(\boldsymbol{x}) - L_{\hat{g}}h(\boldsymbol{x})\right)\boldsymbol{u}, \end{aligned} \tag{6.8}$$

where the dependence on $\boldsymbol{x}$ and $\boldsymbol{u}$ is removed for simplicity. Consequently, these discrepancies propagate into the CLF (6.6b) and CBF (6.6c) constraints, potentially jeopardizing the stability and safety of the system. Using (6.8), the constraints (6.6b) and (6.6c) become

$$L_{\hat{f}}V(\boldsymbol{x}) + L_{\hat{g}}V(\boldsymbol{x})\boldsymbol{u} + \sum_{r=1}^{R} \delta_r d_r^V + \lambda V(\boldsymbol{x}) \leq d, \tag{6.9}$$

$$L_{\hat{f}}h(\boldsymbol{x}) + L_{\hat{g}}h(\boldsymbol{x})\boldsymbol{u} + \sum_{r=1}^{R} \delta_r d_r^h + \alpha(h(\boldsymbol{x})) \geq 0. \tag{6.10}$$

In this chapter, our primary focus is characterizing the adverse effects of uncertainty in the proposed safety-critical control method. We then adopt a supervised learning approach to make our design robust to piecewise residuals of the form (6.8). Thus, a dataset that contains information about piecewise residuals in all regions is needed.

Assumption 6.2. *We have access to the set of samples* $\{(\boldsymbol{x}(t),\boldsymbol{u}(t))\}, t \in [t_0, t_f]$ *collected with sampling time* Δt *from trajectories of the system. We assume this set is rich and contains states from all regions* $\mathcal{R}_r, r \in \mathcal{I}$.

Given these data samples, we can approximately measure d^V and d^h by collecting trajectories from the true system and using the finite difference method to obtain

$$\begin{aligned} \omega_j^V &= \left(V\left(\boldsymbol{x}(t_j+\Delta t)\right) - V\left(\boldsymbol{x}(t_j)\right)\right)/\Delta t - \hat{\dot{V}}\left(\boldsymbol{x}_j,\boldsymbol{u}_j\right), \\ \omega_j^h &= \left(h\left(\boldsymbol{x}(t_j+\Delta t)\right) - h\left(\boldsymbol{x}(t_j)\right)\right)/\Delta t - \hat{\dot{h}}\left(\boldsymbol{x}_j,\boldsymbol{u}_j\right), \end{aligned} \tag{6.11}$$

where $t_j = t_0 + (j-1)\Delta t$, $\boldsymbol{x}_j = \left(\boldsymbol{x}(t_j + \Delta t) + \boldsymbol{x}(t_j)\right)/2$ is the mean of the state, and $\boldsymbol{u}_j = \boldsymbol{u}(t_j)$ is the control input for $t_j \in [t_j, t_j + \Delta t), j \in \{1, \ldots, N\}$.

A dataset can be generated by collecting trajectories of the true system. We denote it by $\mathcal{D} = \{((\boldsymbol{x}_j, \boldsymbol{u}_j), \omega_j^V, \omega_j^h)\}_{j=1}^N$, where $(\boldsymbol{x}_j, \boldsymbol{u}_j) \in \mathcal{X} \times \mathbb{R}^m$ is the input data and outputs ω_j^V and ω_j^h are obtained from (6.11). We will use this dataset to learn piecewise residuals.

6.3 Proposed Batch MOGP-based Design for Piecewise Residuals

6.3.1 Single-output GP for real-valued functions

Typical Single-Output Gaussian Process (SOGP) provides a framework to approximate nonlinear functions. We represent an SOGP approximation of $v\colon \mathcal{X} \to \mathbb{R}$ as $v(\boldsymbol{x}) \sim \mathcal{GP}\left(m(\boldsymbol{x}), k(\boldsymbol{x}, \boldsymbol{x}')\right)$, which is fully specified by its mean function $m\colon \mathcal{X} \to \mathbb{R}$ and covariance (kernel) function $k\colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}$. Without loss of generality, we consider zero prior mean. Our prior knowledge of the problem can shape the form of $k(\boldsymbol{x}, \boldsymbol{x}')$, considering that the covariance function must satisfy positive definiteness [167]. Given noisy measurements $w_j = v(\boldsymbol{x}_j) + \varepsilon_j, j \in \{1, \ldots, N\}$, which are corrupted by Gaussian noise $\varepsilon_j \sim \mathcal{N}(0, \sigma^2)$, a GP model can infer a posterior mean and variance for a test point $\boldsymbol{x}_t$ conditioned on the measurements

$$\begin{aligned} \mu(\boldsymbol{x}_t) &= \boldsymbol{w}^\top \left(K + \sigma_n^2 I\right)^{-1} \bar{K}^\top, \\ \sigma(\boldsymbol{x}_t)^2 &= k\left(\boldsymbol{x}_t, \boldsymbol{x}_t\right) - \bar{K}\left(K + \sigma_n^2 I\right)^{-1} \bar{K}^\top, \end{aligned} \tag{6.12}$$

where $\boldsymbol{w} \in \mathbb{R}^N$ is the vector of measurements w_j, $K \in \mathbb{R}^{N \times N}$ is the Gram matrix with elements $K_{ij} = k(\boldsymbol{x}_i, \boldsymbol{x}_j)$, and $\bar{K} = \begin{bmatrix} k(\boldsymbol{x}_t, \boldsymbol{x}_1) & \ldots & k(\boldsymbol{x}_t, \boldsymbol{x}_N) \end{bmatrix}^\top \in \mathbb{R}^N$.

6.3.2 Batch multi-output GP for piecewise residuals

Going back to the original problem, our objective is to build a Bayesian approximation with uncorrelated target functions d^V and d^h. We need to approximate each individual d_r^V and d_r^h, for $r \in \mathcal{I}$. Since the input data $(\boldsymbol{x}_j, \boldsymbol{u}_j)$ gathered from the same trajectory, we leverage from batch multi-output structure to learn d_r^V and d_r^h simultaneously. As d_r^V and d_r^h are uncorrelated, we consider an independent MOGP framework. This design facilitates

more computationally efficient GP training. We initially collect a dataset $\mathcal{D}$, as described in Section 6.2.3 and partition it into R datasets of the form

$$\mathcal{D}_r = \{((\boldsymbol{x}_j, \boldsymbol{u}_j), \omega_j^V, \omega_j^h) \mid ((\boldsymbol{x}_j, \boldsymbol{u}_j), \omega_j^V, \omega_j^h) \in \mathcal{D}, \boldsymbol{x}_j \in \mathcal{R}_r\},$$

for $r \in \mathcal{I}$, and denote the cardinality of each set by N_r. In a batch GP, each dataset or batch is associated with its own GP, and these GPs can capture the specific behavior and dependencies within each batch. Now, we employ MOGP to approximate R individual GPs for d_r^V and d_r^h simultaneously within a unified framework. Let $\boldsymbol{d}_r(\boldsymbol{x}, \boldsymbol{u}) = \begin{bmatrix} d_r^V(\boldsymbol{x}, \boldsymbol{u}) & d_r^h(\boldsymbol{x}, \boldsymbol{u}) \end{bmatrix}^\top \in \mathbb{R}^2$ be the vector-valued function which follows an MOGP as

$$\boldsymbol{d}_r(\boldsymbol{x}, \boldsymbol{u}) \sim \mathcal{GP}^r \left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} k_r^V & 0 \\ 0 & k_r^h \end{bmatrix} \right), \tag{6.13}$$

where the dependence to the pair $(\boldsymbol{x}, \boldsymbol{u})$ for the covariance functions is removed for simpler notation. As a result, training a MOGP model (6.13) leading to separate posterior distributions for target functions d_r^V and d_r^h:

$$\begin{aligned} d_r^V(\boldsymbol{x}, \boldsymbol{u}) &\sim \mathcal{N}(m_r^V(\mathcal{D}_r, \boldsymbol{x}, \boldsymbol{u}), {\sigma_r^V}^2(\mathcal{D}_r, \boldsymbol{x}, \boldsymbol{u})), \\ d_r^h(\boldsymbol{x}, \boldsymbol{u}) &\sim \mathcal{N}(m_r^h(\mathcal{D}_r, \boldsymbol{x}, \boldsymbol{u}), {\sigma_r^h}^2(\mathcal{D}_r, \boldsymbol{x}, \boldsymbol{u})). \end{aligned} \tag{6.14}$$

From (6.8), we can verify that the residuals are control-affine and can be characterized by

$$d_r^V = \begin{bmatrix} L_{f_r} V - L_{\hat{f}} V & L_{g_r} V - L_{\hat{g}} V \end{bmatrix} \begin{bmatrix} 1 \\ \boldsymbol{u} \end{bmatrix} = \boldsymbol{\varphi}_r^V \boldsymbol{y}, \tag{6.15}$$

$$d_r^h = \begin{bmatrix} L_{f_r} h - L_{\hat{f}} h & L_{g_r} h - L_{\hat{g}} h \end{bmatrix} \begin{bmatrix} 1 \\ \boldsymbol{u} \end{bmatrix} = \boldsymbol{\varphi}_r^h \boldsymbol{y}. \tag{6.16}$$

We denote the concatenation of one and $\boldsymbol{u}$ by $\boldsymbol{y} = \begin{bmatrix} 1, \boldsymbol{u}^\top \end{bmatrix}^\top \in \mathbb{R}^{m+1}$ and $\boldsymbol{\varphi}_r^V, \boldsymbol{\varphi}_r^h \in \mathbb{R}^{1 \times m+1}$. This prior knowledge of the problem can be embedded into the structure of the kernel function.

Proposition 6.1. *Let $\boldsymbol{x} \in \mathcal{R}_r$, $\boldsymbol{y} \in \mathbb{R}^{m+1}$, and define the input domain $\bar{\mathcal{X}}_r = \mathcal{R}_r \times \mathbb{R}^{m+1}$ for a given $r \in \mathcal{I}$. Then, consider the real-valued function $k_r \colon \bar{\mathcal{X}}_r \times \bar{\mathcal{X}}_r \to \mathbb{R}$, defined by*

$$k_r \left(\begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{y} \end{bmatrix}, \begin{bmatrix} \boldsymbol{x}' \\ \boldsymbol{y}' \end{bmatrix} \right) = \boldsymbol{y}^\top \Lambda_r(\boldsymbol{x}, \boldsymbol{x}') \boldsymbol{y}', \tag{6.17}$$

where $\Lambda_r(\boldsymbol{x}, \boldsymbol{x}') = \operatorname{diag}(\begin{bmatrix} k_r^1(\boldsymbol{x}, \boldsymbol{x}'), \ldots, k_r^{m+1}(\boldsymbol{x}, \boldsymbol{x}') \end{bmatrix})$ and $k_r^i \colon \mathcal{R}_r \times \mathcal{R}_r \to \mathbb{R}, i \in \{1, \ldots, m+1\}$, we denote the individual components k_r^i's as base kernels. If base kernels are all positive-definite, then k_r is a positive-definite kernel.

Proof. Since $k_r^i(\boldsymbol{x}, \boldsymbol{x}')$'s are positive definite kernels, then by the definition, there is a feature map $\varphi(\boldsymbol{x})$ such that $k_r^i(\boldsymbol{x}, \boldsymbol{x}') = \varphi_i^\top(\boldsymbol{x})\varphi_i(\boldsymbol{x}')$ for $i \in \{1, \dots, m+1\}$ [23]. Now, Let y^i and y'^i be the i^{th} element of the corresponding vectors. We have

$$\begin{aligned} k_r &= \boldsymbol{y}^\top \operatorname{diag}\left([\varphi_1^\top(\boldsymbol{x})\varphi_1(\boldsymbol{x}'), \dots, \varphi_{m+1}^\top(\boldsymbol{x})\varphi_{m+1}(\boldsymbol{x}')]\right) \boldsymbol{y}' \\ &= \sum_{i=1}^{m+1} y^i \varphi_i^\top(\boldsymbol{x})\varphi_i(\boldsymbol{x}')y'^i \\ &= \begin{bmatrix} y^1\varphi_1^\top(\boldsymbol{x}) & \dots & y^{m+1}\varphi_{m+1}^\top(\boldsymbol{x}) \end{bmatrix} \begin{bmatrix} y'^1\varphi_1(\boldsymbol{x}') \\ \vdots \\ y'^{m+1}\varphi_{m+1}(\boldsymbol{x}') \end{bmatrix} \\ &= \psi^\top\left(\begin{bmatrix}\boldsymbol{x}\\ \boldsymbol{y}\end{bmatrix}\right)\psi\left(\begin{bmatrix}\boldsymbol{x}'\\ \boldsymbol{y}'\end{bmatrix}\right), \end{aligned}$$

which again by the definition of kernels, proves that $k_r(\cdot, \cdot)$ is a positive-definite kernel. □

An immediate consequence of Proposition 6.1 is that k_r is the reproducing kernel of an RKHS $\mathcal{H}_{k_r}(\bar{\mathcal{X}})$. Also, this structure allows us to choose k_r^i's specifically for each region $r \in \mathcal{I}$.

We denote the input samples and domain by $(\boldsymbol{x}, \boldsymbol{y}) \in \bar{\mathcal{X}}$ as defined above. Let $X_r \in \mathbb{R}^{n \times N_r}$ and $Y_r \in \mathbb{R}^{(m+1)\times N_r}$ be matrices whose columns are vectors $\boldsymbol{x}_j$ and $\boldsymbol{y}_j$ of the corresponding dataset $\mathcal{D}_r$ and let outputs $\boldsymbol{\omega}_r \in \mathbb{R}^{N_r}$ be the stacked vector of the corresponding outputs to the states $\boldsymbol{x}_j \in \mathcal{R}_r$. Based on the collected batches, the MOGP model (6.13) equipped with the kernels of the form k_r in Proposition 6.1, gives the following expression for the posterior distribution of a test point $(\boldsymbol{x}_*, \boldsymbol{y}_*) \in \bar{\mathcal{X}}_r$:

$$m_r^* = \boldsymbol{\omega}_r^\top (k_r + \sigma_n^2 I)^{-1} \bar{K}_r^\top \boldsymbol{y}_*, \tag{6.18}$$

$$\sigma_r^{*2} = \boldsymbol{y}_*^\top (\Lambda_r(\boldsymbol{x}_*, \boldsymbol{x}_*) - \bar{K}_r(K_r + \sigma_n^2 I)^{-1}\bar{K}_r^\top)\boldsymbol{y}_*, \tag{6.19}$$

where K_r is the Gram matrix of $k_r(\cdot, \cdot)$ for the input data pair (X_r, Y_r), and $\bar{K}_r \in \mathbb{R}^{(m+1)\times N_r}$ is given by

$$\begin{aligned} \bar{K}_r &= \begin{bmatrix} \bar{\boldsymbol{k}}_r^1 & \bar{\boldsymbol{k}}_r^2 & \dots & \bar{\boldsymbol{k}}_r^{N_r} \end{bmatrix} \circ Y_r, \\ \bar{\boldsymbol{k}}_r^i &= \begin{bmatrix} k_r^1(x_*, x_i) & \dots & k_r^{m+1}(x_*, x_i) \end{bmatrix}^\top, i \in \{1, \dots, N_r\}, \end{aligned}$$

where $\circ$ denotes the element-wise multiplication (Hadamard product) of two matrices with identical dimensions. The posterior mean and variance of d_r^V and d_r^V are obtained by

substituting the output vectors $\boldsymbol{\omega}_r^V$ and $\boldsymbol{\omega}_r^h$ and their corresponding kernel structure into (6.18) and (6.19).

This result highlights a key advantage of using the proposed kernel. The resulting expression for the posterior mean $m_r^* = \boldsymbol{\mu}_r(\boldsymbol{x}_*)\boldsymbol{y}_*$ and the variance $\sigma_r^{*2} = \boldsymbol{y}_*^\top \Sigma_r(\boldsymbol{x}_*)\boldsymbol{y}_*$ are linear and quadratic in $\boldsymbol{y}_*$ (and the control input), respectively. We will exploit this feature in the next section to establish a convex GP-based safety filter.

6.3.3 Confidence bounds for the estimation of the effect of uncertainty

Although GP is inherently a probabilistic model, a high probability error bound can be derived for the distance between the true value and the GP prediction. This requires an additional assumption on d_r^V and d_r^h. Let d_r be either d_r^V or d_r^h, and $\boldsymbol{\varphi}_r$ be either $\boldsymbol{\varphi}_r^V$ or $\boldsymbol{\varphi}_r^h$ from (6.15) and (6.16).

Assumption 6.3. *We assume that each i^{th} element of $\boldsymbol{\varphi}_r$ is a member of $\mathcal{H}_{k_r^i}$ for $i \in \{1, \dots, m+1\}$ with bounded RKHS norm $\|\varphi_r^i\|_{k_r^i} \leq \eta_r$ for $i \in \{1, \dots, m+1\}$ within a given region $r \in \mathcal{I}$.*

Lemma 6.1. *([141]) Let Assumption 6.3 hold. Then, with a probability of at least $1-\delta$, the following holds*

$$|m_r^* - d_r(\boldsymbol{x}, \boldsymbol{u})| \leq \beta_r \sigma_r^*, \tag{6.20}$$

on $\mathcal{R}_r \subset \mathcal{X}, \delta \in (0,1)$ and

$$\beta_r = \sqrt{2\eta_r^2 + 300\kappa_{N_r+1}\log^3((N_r+1)/\delta)},$$

where κ_{N_r+1} is the maximum mutual information that can be obtained after getting N_r+1 data, and η_r is the upper bound of the corresponding RKHS norm, and m_r^* and σ_r^* are the posterior mean and standard deviation of a test point $(\boldsymbol{x}_*, \boldsymbol{y}_*) \in \bar{\mathcal{X}}_r$.

Based on the probabilistic bounds on (6.20) and the CLF and CBF constraints (6.9) and (6.10), we can conclude that the following hold with a probability of at least $1-\delta$

$$\dot{V}(\boldsymbol{x}, \boldsymbol{u}) \leq \hat{\dot{V}}(\boldsymbol{x}, \boldsymbol{u}) + \sum_{r=1}^{R} \delta_r (m_r^V + \beta_r \sigma_r^V), \tag{6.21}$$

$$\dot{h}(\boldsymbol{x}, \boldsymbol{u}) \geq \hat{\dot{h}}(\boldsymbol{x}, \boldsymbol{u}) + \sum_{r=1}^{R} \delta_r (m_r^h - \beta_r \sigma_r^h), \tag{6.22}$$

where $m_r^V(\boldsymbol{x},\boldsymbol{y})$ and $\sigma_r^V(\boldsymbol{x},\boldsymbol{y})$ are the mean and standard deviation of a query point derived from (6.18), (6.19). Incorporating (6.21) and (6.22) into the optimization problem (6.6), we have

$$\boldsymbol{u}_s = \underset{(\boldsymbol{u},d)\in\mathbb{R}^{m+1}}{\arg\min} \ \|\boldsymbol{u}\|_2^2 + \rho d^2 \tag{6.23a}$$

$$\text{s.t.} \quad \hat{\dot{V}}(\boldsymbol{x},\boldsymbol{u}) + \sum_{r=1}^{R} \delta_r (m_r^V + \beta_r \sigma_r^V) + \lambda V(\boldsymbol{x}) \leq d, \tag{6.23b}$$

$$\hat{\dot{h}}(\boldsymbol{x},\boldsymbol{u}) + \sum_{r=1}^{R} \delta_r (m_r^h - \beta_r \sigma_r^h) + \alpha(h(\boldsymbol{x})) \geq 0, \tag{6.23c}$$

Note that the constraint in (6.23) is constructed regardless of the underlying true dynamics.

Theorem 6.1. *The optimization problem (6.23) is convex and can be converted into the standard SOCP of the form (6.24), if m_r^V, m_r^h, and σ_r^V, σ_r^h satisfy (6.18) and (6.19), respectively.*

$$\begin{bmatrix} \boldsymbol{u}_{socp}^\top & d_* & t_* \end{bmatrix}^\top = \underset{\boldsymbol{z}}{\arg\min} \ \boldsymbol{f}^\top \boldsymbol{z}$$
$$s.t. \ \sum_{r=1}^{R} \delta_r \left\| M_r^i \boldsymbol{z} + \boldsymbol{n}_r^i \right\|_2 \leq \sum_{r=1}^{R} \delta_r ({\boldsymbol{p}_r^i}^\top \boldsymbol{z} + q_r^i), \tag{6.24}$$

for $i = 1, \ldots, n_c$, where $\boldsymbol{z} \in \mathbb{R}^{n_z}$ and $\boldsymbol{f} \in \mathbb{R}^{n_z}$, and the matrix M_r^i and $\boldsymbol{n}_r^i, \boldsymbol{p}_r^i, q_r^i$ have appropriate dimensions for n_c constraints.

Proof. Let's denote the objective function of (6.23) by $J = \|\boldsymbol{u}\|_2^2 + \rho d^2$. Let $\boldsymbol{p} = \begin{bmatrix} \boldsymbol{u}^\top & d \end{bmatrix}^\top \in \mathbb{R}^{m+1}$. Then, we can rewrite J as the equivalent form of $J_1 = \boldsymbol{p}^\top Q \boldsymbol{p}$, where Q is a positive definite diagonal matrix with matrix square root E. Then, we convert J_1 into $J_2 = \|E\boldsymbol{p}\|_2^2$. Since the Euclidean norm is always positive, we can consider the equivalent objective function $J_3 = \|E\boldsymbol{p}\|_2$ and set $\|E\boldsymbol{p}\|_2 \leq t$, where $t \in \mathbb{R}$ is an auxiliary variable. Then, we convert J_3 to a second-order cone constraint (6.25). Now, we need to solve a new minimization problem with a new augmented variable $\boldsymbol{z} = \begin{bmatrix} \boldsymbol{u}^\top & d & t \end{bmatrix}^\top \in \mathbb{R}^{m+2}$ as

$$\begin{aligned} &\min \begin{bmatrix} \mathbf{0}_{m+1}^\top & 1 \end{bmatrix} \boldsymbol{z} \\ &\text{s.t.} \ \| \underbrace{\begin{bmatrix} E & \mathbf{0}_{m+1} \end{bmatrix}}_{:=M_r^1} \boldsymbol{z} \| \leqslant \underbrace{\begin{bmatrix} \mathbf{0}_{m+1}^\top & 1 \end{bmatrix}}_{:={\boldsymbol{p}_r^1}^\top} \boldsymbol{z}, \end{aligned} \tag{6.25}$$

where $\mathbf{0}_{m+1} \in \mathbb{R}^{m+1}$ is a vector of zeros and $\boldsymbol{f} = \begin{bmatrix}\mathbf{0}_{m+1}^\top & 1\end{bmatrix}^\top$.

Next, we need to show that constraints (6.23b) and (6.23c) are second-order cone constraints. We need to consider the constraints for a given region index $r \in \mathcal{I}$ as at each time step one region is active according to the defined switching signal δ_r in Section 6.2. As the proof is the same for both constraints, we consider the CBF constraint (6.23c). Note that based on (6.18), we have $m_r^h(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{\mu}_r^h(\boldsymbol{x})\boldsymbol{y}$, which can be written as

$$m_r^h(\boldsymbol{x}, \boldsymbol{y}) = \mu_r^{h1} + \boldsymbol{\mu}_r^{hm}\boldsymbol{u},$$

where the notation μ_r^{h1} refers the first element and $\boldsymbol{\mu}_r^{hm}$ refers the last m elements of the row vector $\boldsymbol{\mu}_r^h \in \mathbb{R}^{1\times(m+1)}$. Also, from (6.16), we know that $\boldsymbol{\varphi}_r^h$ is control affine, i.e. $\boldsymbol{\varphi}_r^h = \varphi_r^{hf} + \boldsymbol{\varphi}_r^{hg}\boldsymbol{u}$, where $\varphi_r^{hf} \in \mathbb{R}$ and the row vectors $\boldsymbol{\varphi}_r^{hg} \in \mathbb{R}^{1\times m}$. Hence, the right-hand side of the inequality in (6.23c) is affine in $\boldsymbol{u}$ as desired.

Based on (6.19), we have that ${\sigma_r^h}^2(\boldsymbol{x}, \boldsymbol{y}) = \boldsymbol{y}^\top \Sigma_r^h(\boldsymbol{x})\boldsymbol{y}$, where $\Sigma_r^h \in \mathbb{R}^{(m+1)\times(m+1)}$. Since k_r is a valid kernel and $\sigma_n > 0$, Σ_r^h is positive definite. Then, we have

$$\sigma_r^h(\boldsymbol{x}, \boldsymbol{y}) = \sqrt{\boldsymbol{y}^\top \Sigma_r^h \boldsymbol{y}} = \sqrt{\boldsymbol{y}^\top L_r^\top L_r \boldsymbol{y}} = \|L_r \boldsymbol{y}\|_2,$$

where $L_r \in \mathbb{R}^{(m+1)\times(m+1)}$ is the matrix square root of Σ_r^h. By the definition $\boldsymbol{y} = \begin{bmatrix}1 & \boldsymbol{u}^\top\end{bmatrix}^\top$, we have $\sigma_r^h(\boldsymbol{x}, \boldsymbol{y}) = \left\|\boldsymbol{l}_r^1 + L_r^m \boldsymbol{u}\right\|_2$, where the notation $\boldsymbol{l}_r^1$, and L_r^m refers to the first column and the last m columns of the matrix L_r, respectively. Now, we can rewrite the safety certificate (6.23c) as a second-order cone constraint of the form

$$\left\|A_r^h(\boldsymbol{x})\boldsymbol{u} + \boldsymbol{b}_r^h(\boldsymbol{x})\right\|_2 \leq \boldsymbol{c}_r^h(\boldsymbol{x})\boldsymbol{u} + d_r^h(\boldsymbol{x}), \tag{6.26}$$

where

$$\begin{aligned} A_r^h(\boldsymbol{x}) &= \beta_r L_r^m \in \mathbb{R}^{(m+1)\times(m)}, \\ \boldsymbol{b}_r^h(\boldsymbol{x}) &= \beta_r \boldsymbol{l}_r^1 \in \mathbb{R}^{m+1}, \\ \boldsymbol{c}_r^h(\boldsymbol{x}) &= \boldsymbol{\varphi}_r^{hg} + \boldsymbol{\mu}_r^{hm} \in \mathbb{R}^{1\times m} \\ d_r^h(\boldsymbol{x}) &= \varphi_r^{hf} + \mu_r^{h1} \in \mathbb{R}. \end{aligned} \tag{6.27}$$

It can be expressed in terms of the new variable $\boldsymbol{z}$ as

$$\| \underbrace{\begin{bmatrix}A_r^h(\boldsymbol{x}) & \mathbf{0}_{m+1}\end{bmatrix}}_{:=M_r^2} \boldsymbol{z} + \underbrace{\boldsymbol{b}_r^h(\boldsymbol{x})}_{:=\boldsymbol{n}_r^2} \|_2 \leq \underbrace{\begin{bmatrix}\boldsymbol{c}_r^h(\boldsymbol{x}) & 0\end{bmatrix}}_{:=\boldsymbol{p}_r^{2\top}} \boldsymbol{z} + \underbrace{d_r^h(\boldsymbol{x})}_{:=q_r^2}.$$

The same proof holds for the CLF constraint (6.23b). Adding the resulting constraints to (6.25) leads to (6.24), which concludes the proof. □

6.3.4 Feasibility analysis

In this section, we will analyze the point-wise feasibility of the SOCP (6.24). As the CLF is considered as a soft constraint, only the CBF constraint restricts the feasibility. Intuitively, if the batch MOGP prediction is less accurate, the resulting standard deviation $\sigma_r^h(\boldsymbol{x}, \boldsymbol{y})$ will be large, leading to a more conservative approach. In particular, it restricts the space on which a safe control input can be selected, which may make the SOCP infeasible. In the following, we theoretically analyze the conditions for feasibility.

Theorem 6.2. *Given a state $\boldsymbol{x} \in \mathcal{R}_r$, the SOCP (6.24) is feasible if and only if there exists a control input $\boldsymbol{u} \in \mathbb{R}^m$ that satisfies the following conditions*

$$\begin{bmatrix} d_r^h(\boldsymbol{x}) & \boldsymbol{c}_r^h(\boldsymbol{x}) \end{bmatrix} \begin{bmatrix} 1 \\ \boldsymbol{u} \end{bmatrix} \geq 0, \tag{6.28}$$

$$\begin{bmatrix} 1 & \boldsymbol{u}^\top \end{bmatrix} S_r(\boldsymbol{x}) \begin{bmatrix} 1 \\ \boldsymbol{u} \end{bmatrix} \leq 0, \tag{6.29}$$

where $S_r \in \mathbb{R}^{(m+1)\times(m+1)}$ is of the form

$$\begin{aligned} S_r(\boldsymbol{x}) &= \begin{bmatrix} S_r^1(\boldsymbol{x}) & S_r^2(\boldsymbol{x}) \\ {S_r^2}^\top(\boldsymbol{x}) & S_r^3(\boldsymbol{x}) \end{bmatrix}, \\ S_r^1(\boldsymbol{x}) &= {\boldsymbol{b}_r^h}^\top(\boldsymbol{x})\boldsymbol{b}_r^h(\boldsymbol{x}) - {d_r^h}^\top(\boldsymbol{x})d_r^h(\boldsymbol{x}), \\ S_r^2(\boldsymbol{x}) &= {\boldsymbol{b}_r^h}^\top(\boldsymbol{x})A_r^h(\boldsymbol{x}) - {d_r^h}^\top(\boldsymbol{x})\boldsymbol{c}_r^h(\boldsymbol{x}), \\ S_r^3(\boldsymbol{x}) &= {A_r^h}^\top(\boldsymbol{x})A_r^h(\boldsymbol{x}) - {\boldsymbol{c}_r^h}^\top(\boldsymbol{x})\boldsymbol{c}_r^h(\boldsymbol{x}), \end{aligned} \tag{6.30}$$

with $A_r^h(\boldsymbol{x}), \boldsymbol{b}_r^h(\boldsymbol{x}), \boldsymbol{c}_r^h(\boldsymbol{x})$, and $d_r^h(\boldsymbol{x})$ are defined in (6.27).

Proof. In order to analyze the feasibility of the SOCP (6.24), we need to verify that the safety second-order cone constraint (6.26) is feasible. Since the left-hand side is always non-negative, the right-hand side must be also non-negative, which leads to the first condition.

Now, since both sides of (6.26) are non-negative, we can take squares of both sides and write the resulting expression in quadratic form. and collect all the terms on the left. Factorizing the similar terms gives the second condition. □

Based on this result, we obtain a necessary condition for point-wise feasibility as stated in the following:

Corollary 6.2. *Given $\boldsymbol{x} \in \mathcal{R}_r$, if the SOCP (6.24) is feasible, then the following condition must be satisfied*

$$1 - \frac{1}{\beta_r^2} \boldsymbol{\phi}_r {\Sigma_r^h}^{-1} \boldsymbol{\phi}_r^\top \leq 0, \tag{6.31}$$

where $\boldsymbol{\phi}_r = \begin{bmatrix} \varphi_r^{hf} + \mu_r^{h1} & \boldsymbol{c}_r^h(\boldsymbol{x}) \end{bmatrix} \in \mathbb{R}^{1\times(m+1)}$.

Proof. We prove this by contradiction. Let's assume that there exists a solution $\boldsymbol{u} \in \mathbb{R}^m$ of the SOCP (6.24), which satisfies $1 - \frac{1}{\beta_r^2} \boldsymbol{\phi_r} {\Sigma_r^h}^{-1} \boldsymbol{\phi}_r^\top > 0$. Let's define the matrix M_r partitioned as

$$M_r = \begin{bmatrix} 1 & \boldsymbol{\phi}_r \\ \boldsymbol{\phi}_r^\top & \beta_r^2 \Sigma_r^h(\boldsymbol{x}) \end{bmatrix} \in \mathbb{R}^{(m+2)\times(m+2)}.$$

The Schur complements of M_r is obtained by

$$\begin{cases} M_r/1 = \beta_r^2 \Sigma_r^h(\boldsymbol{x}) - \boldsymbol{\phi}_r^\top \boldsymbol{\phi}_r, \\ M_r/\beta_r^2 \Sigma_r^h(\boldsymbol{x}) = 1 - \frac{1}{\beta_r^2} \boldsymbol{\phi}_r {\Sigma_r^h}^{-1} \boldsymbol{\phi}_r^\top. \end{cases}$$

We can easily verify that $\beta_r^2 \Sigma_r^h = \begin{bmatrix} \beta_r \boldsymbol{l}_r^1 & A_r^h(\boldsymbol{x}) \end{bmatrix}^\top \begin{bmatrix} \beta_r \boldsymbol{l}_r^1 & A_r^h(\boldsymbol{x}) \end{bmatrix}$ and $M_r/1 = S_r(\boldsymbol{x})$. We know that $\Sigma_r^h(\boldsymbol{x})$ is SPD. So, M_r is also symmetric and we can characterize its definiteness by Schur complement theorem. Since its lower right block is positive definite and its corresponding Schur complement $M_r/\beta_r^2 \Sigma_r^h(\boldsymbol{x}) > 0$ by our assumption, based on [27], we can conclude that M_r is also positive definite.

Now, given that M_r is SPD, and its upper left block is always positive definite, we can conclude that $M_r/1 = S_r(\boldsymbol{x})$ must be positive definite, which is a contradiction to the condition (6.29). Thus, $1 - \frac{1}{\beta_r^2} \boldsymbol{\phi}_r {\Sigma_r^h}^{-1} \boldsymbol{\phi}_r^\top \leq 0$ holds. □

Theorem 6.2 also provides the theoretical background to obtain a sufficient condition of feasibility.

Corollary 6.3. *Given $\boldsymbol{x} \in \mathcal{R}_r$, the SOCP (6.24) is feasible at $\boldsymbol{x}$, if $S_r^3(\boldsymbol{x})$ is negative definite.*

Proof. From (6.30), we know that $S_r^3(\boldsymbol{x})$ is a symmetric matrix, thus it has real eigenvalues. Let's denote its maximum eigenvalue and the corresponding eigenvector by λ_r^m, $\boldsymbol{e}_r^m$. Then, by the assumption that $S_r^3(\boldsymbol{x})$ is negative definite, we can conclude that $\lambda_r^m < 0$ and ${\boldsymbol{e}_r^m}^\top S_r^3(\boldsymbol{x}) \boldsymbol{e}_r^m < 0$. Substituting (6.30) into the former, we have

$${\boldsymbol{e}_r^m}^\top {A_r^h}^\top(\boldsymbol{x}) A_r^h(\boldsymbol{x}) \boldsymbol{e}_r^m - (\boldsymbol{c}_r^h(\boldsymbol{x}) \boldsymbol{e}_r^m)^\top (\boldsymbol{c}_r^h(\boldsymbol{x}) \boldsymbol{e}_r^m) < 0.$$

Since $A_r^h(\boldsymbol{x})^\top A_r^h(\boldsymbol{x})$ is positive definite, $\boldsymbol{c}_r^h(\boldsymbol{x})\boldsymbol{e}_r^m \neq 0$ must hold. We take a control input in the direction of this eigenvector as $\boldsymbol{u}_e = \alpha\,\mathrm{sgn}(\boldsymbol{c}_r^h(\boldsymbol{x})\boldsymbol{e}_r^m)\cdot \boldsymbol{e}_r^m, \alpha > 0$. Next, we need to check if the resulting control input can satisfy the conditions of Theorem 6.2. Substituting $\boldsymbol{u}_e$ in (6.29), we have

$$S_r^1 + 2\alpha S_r^2((\boldsymbol{c}_r^h(\boldsymbol{x})\boldsymbol{e}_r^m)\cdot \boldsymbol{e}_r^m) + \alpha^2 \boldsymbol{e}_r^{m\top} S_r^3(\boldsymbol{x})\boldsymbol{e}_r^m.$$

By choosing large enough α, the above statement can be made negative since $\boldsymbol{e}_r^{m\top} S_3(\boldsymbol{x})\boldsymbol{e}_r^m < 0$. Also, substituting $\boldsymbol{u}_e$ into (6.28), we have

$$\boldsymbol{c}_r^h(\boldsymbol{x})\boldsymbol{u}_e + (\varphi_r^{hf} + \mu_r^{h1}) = \alpha|\boldsymbol{c}_r^h(\boldsymbol{x})\boldsymbol{e}_r^m| + (\varphi_r^{hf} + \mu_r^{h1}).$$

Again, by choosing a large enough α, we can make the above expression positive, which concludes the proof. □

6.3.5 Data collection and model fitting

We use an episodic data collection method. In the first episode, we run an initial roll-out using the CLF and CBF designed based on the nominal system. Using nominal system (6.7), we run QP controller (6.6) with the nominal values of the time derivative of $V(\boldsymbol{x})$ and $h(\boldsymbol{x})$. Then, we record the system's trajectory until it reaches an unsafe state. Next, we partition the resulting dataset into R datasets $\mathcal{D}_r$ as described in Section 6.3.2. Subsequently, we fit MOGPs configured with the kernel functions k_r to batches $\mathcal{D}_r$, for $r = 1, \ldots, R$. To use (6.18) and (6.19) for prediction, we need to infer the hyperparameters $\boldsymbol{\theta}_r$ for each individual kernel $k_r^i, i \in \{1, \ldots, m+1\}$ in the structure of k_r. Each set of parameters is obtained by minimizing the negative log marginal likelihood [97] as

$$\boldsymbol{\theta}_r^{opt} = \arg\min_{\boldsymbol{\theta}_r} -\log p(\boldsymbol{\omega}_r \mid \bar{X}_r, \boldsymbol{\theta}_r), \tag{6.32}$$

where for the $J_{\boldsymbol{\theta}_r} = -\log p(\boldsymbol{\omega}_r \mid \bar{X}_r, \boldsymbol{\theta}_r)$, we have

$$\begin{aligned} J_{\boldsymbol{\theta}_r} =& \frac{1}{2}\boldsymbol{\omega_r}^{\mathrm{T}} \left[K_r(\bar{X}_r, \bar{X}_r) + \sigma_n^2 I\right]^{-1} \boldsymbol{\omega_r} \\ &+ \frac{1}{2}\log\left|K_r(\bar{X}_r, \bar{X}_r) + \sigma_n^2 I\right| + \frac{N_r}{2}\log 2\pi, \end{aligned}$$

where $\bar{X}_r = (X_r, Y_r)$ such that $((X_r, Y_r), \boldsymbol{\omega}_r) \in \mathcal{D}_r$. $\boldsymbol{\omega}_r$ is either the vector of outputs $\boldsymbol{\omega}_j^V$ or $\boldsymbol{\omega}_j^h$ for approximating d_r^V or d_r^h, respectively. Then we use the uncertainty-aware SOCP optimization (6.23) with a high probability bound ($1 - \delta = 0.95$) and run the system with

the proposed method until the system reaches an unsafe state or the problem becomes infeasible. The data collected during each episode will be added to the dataset and the hyperparameters $\boldsymbol{\theta}_r$ will be optimized using (6.32) with the new dataset. We repeat this process until the simulation is completed without encountering any unsafe behavior.

6.4 Simulation Results

In this section, we highlight the effectiveness of the proposed method by applying it to a switching adaptive cruise control system that is moving on varying road conditions. For comparison, we also implemented a baseline GP-CBF-CLF-SOCP [34] that fits a single GP to piecewise residuals and a nominal QP controller. The GP models were trained using GPyTorch [61].

Consider a switching adaptive cruise control system modeled by

$$\begin{aligned} \dot{\boldsymbol{x}} &= \sum_{r=1}^{R} \delta_r \left(f_r(\boldsymbol{x}) + g_r(\boldsymbol{x}) \right) u, \\ f_r(\boldsymbol{x}) &= \begin{bmatrix} x_2 \\ -\frac{F_r(x_2)}{m} \end{bmatrix}, g_r(\boldsymbol{x}) = \begin{bmatrix} 0 \\ \frac{c_r}{m} \end{bmatrix}, \end{aligned} \tag{6.33}$$

where the system state $\boldsymbol{x} = \begin{bmatrix} x_1 & x_2 \end{bmatrix}^\top \in \mathcal{X}$ containing the position of the ego car x_1, and its forward velocity x_2 in the state space $\mathcal{X} = [0, 700] \times [0, 30]$. The control input is the wheel force $u \in \mathbb{R}$. The mass of the ego car is denoted by m and the rolling resistance is modeled by $F_r(x_2) = f_r^0 + f_r^1 x_2 + f_r^2 x_2^2$. The switching behavior of the system arises from the changing road conditions which impact the rolling resistance $F_r(x_2)$ and traction control of the vehicle c_r. As the vehicle encounters different road surfaces $F_r(x_2)$ and c_r will change, leading to variations in the system dynamics. We consider hyper-rectangular partitions of $\mathcal{X}$ with bounds $\mathcal{R}_1 = ([0, 15) \cup (25, 700]) \times [0, 30]$, and $\mathcal{R}_2 = [15, 25] \times [0, 30]$ that represent two different road conditions. The state-dependent switching signal δ_r in (6.1) determines the active region index at each time step. We only have access to the nominal model and the true system is unknown in both operating regions. The parameters for each model are shown in Table 6.1.

The distance between the ego car and the front car, denoted by z, is governed by the equation $\dot{z} = v_0 - x_2$, where $v_0 = 10\, m/s$ represents the constant velocity of the front car. The control objective is to achieve a target speed of v_d while ensuring a safe distance from the front vehicle. To stabilize the velocity at the desired value, a CLF $V(\boldsymbol{x}) = (x_2 - v_d)^2$

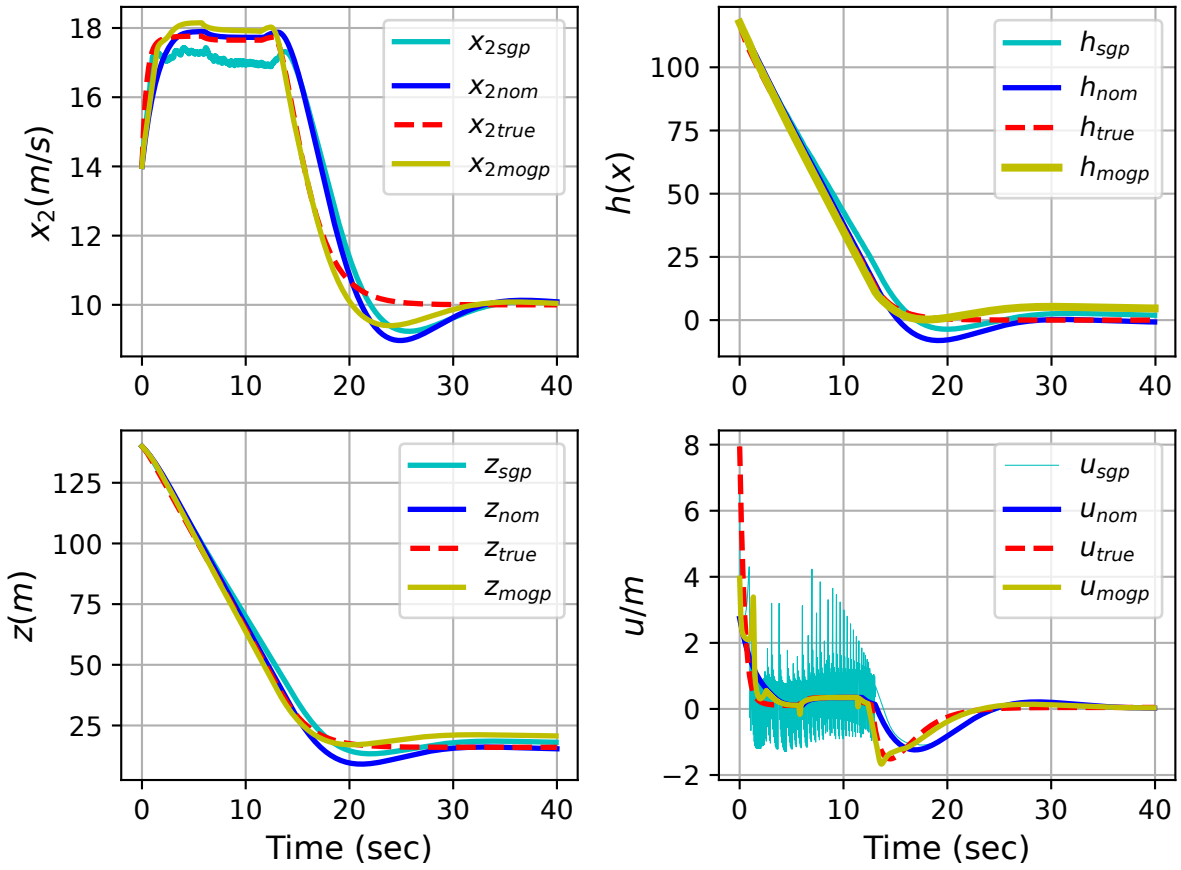


Figure 6.1: Comparison of the MOGP-SOCP (yellow), single GP-SOCP (cyan), nominal QP (blue), and true design (red dashed). System state x_2 and distance between two cars z (left column). The CBF and control input u (right column).

Table 6.1: Parameters for the nominal and true system in $\mathcal{R}_1, \mathcal{R}_2$

	$m(kg)$	f_1^0	f_1^1	f_1^2	c_r
True $\mathcal{R}_1$	3300	0.2	10	0.5	1
True $\mathcal{R}_2$	3300	1	50	4.5	0.5
Nominal	1050	0.1	15	2.25	1

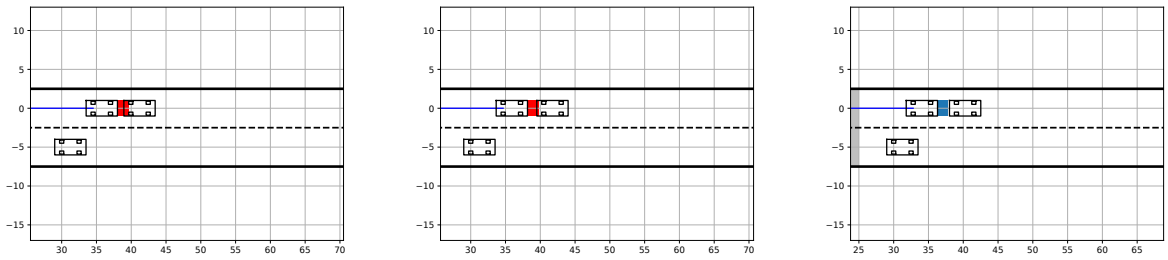


Figure 6.2: Snapshots of the simulation to show the unsafe (red color) and safe (green color) distance between cars in different implementations. The nominal QP controller (left) and single GP-SOCP controller (middle) violate the safety distance. MOGP-based controller (right) ensures safety. See this link for the simulation video.

is considered. Additionally, a CBF $h(\boldsymbol{x}) = z - T_h x_2$, with $T_h = 1.6$, is utilized to maintain a safe distance proportional to the current velocity.

We selected infinitely differentiable squared exponential kernel [167], for $k_i, i = 1, 2$ base kernels in the configuration of k_r. We used the process in Section 6.3.5 to collect the dataset and optimize the batch MOGP model hyperparameters, which collected a total number of 390 data samples in the initial roll-out and a total of 999 data samples in all episodes. The simulation is started from the initial state $\boldsymbol{x}_0 = \begin{bmatrix} 0 & 14 \end{bmatrix}^\top$ and $z_0 = 140\,m$. The ego car starts in the region $\mathcal{R}_1$ from a safe distance behind the front car. First, the controller increases the velocity to converge to the desired speed resulting in a decrease in z. Then, the ego car transits into region $\mathcal{R}_2$ and approaches the front car which activates the safety constraint (6.23c). Consequently, the controller must reduce the velocity to keep the safe distance. Due to the adverse effect of the piecewise residuals, the Single GP-SOCP and nominal QP controller cannot maintain the safe distance. However, the proposed MOGP-SOCP controller effectively avoids this unsafe behavior by leveraging the batch MOGP design. A simulation video showing the performance of the nominal and MOGP-SOCP controller is provided. Figure 6.2 shows a snapshot of a simulation when all controllers result in the closest distance to the front car. It can be verified that only MOGP-SOCP controllers can keep a safe distance during the simulation.

In Figure 6.1, we aim to compare the performance of the proposed method with the baseline and nominal controllers. For your reference, we provided the trajectories of the oracle true design. It has been illustrated in Figure 6.1 that the single GP-SOCP controller did not converge to the desired velocity, while MOGP-SOCP and nominal QP controllers could reach v_d in the first 4 seconds. Then, the ego car transited to $\mathcal{R}_2$ at $t = 6\,s$ which

affects the rate of change of the velocity. As it approaches the front car, the controllers must decrease the velocity in response to remain safe. The CBF $h(\boldsymbol{x})$ and control input u are also illustrated in this figure. This plot shows that the single GP-SOCP and nominal QP controllers violate the safety condition approximately at $t = 15\,s$ and $t = 17\,s$, respectively. However, the MOGP-SOCP controller could maintain the safe distance the front car during the simulation. Also, it can be verified from the control input trajectory that the MOGP-SOCP controller could recover the true system's performance. However, the single GP-SOCP controller does not generate a smooth control input which is a result of poor uncertainty quantification between the switching times. A simulation video for the results is available in this link.

6.5 Chapter Summary

In this chapter, a batch MOGP framework is developed to approximate the effect of the uncertainty on the CLF and CBF constraints. The switching dynamics of the true system result in piecewise residuals in CLF and CBF constraints corresponding to R regions that cover the state space. A batch MOGP model is designed to capture the piecewise residuals in each region. Then, the resulting constrained optimization problem with the uncertainty-aware chance constraint is converted into a SOCP. This optimization is proven to be convex and can be solved in real time. Finally, the feasibility of the resulting SOCP is addressed.

Chapter 7

Conclusions and Future Works

7.1 Conclusion

In this thesis, a formal perspective of the safety-critical control methods for autonomous mobile robots is presented. In our design, provably safe control, robustness to the uncertainty, and computational efficiency are considered as the top priorities. Our method is established based on set invariance, which provides a theoretical foundation to ensure safety for dynamical systems. Among the state-of-the-art safety implementation methods via set invariance, CBF is selected due to its capabilities such as high scalability, smooth transition to safe behavior, ease of synthesis and implementation, in comparison with its counterparts.

By observing the limitation of applying CBFs to multi-body mobile robots, we developed a multiple CBFs scheme which enforces several safety constraints with high relative degrees. This control structure is essential in many challenging robotic applications that have several components. Commonly, this problem is solved by considering all the components as one unit and the safety is guaranteed for the whole system, which often leads to a conservative approach. Our method formalizes the design of multiple CBFs in a systematic fashion such that all of the control inputs contribute to the safety of the system components, resulting a high fidelity design. It also provides theoretical safety guarantee for the intersection of the individual safe sets, proving the safety of the whole system. A reactive obstacle avoidance technique is designed for tractor-trailer systems which are a class of multi-body mobile robots based on the proposed multiple CBFs scheme.

CBFs is a model-based method, meaning lack of perfect knowledge of the system dynamics directly affects the safety of the system. One approach to tackle this issue is using

machine learning techniques. We integrate the flexibility of the data-driven methods with provably safe design of CBFs. We use GP for approximation of the effect of uncertainty on safety constraint which has properties suitable for integrating with rigorous formal safety guarantees. However, GP provides posterior distributions for approximations which results in chance constraints. A special covariance structure is proposed, which together with the bounds on the posterior mean and variance enables us to convert the chance constraint to a second-order cone constraint. Integrating the resulting safety constraint leads to a convex optimization which can be solved in real time. A discussion on feasibility analysis of the optimization problem is provided and the necessary and sufficient conditions of feasibility are obtained. Another approach is to fit a GP model to the system dynamics and use the approximation to derive the safety constraints. A comparative study is done which shows the similarities and differences of both methods.

In another area of research, safety and stability of the switching systems in the presence of uncertainty is considered, which enables the deployment of CBFs to systems that a nominal model of a true underlying switching system is available. It has been shown that the effect of uncertainty on the CBF and CLF form piece-wise residuals for each switching surface. A multi-output structure is used to fit GPs to the effect of uncertainty on CLF and CBF design. A particular form of covariance function is designed to convert the set of chance constraints to second-order cone constraint, which results in a convex optimization problem. Similar to the HOCBF design, the necessary and sufficient conditions of feasibility are derived. It is worth mentioning that obtaining these conditions in both methodologies are important in the sense that they can provide a direction to collect the dataset which ensures persistent excitation. Thus, as long as a safe solution exists the GP model provides posterior distribution with high level of accuracy to obtain the safe control input.

7.2 Future Works

Based on the key contributions of this thesis, we outline directions for future research.

CBF-based safety filters rely on the assumption that the solution to the QP optimization, representing the safe input, is Lipschitz continuous. This property is crucial to ensuring the Lipschitz continuity of the closed-loop system. Investigating the proof of Lipschitz continuity of the safe input, especially for systems with constrained inputs, would be highly beneficial, as it would enable the application of provably safe multiple CBFs to a broader range of practical problems.

In the proposed method, a condition to certify that all control inputs contribute to the safety of the system is stated using MIMO feedback linearization technique. However, the

current approach does not account for model uncertainty. Future research should focus on developing a systematic design procedure for multiple CBFs under bounded uncertainty. A key challenge in this area is analytically characterizing the nonlinear and time-varying impact of uncertainty on the singularity of the decoupling matrix in the feedback linearized system.

A general assumption in CBF-based safety filter design is that the relative degree with respect to the barrier function is well-defined and remains constant across the state domain.However, in practice, the CBF method, originally developed for continuous-time systems, is implemented in discrete time. This mismatch between design and implementation introduces challenges, as it may lead to temporary violations of the relative degree conditions.As a result, the relative degree can change over finite time intervals, potentially invalidating the safety constraint and causing issues such as chattering or safety violations. Future research on GP-based HOCBFs could focus on extending the framework to accommodate systems with varying relative degrees, which requires a thorough characterization of the impact on residual errors.

One of the challenges limiting the application of GP-based HOCBFs in real-world robotics is the potential infeasibility of the SOCP. This issue can arise when the collected dataset lacks persistent excitation which results in high posterior variance. To address this, we derived the necessary and sufficient conditions for the feasibility of GP-based HOCBFs. In the case of infeasibility, these constraints can be utilized to identify an input domain that resolves the persistent excitation problem. Once this domain is determined, the data collection process can be repeated to generate a new training set, enabling improved posterior inference and enhancing the feasibility of the safety filter.

In designing the GP-based CBF controller for switching systems, we assumed non-overlapping regions to ensure well-posedness. Future work could focus on analyzing the impact of uncertainty in scenarios where this assumption does not hold, providing insights into more general cases.

References

[1] Mohammad Aali and Jun Liu. Multiple control barrier functions: An application to reactive obstacle avoidance for a multi-steering tractor-trailer system. In *IEEE Conference on Decision and Control (CDC)*, pages 6993–6998, 2022.

[2] Mohammad Aali and Jun Liu. Multiple control barrier functions: An application to reactive obstacle avoidance for a multi-steering tractor-trailer system. *arXiv preprint arXiv:2209.05156*, 2022.

[3] Mohammad Aali and Jun Liu. Learning high-order control barrier functions for safety-critical control with Gaussian processes. In *American Control Conference (ACC)*, 2024.

[4] Mohammad Aali and Jun Liu. Learning piecewise residuals of control barrier functions for safety of switching systems using multi-output Gaussian processes. In *European Control Conference (ECC)*, 2024.

[5] National Highway Traffic Safety Administration et al. Critical reasons for crashes investigated in the national motor vehicle crash causation survey. *Washington, DC: US Department of Transportation*, 2:1–2, 2015.

[6] Ayush Agrawal and Koushil Sreenath. Discrete control barrier functions for safety-critical control of discrete systems with application to bipedal robot navigation. In *Robotics: Science and Systems*, volume 13, pages 1–10. Cambridge, MA, USA, 2017.

[7] Anayo K Akametalu, Shromona Ghosh, Jaime F Fisac, and Claire J Tomlin. A minimum discounted reward Hamilton-Jacobi formulation for computing reachable sets. *arXiv preprint arXiv:1809.00706*, 2018.

[8] Anil Alan, Andrew J Taylor, Chaozhe R He, Aaron D Ames, and Gábor Orosz. Control barrier functions and input-to-state safety with application to automated vehicles. *IEEE Transactions on Control Systems Technology*, 31(6):2744–2759, 2023.

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

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

[11] Aaron D Ames, Kevin Galloway, Koushil Sreenath, and Jessy W Grizzle. Rapidly exponentially stabilizing control Lyapunov functions and hybrid zero dynamics. *IEEE Transactions on Automatic Control*, 59(4):876–891, 2014.

[12] Aaron D Ames, Jessy W Grizzle, and Paulo Tabuada. Control barrier function based quadratic programs with application to adaptive cruise control. In *IEEE Conference on Decision and Control (CDC)*, pages 6271–6278, 2014.

[13] Aaron D Ames, Xiangru Xu, Jessy W Grizzle, and Paulo Tabuada. Control barrier function based quadratic programs for safety critical systems. *IEEE Transactions on Automatic Control*, 62(8):3861–3876, 2016.

[14] Hirokazu Anai and Volker Weispfenning. Reach set computations using real quantifier elimination. In *International Workshop on Hybrid Systems: Computation and Control*, pages 63–76. Springer, 2001.

[15] Zvi Artstein. Stabilization with relaxed controls. *Nonlinear Analysis: Theory, Methods and Applications*, 7(11):1163–1173, 1983.

[16] Jean-Pierre Aubin, Alexandre M Bayen, and Patrick Saint-Pierre. *Viability theory: new directions*. Springer Science & Business Media, 2011.

[17] Andrea Bajcsy, Somil Bansal, Eli Bronstein, Varun Tolani, and Claire J Tomlin. An efficient reachability-based framework for provably safe autonomous navigation in unknown environments. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, pages 1758–1765. IEEE, 2019.

[18] Somil Bansal, Mo Chen, Sylvia Herbert, and Claire J Tomlin. Hamilton-Jacobi reachability: A brief overview and recent advances. In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pages 2242–2253. IEEE, 2017.

[19] Osbert Bastani. Safe reinforcement learning with nonlinear dynamics via model predictive shielding. In *American Control Conference (ACC)*, pages 3488–3494. IEEE, 2021.

[20] Felix Berkenkamp, Riccardo Moriconi, Angela P Schoellig, and Andreas Krause. Safe learning of regions of attraction for uncertain, nonlinear systems with Gaussian processes. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, pages 4661–4666. IEEE, 2016.

[21] Felix Berkenkamp and Angela P Schoellig. Safe and robust learning control with Gaussian processes. In *European Control Conference (ECC)*, pages 2496–2501, 2015.

[22] Dimitri Bertsekas. Infinite time reachability of state-space regions by using feedback control. *IEEE Transactions on Automatic Control*, 17(5):604–613, 1972.

[23] Christopher M Bishop and Nasser M Nasrabadi. *Pattern Recognition and Machine Learning*, volume 4. Springer, 2006.

[24] Franco Blanchini. Set invariance in control. *Automatica*, 35(11):1747–1767, 1999.

[25] Franco Blanchini, Stefano Miani, et al. *Set-theoretic methods in control*, volume 78. Springer, 2008.

[26] Francesco Borrelli, Alberto Bemporad, and Manfred Morari. *Predictive control for linear and hybrid systems*. Cambridge University Press, 2017.

[27] Stephen P Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[28] Michael S Branicky. Multiple Lyapunov functions and other analysis tools for switched and hybrid systems. *IEEE Transactions on Automatic Control*, 43(4):475–482, 1998.

[29] Roger W Brockett et al. Asymptotic stability and feedback stabilization. *Differential Geometric Control Theory*, 27(1):181–191, 1983.

[30] Lukas Brunke, Siqi Zhou, and Angela P Schoellig. Barrier Bayesian linear regression: Online learning of control barrier conditions for safety-critical control of uncertain systems. In *Learning for Dynamics and Control Conference*, pages 881–892, 2022.

[31] Florian D Brunner, Mircea Lazar, and Frank Allgöwer. Stabilizing linear model predictive control: On the enlargement of the terminal set. In *2013 European Control Conference (ECC)*, pages 511–517. IEEE, 2013.

[32] Linda G Bushnell, Dawn M Tilbury, and SS Sastry. Steering three-input nonholonomic systems: the fire truck example. *The International Journal of Robotics Research*, 14(4):366–381, 1995.

[33] Fernando Castaneda, Jason J Choi, Bike Zhang, Claire J Tomlin, and Koushil Sreenath. Gaussian process-based min-norm stabilizing controller for control-affine systems with uncertain input effects and dynamics. In *American Control Conference (ACC)*, pages 3683–3690, 2021.

[34] Fernando Castañeda, Jason J Choi, Bike Zhang, Claire J Tomlin, and Koushil Sreenath. Pointwise feasibility of Gaussian process-based safety-critical control under model uncertainty. In *IEEE Conference on Decision and Control (CDC)*, pages 6762–6769, 2021.

[35] Mo Chen, Sylvia L Herbert, Haimin Hu, Ye Pu, Jaime Fernandez Fisac, Somil Bansal, SooJean Han, and Claire J Tomlin. Fastrack: a modular framework for real-time motion planning and guaranteed safe tracking. *IEEE Transactions on Automatic Control*, 66(12):5861–5876, 2021.

[36] Mo Chen, Sylvia L Herbert, Mahesh S Vashishtha, Somil Bansal, and Claire J Tomlin. Decomposition of reachable sets and tubes for a class of nonlinear systems. *IEEE Transactions on Automatic Control*, 63(11):3675–3688, 2018.

[37] Yuxiao Chen, Huei Peng, and Jessy Grizzle. Obstacle avoidance for low-speed autonomous vehicles with barrier function. *IEEE Transactions on Control Systems Technology*, 26(1):194–206, 2018.

[38] Richard Cheng, Mohammad Javad Khojasteh, Aaron D Ames, and Joel W Burdick. Safe multi-agent interaction through robust control barrier functions with learned uncertainties. In *IEEE Conference on Decision and Control (CDC)*, pages 777–783, 2020.

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

[40] Jason Choi, Fernando Castaneda, Claire J Tomlin, and Koushil Sreenath. Reinforcement learning for safety-critical control under model uncertainty, using control Lyapunov functions and control barrier functions. *arXiv preprint arXiv:2004.07584*, 2020.

[41] Andrew Clark. Control barrier functions for stochastic systems. *Automatica*, 130:109688, 2021.

[42] Andrew Clark. Verification and synthesis of control barrier functions. In *2021 60th IEEE Conference on Decision and Control (CDC)*, pages 6105–6112. IEEE, 2021.

[43] Wenceslao Shaw Cortez, Denny Oetomo, Chris Manzie, and Peter Choong. Control barrier functions for mechanical systems: Theory and application to robotic grasping. *IEEE Transactions on Control Systems Technology*, 29(2):530–545, 2019.

[44] Ryan K Cosner, Andrew W Singletary, Andrew J Taylor, Tamas G Molnar, Katherine L Bouman, and Aaron D Ames. Measurement-robust control barrier functions: Certainty in safety with uncertainty in state. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6286–6291. IEEE, 2021.

[45] Michael G Crandall, Lawrence C Evans, and P-L Lions. Some properties of viscosity solutions of Hamilton-Jacobi equations. *Transactions of the American Mathematical Society*, 282(2):487–502, 1984.

[46] Noel Csomay-Shanklin, Ryan K Cosner, Min Dai, Andrew J Taylor, and Aaron D Ames. Episodic learning for safe bipedal locomotion with control barrier functions and projection-to-state safety. In *Learning for Dynamics and Control Conference*, pages 1041–1053. PMLR, 2021.

[47] Jamal Daafouz, Pierre Riedinger, and Claude Iung. Stability analysis and control synthesis for switched systems: a switched Lyapunov function approach. *IEEE Transactions on Automatic Control*, 47(11):1883–1887, 2002.

[48] Charles Dawson, Zengyi Qin, Sicun Gao, and Chuchu Fan. Safe nonlinear control using robust neural Lyapunov-barrier functions. In *Conference on Robot Learning*, pages 1724–1735. PMLR, 2022.

[49] Sarah Dean, Andrew Taylor, Ryan Cosner, Benjamin Recht, and Aaron Ames. Guaranteeing safety of learned perception modules via measurement-robust control barrier functions. In *Conference on Robot Learning*, pages 654–670. PMLR, 2021.

[50] Vikas Dhiman, Mohammad Javad Khojasteh, Massimo Franceschetti, and Nikolay Atanasov. Control barriers in Bayesian learning of system dynamics. *IEEE Transactions on Automatic Control*, 68(1):214–229, 2021.

[51] Steven Diamond and Stephen Boyd. Cvxpy: A python-embedded modeling language for convex optimization. *The Journal of Machine Learning Research*, 17(1):2909–2913, 2016.

[52] Leroy D'Souza and Yash Vardhan Pant. Stochastic hybrid model predictive control using Gaussian processes for systems with piecewise residual dynamics. In *American Control Conference (ACC)*, pages 2670–2677. IEEE, 2023.

[53] David D. Fan, Jennifer Nguyen, Rohan Thakker, Nikhilesh Alatur, Ali Akbar Agha-Mohammadi, and Evangelos A. Theodorou. Bayesian learning-based adaptive control for safety critical systems. *arXiv*, pages 4093–4099, 2019.

[54] David D Fan, Jennifer Nguyen, Rohan Thakker, Nikhilesh Alatur, Ali-akbar Agha-mohammadi, and Evangelos A Theodorou. Bayesian learning-based adaptive control for safety critical systems. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4093–4099, 2020.

[55] Anthony V Fiacco and Garth P McCormick. *Nonlinear programming: sequential unconstrained minimization techniques*. SIAM, 1990.

[56] Jaime F Fisac, Anayo K Akametalu, Melanie N Zeilinger, Shahab Kaynama, Jeremy Gillula, and Claire J Tomlin. A general safety framework for learning-based control in uncertain robotic systems. *IEEE Transactions on Automatic Control*, 64(7):2737–2752, 2018.

[57] Jaime F Fisac, Mo Chen, Claire J Tomlin, and S Shankar Sastry. Reach-avoid problems with time-varying dynamics, targets and constraints. In *Proceedings of the 18th International Conference on Hybrid Systems: Computation and Control*, pages 11–20, 2015.

[58] Carlos Flores and Xiao-Yun Lu. Robust preview-based tractor-trailer lateral control for lane keeping. In *2020 American Control Conference (ACC)*, pages 1229–1234. IEEE, 2020.

[59] Jodi Forlizzi and Carl DiSalvo. Service robots in the domestic environment: a study of the roomba vacuum in the home. In *Proceedings of the 1st ACM SIGCHI/SIGART Conference on Human-Robot Interaction*, pages 258–265, 2006.

[60] Randy Freeman and Petar V Kokotovic. *Robust nonlinear control design: state-space and Lyapunov techniques*. Springer Science & Business Media, 2008.

[61] Jacob Gardner, Geoff Pleiss, Kilian Q Weinberger, David Bindel, and Andrew G Wilson. Gpytorch: Blackbox matrix-matrix Gaussian process inference with gpu acceleration. *Advances in Neural Information Processing Systems*, 31, 2018.

[62] Kunal Garg and Dimitra Panagou. Control-Lyapunov and control-barrier functions based quadratic program for spatio-temporal specifications. In *IEEE Conference on Decision and Control (CDC)*, pages 1422–1429, 2019.

[63] Kunal Garg and Dimitra Panagou. Robust control barrier and control Lyapunov functions with fixed-time convergence guarantees. In *2021 American Control Conference (ACC)*, pages 2292–2297. IEEE, 2021.

[64] Kunal Garg, James Usevitch, Joseph Breeden, Mitchell Black, Devansh Agrawal, Hardik Parwana, and Dimitra Panagou. Advances in the theory of control barrier functions: Addressing practical challenges in safe control synthesis for autonomous and robotic systems. *Annual Reviews in Control*, 57:100945, 2024.

[65] Farid Gharagozloo, Barbara Tempesta, Mark Meyer, Duy Nguyen, Stephan Gruessner, and Jay Redan. *History of robotic surgery*. Springer, 2021.

[66] Paul Glotfelter, Jorge Cortés, and Magnus Egerstedt. Nonsmooth barrier functions with applications to multi-robot systems. *IEEE Control Systems Letters*, 1(2):310–315, 2017.

[67] Sergio Grammatico, Franco Blanchini, and Andrea Caiti. Control-sharing and merging control Lyapunov functions. *IEEE Transactions on Automatic Control*, 59(1):107–119, 2014.

[68] Ruben Grandia, Andrew J Taylor, Aaron D Ames, and Marco Hutter. Multi-layered safety for legged robots via control barrier functions and model predictive control. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8352–8358, 2021.

[69] AK Gupta and DK Nagar. *Matrix Variate Distributions*, volume 104. CRC Press, 1999.

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

[71] Jack K Hale. *Ordinary differential equations*. Courier Corp., 2009.

[72] 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)*, pages 4542–4548. IEEE, 2015.

[73] Alberto Isidori. *Nonlinear control systems: an introduction*. Springer, 1985.

[74] Alberto Isidori, ED Sontag, and M Thoma. *Nonlinear Control Systems*. Springer, 1995.

[75] Pushpak Jagtap, George J Pappas, and Majid Zamani. Control barrier functions for unknown nonlinear systems using Gaussian processes. In *IEEE Conference on Decision and Control (CDC)*, pages 3699–3704, 2020.

[76] Mrdjan Jankovic. Robust control barrier functions for constrained stabilization of nonlinear systems. *Automatica*, 96:359–367, 2018.

[77] Xu Jin, Shi-Lu Dai, Jianjun Liang, Dejun Guo, and Huan Tan. Constrained line-of-sight tracking control of a tractor-trailer mobile robot system with multiple constraints. In *2021 American Control Conference (ACC)*, pages 1046–1051, 2021.

[78] Thomas Kailath. *Linear systems*, volume 156. Prentice-Hall Englewood Cliffs, NJ, 1980.

[79] Pouya Kassaeiyan, Khalil Alipour, and Bahram Tarvirdizadeh. A full-state trajectory tracking controller for tractor-trailer wheeled mobile robots. *Mechanism and Machine Theory*, 150:103872, 2020.

[80] Erkan Kayacan, Joshua M Peschel, and Erdal Kayacan. Centralized, decentralized and distributed nonlinear model predictive control of a tractor-trailer system: A comparative study. In *2016 American control conference (ACC)*, pages 4403–4408. IEEE, 2016.

[81] Erkan Kayacan, Wouter Saeys, Herman Ramon, Calin Belta, and Joshua M. Peschel. Experimental validation of linear and nonlinear MPC on an articulated unmanned ground vehicle. *IEEE/ASME Transactions on Mechatronics*, 23(5):2023–2033, 2018.

[82] Hassan K Khalil. *Nonlinear Systems*. Prentice hall, 2002.

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

[84] Shishir Kolathaya and Aaron D Ames. Input-to-state safety with control barrier functions. *IEEE Control Systems Letters*, 3(1):108–113, 2018.

[85] Shreyas Kousik, Sean Vaskov, Fan Bu, Matthew Johnson-Roberson, and Ram Vasudevan. Bridging the gap between safety and real-time performance in receding-horizon trajectory design for mobile robots. *The International Journal of Robotics Research*, 39(12):1419–1469, 2020.

[86] Gerardo Lafferriere, George J Pappas, and Sergio Yovine. Symbolic reachability computation for families of linear vector fields. *Journal of Symbolic Computation*, 32(3):231–253, 2001.

[87] Vangipuram Lakshmikantham and Srinivasa Leela. *Differential and integral inequalities: theory and applications: volume I: ordinary differential equations*. Academic Press, 1969.

[88] Chiara Talignani Landi, Federica Ferraguti, Silvia Costi, Marcello Bonfè, and Cristian Secchi. Safety barrier functions for human-robot interaction with industrial manipulators. In *European Control Conference (ECC)*, pages 2565–2570, 2019.

[89] Negin Lashkari, Mohammad Biglarbegian, and Simon X Yang. Development of novel motion planning and controls for a series of physically connected robots in dynamic environments. *Journal of Intelligent and Robotic Systems*, 95(2):291–310, 2019.

[90] J-P Laumond. Controllability of a multibody mobile robot. *IEEE Transactions on Robotics and Automation*, 9(6):755–763, 1993.

[91] Jean-Paul Laumond et al. *Robot motion planning and control*, volume 229. Springer, 1998.

[92] Mircea Lazar and Martin Tetteroo. Computation of terminal costs and sets for discrete–time nonlinear MPC. *IFAC-PapersOnLine*, 51(20):141–146, 2018.

[93] Jean Levine. *Analysis and control of nonlinear systems: A flatness-based approach*. Springer Science & Business Media, 2009.

[94] Daniel Liberzon. *Switching in systems and control*, volume 190. Springer, 2003.

[95] Yuandan Lin and Eduardo D Sontag. Control-Lyapunov universal formulas for restricted inputs. *Control-Theory and Advanced Technology*, 10(4), 1995.

[96] Lars Lindemann, Alexander Robey, Lejun Jiang, Stephen Tu, and Nikolai Matni. Learning robust output control barrier functions from safe expert demonstrations. *arXiv preprint arXiv:2111.09971*, 2021.

[97] Haitao Liu, Jianfei Cai, and Yew-Soon Ong. Remarks on multi-output Gaussian process regression. *Knowledge-Based Systems*, 144:102–121, 2018.

[98] Jun Liu. Converse barrier functions via Lyapunov functions. *IEEE Transactions on Automatic Control*, 67(1):497–503, 2022.

[99] Jun Liu, Xinzhi Liu, and Wei-Chau Xie. Input-to-state stability of impulsive and switching hybrid systems with time-delay. *Automatica*, 47(5):899–908, 2011.

[100] Simin Liu, Changliu Liu, and John Dolan. Safe control under input limits with neural control barrier functions. In *Conference on Robot Learning*, pages 1970–1980. PMLR, 2023.

[101] Zhiyuan Liu, Ming Yue, Lie Guo, and Yongshun Zhang. Trajectory planning and robust tracking control for a class of active articulated tractor-trailer vehicle with on-axle structure. *European Journal of Control*, 54:87–98, 2020.

[102] Oskar Ljungqvist and Daniel Axehill. A predictive path-following controller for multi-steered articulated vehicles. *IFAC-PapersOnLine*, 53(2):15725–15732, 2020.

[103] Christos Louizos and Max Welling. Structured and efficient variational deep learning with matrix Gaussian posteriors. In *International Conference on Machine Learning*, pages 1708–1716. PMLR, 2016.

[104] John Lygeros. On reachability and minimum cost optimal control. *Automatica*, 40(6):917–927, 2004.

[105] Olvi L Mangasarian and Stan Fromovitz. The fritz john necessary optimality conditions in the presence of equality and inequality constraints. *Journal of Mathematical Analysis and Applications*, 17(1):37–47, 1967.

[106] Mathias Marley, Roger Skjetne, and Andrew R Teel. Synergistic control barrier functions with application to obstacle avoidance for nonholonomic vehicles. In *American Control Conference (ACC)*, pages 243–249. IEEE, 2021.

[107] Jorge L Martínez, Jesús Morales, Anthony Mandow, and Alfonso García-Cerezo. Steering limitations for a vehicle pulling passive trailers. *IEEE Transactions on Robotics*, 16(4):809–818, 2008.

[108] Pierre-François Massiani, Steve Heim, Friedrich Solowjow, and Sebastian Trimpe. Safe value functions. *arXiv preprint arXiv:2105.12204*, 2021.

[109] Yiming Meng and Jun Liu. Lyapunov-barrier characterization of robust reach–avoid–stay specifications for hybrid systems. *Nonlinear Analysis: Hybrid Systems*, 49:101340, 2023.

[110] Marcel Menner and Eugene Lavretsky. Translation of nagumo's foundational work on barrier functions: On the location of integral curves of ordinary differential equations. *arXiv preprint arXiv:2406.18614*, 2024.

[111] Benjamin Morris, Matthew J Powell, and Aaron D Ames. Sufficient conditions for the Lipschitz continuity of qp-based multi-objective control of humanoid robots. In *52nd IEEE Conference on Decision and Control*, pages 2920–2926. IEEE, 2013.

[112] Benjamin J Morris, Matthew J Powell, and Aaron D Ames. Continuity and smoothness properties of nonlinear optimization-based feedback controllers. In *2015 54th IEEE Conference on Decision and Control (CDC)*, pages 151–158. IEEE, 2015.

[113] Mitio Nagumo. Über die lage der integralkurven gewöhnlicher differentialgleichungen. *Proceedings of the Physico-Mathematical Society of Japan. 3rd Series*, 24:551–559, 1942.

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

[115] Tudor Andrei Oancea, Yuning Jiang, and Colin N Jones. Relaxed recentered log-barrier function based nonlinear model predictive control. In *2023 European Control Conference (ECC)*, pages 1–6. IEEE, 2023.

[116] Dimitra Panagou, Dušan M Stipanovič, and Petros G Voulgaris. Multi-objective control for multi-agent systems using Lyapunov-like barrier functions. In *52nd IEEE Conference on Decision and Control (CDC)*, pages 1478–1483. IEEE, 2013.

[117] Stephen Prajna. Barrier certificates for nonlinear model validation. *Automatica*, 42(1):117–126, 2006.

[118] Stephen Prajna and Ali Jadbabaie. Safety verification of hybrid systems using barrier certificates. In *International Workshop on Hybrid Systems: Computation and Control*, pages 477–492. Springer, 2004.

[119] Zengyi Qin, Dawei Sun, and Chuchu Fan. Sablas: Learning safe control for black-box dynamical systems. *IEEE Robotics and Automation Letters*, 7(2):1928–1935, 2022.

[120] Saša V Raković, Sixing Zhang, Haidi Sun, and Yuanqing Xia. Model predictive control for linear systems under relaxed constraints. *IEEE Transactions on Automatic Control*, 68(1):369–376, 2021.

[121] Megnath Ramesh, Frank Imeson, Baris Fidan, and Stephen L Smith. Optimal partitioning of non-convex environments for minimum turn coverage planning. *IEEE Robotics and Automation Letters*, 7(4):9731–9738, 2022.

[122] Manuel Rauscher, Melanie Kimmel, and Sandra Hirche. Constrained robot control using control barrier functions. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 279–285. IEEE, 2016.

[123] James Blake Rawlings, David Q Mayne, Moritz Diehl, et al. *Model predictive control: theory, computation, and design*, volume 2. Nob Hill Publishing Madison, WI, 2017.

[124] Paul Ritzen, Erik Roebroek, Nathan Van De Wouw, Zhong Ping Jiang, and Henk Nijmeijer. Trailer steering control of a tractor-trailer robot. *IEEE Transactions on Control Systems Technology*, 24(4):1240–1252, 2016.

[125] Alexander Robey, Haimin Hu, Lars Lindemann, Hanwen Zhang, Dimos V Dimarogonas, Stephen Tu, and Nikolai Matni. Learning control barrier functions from expert demonstrations. In *2020 59th IEEE Conference on Decision and Control (CDC)*, pages 3717–3724. IEEE, 2020.

[126] Stephen M Robinson. *Generalized equations and their solutions, Part I: Basic theory*. Springer, 1979.

[127] Muhammad Zakiyullah Romdlony and Bayu Jayawardhana. Uniting control Lyapunov and control barrier functions. In *IEEE Conference on Decision and Control (CDC)*, pages 2293–2298, 2014.

[128] Muhammad Zakiyullah Romdlony and Bayu Jayawardhana. On the new notion of input-to-state safety. In *2016 IEEE 55th conference on decision and control (CDC)*, pages 6403–6409. IEEE, 2016.

[129] Muhammad Zakiyullah Romdlony and Bayu Jayawardhana. Stabilization with guaranteed safety using control Lyapunov-barrier function. *Automatica*, 66:39–47, 2016.

[130] Pierre Rouchon, Michel Fliess, Jean Lévine, and Philippe Martin. Flatness and motion planning: the car with n trailers. In *European Control Conference (ECC)*, volume 93, pages 1518–1522, 1993.

[131] Ji-Chul Ryu, Sunil K Agrawal, and Jaume Franch. Motion planning and control of a tractor with a steerable trailer using differential flatness. *Journal of Computational and Nonlinear Dynamics*, 2008.

[132] Ji Chul Ryu, Sunil K. Agrawal, and Jaume Franch. Motion planning and control of a tractor with a steerable trailer using differential flatness. *Journal of Computational and Nonlinear Dynamics*, 3(3):1–8, 2008.

[133] Cesar Santoyo, Maxence Dutreix, and Samuel Coogan. Verification and control for finite-time safety of stochastic systems via barrier functions. In *2019 IEEE Conference on Control Technology and Applications (CCTA)*, pages 712–717. IEEE, 2019.

[134] Shankar Sastry. *Nonlinear systems: analysis, stability, and control*, volume 10. Springer Science & Business Media, 2013.

[135] Ian Savage. Comparing the fatality risks in united states transportation across modes and over time. *Research in Transportation Economics*, 43(1):9–22, 2013.

[136] Matteo Saveriano and Dongheui Lee. Learning barrier functions for constrained motion planning with dynamical systems. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 112–119. IEEE, 2019.

[137] Peter Seiler, Mrdjan Jankovic, and Erik Hellstrom. Control barrier functions with unmodeled input dynamics using integral quadratic constraints. *IEEE Control Systems Letters*, 6:1664–1669, 2021.

[138] Mohamed Serry, Haocheng Chang, and Jun Liu. Reach-avoid control synthesis for a quadrotor uav with formal safety guarantees. *arXiv preprint arXiv:2405.20502*, 2024.

[139] Eduardo D Sontag. Feedback stabilization of nonlinear systems. In *Robust Control of Linear Systems and Nonlinear Control: Proceedings of the International Symposium MTNS-89, Volume II*, pages 61–81. Springer, 1990.

[140] Eduardo D Sontag and Yuan Wang. On characterizations of the input-to-state stability property. *Systems & Control Letters*, 24(5):351–359, 1995.

[141] Niranjan Srinivas, Andreas Krause, Sham M Kakade, and Matthias Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. *arXiv preprint arXiv:0912.3995*, 2009.

[142] Dušan M Stipanović, Claire J Tomlin, and George Leitmann. Monotone approximations of minimum and maximum functions and multi-objective problems. *Applied Mathematics and Optimization*, 66(3):455–473, 2012.

[143] Shengyang Sun, Changyou Chen, and Lawrence Carin. Learning structured weight uncertainty in Bayesian neural networks. In *Artificial Intelligence and Statistics*, pages 1283–1292. PMLR, 2017.

[144] Rin Takano and Masaki Yamakita. Robust control barrier function for systems affected by a class of mismatched disturbances. *SICE Journal of Control, Measurement, and System Integration*, 13(4):165–172, 2020.

[145] Xiao Tan, Wenceslao Shaw Cortez, and Dimos V Dimarogonas. High-order barrier functions: Robustness, safety, and performance-critical control. *IEEE Transactions on Automatic Control*, 67(6):3021–3028, 2021.

[146] Yucheng Tang, Ilshat Mamaev, Jing Qin, Christian Wurll, and Björn Hein. Reachability-aware collision avoidance for tractor-trailer system with non-linear MPC and control barrier function. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 274–281. IEEE, 2023.

[147] Andrew Taylor, Andrew Singletary, Yisong Yue, and Aaron Ames. Learning for safety-critical control with control barrier functions. In *Learning for Dynamics and Control Conference*, pages 708–717, 2020.

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

[149] Andrew J Taylor, Andrew Singletary, Yisong Yue, and Aaron D Ames. A control barrier perspective on episodic learning via projection-to-state safety. *IEEE Control Systems Letters*, 5(3):1019–1024, 2020.

[150] Ashish Tiwari. Approximate reachability for linear systems. In *International Workshop on Hybrid Systems: Computation and Control*, pages 514–525. Springer, 2003.

[151] Jonas Umlauft, Thomas Beckers, Melanie Kimmel, and Sandra Hirche. Feedback linearization using Gaussian processes. In *IEEE Conference on Decision and Control (CDC)*, pages 5249–5255, 2017.

[152] 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)*, pages 2044–2049. IEEE, 2020.

[153] Kim P Wabersich, Florian A Bayer, Matthias A Müller, and Frank Allgüwer. Economic model predictive control for robust periodic operation with guaranteed closed-loop performance. In *2018 European Control Conference (ECC)*, pages 507–513. IEEE, 2018.

[154] Kim P Wabersich, Andrew J Taylor, Jason J Choi, Koushil Sreenath, Claire J Tomlin, Aaron D Ames, and Melanie N Zeilinger. Data-driven safety filters: Hamilton-Jacobi reachability, control barrier functions, and predictive methods for uncertain systems. *IEEE Control Systems Magazine*, 43(5):137–177, 2023.

[155] Kim P Wabersich and Melanie N Zeilinger. Linear model predictive safety certification for learning-based control. In *2018 IEEE Conference on Decision and Control (CDC)*, pages 7130–7135. IEEE, 2018.

[156] Kim Peter Wabersich and Melanie N Zeilinger. A predictive safety filter for learning-based control of constrained nonlinear dynamical systems. *Automatica*, 129:109597, 2021.

[157] Chuanzheng Wang, Yiming Meng, Yinan Li, Stephen L Smith, and Jun Liu. Learning control barrier functions with high relative degree for safety-critical control. In *European Control Conference (ECC)*, pages 1459–1464, 2021.

[158] Chuanzheng Wang, Yiming Meng, Stephen L Smith, and Jun Liu. Safety-critical control of stochastic systems using stochastic control barrier functions. In *IEEE Conference on Decision and Control (CDC)*, pages 5924–5931, 2021.

[159] Li Wang, Aaron Ames, and Magnus Egerstedt. Safety barrier certificates for heterogeneous multi-robot systems. In *2016 American Control Conference (ACC)*, pages 5213–5218. IEEE, 2016.

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

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

[162] Li Wang, Aaron D. Ames, and Magnus Egerstedt. Safety barrier certificates for collisions-free multirobot systems. *IEEE Transactions on Robotics*, 33(3):661–674, 2017.

[163] Li Wang, Dongkun Han, and Magnus Egerstedt. Permissive barrier certificates for safe stabilization using sum-of-squares. In *2018 Annual American Control Conference (ACC)*, pages 585–590. IEEE, 2018.

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

[165] Xiao Wang. Ensuring safety of learning-based motion planners using control barrier functions. *IEEE Robotics and Automation Letters*, 7(2):4773–4780, 2022.

[166] Peter Wieland and Frank Allgöwer. Constructive safety using control barrier functions. *IFAC Proceedings Volumes*, 40(12):462–467, 2007.

[167] Christopher KI Williams and Carl Edward Rasmussen. *Gaussian Processes for Machine Learning*, volume 2. MIT Press, 2006.

[168] Wei Xiao and Calin Belta. Control barrier functions for systems with high relative degree. In *2019 IEEE 58th conference on decision and control (CDC)*, pages 474–479. IEEE, 2019.

[169] Wei Xiao and Calin Belta. High order control barrier functions. *IEEE Transactions on Automatic Control*, 67(7):3655–3662, 2022.

[170] Wei Xiao, Calin Belta, and Christos G Cassandras. Adaptive control barrier functions. *IEEE Transactions on Automatic Control*, 67(5):2267–2281, 2022.

[171] Wei Xiao, Calin A Belta, and Christos G Cassandras. Sufficient conditions for feasibility of optimal control problems using control barrier functions. *Automatica*, 135:109960, 2022.

[172] Xiangru Xu. Constrained control of input–output linearizable systems using control sharing barrier functions. *Automatica*, 87:195–201, 2018.

[173] Xiangru Xu, Paulo Tabuada, Jessy W Grizzle, and Aaron D Ames. Robustness of control barrier functions for safety critical control. *IFAC-PapersOnLine*, 48(27):54–61, 2015.

[174] Xiangru Xu, Thomas Waters, Daniel Pickem, Paul Glotfelter, Magnus Egerstedt, Paulo Tabuada, Jessy W Grizzle, and Aaron D Ames. Realizing simultaneous lane keeping and adaptive speed regulation on accessible mobile robot testbeds. In *2017 IEEE Conference on Control Technology and Applications (CCTA)*, pages 1769–1775. IEEE, 2017.

[175] Jing Yuan. Hierarchical Motion Planning for Multisteering Tractor-Trailer Mobile Robots with On-Axle Hitching. *IEEE/ASME Transactions on Mechatronics*, 22(4):1652–1662, 2017.

[176] Jing Yuan, Shaokun Yang, and Jingxin Cai. Consistent path planning for on-axle-hitching multisteering trailer systems. *IEEE Transactions on Industrial Electronics*, 65(12):9625–9634, 2018.

[177] Jun Zeng, Bike Zhang, and Koushil Sreenath. Safety-critical model predictive control with discrete-time control barrier function. In *2021 American Control Conference (ACC)*, pages 3882–3889. IEEE, 2021.

[178] Hongchao Zhao, Wen Chen, Shunbo Zhou, Zhe Liu, Fan Zheng, and Yun-Hui Liu. Online trajectory planning for an industrial tractor towing multiple full trailers. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6089–6095, 2020.

Appendix A

Appendix

A.1 Tractor and Trailer's Positional Safety Constraint

We need to obtain the following derivatives of $h_{1k}(\boldsymbol{x})$ to assemble (4.45). Substituting the derivatives of the position variables x_1 and y_1 from (4.41) up to order three yields

$$\begin{aligned}
\dot{h}_{1k}(\boldsymbol{x}) &= 2v[d_k^x c\theta + d_k^y s\theta], \\
\ddot{h}_{1k}(\boldsymbol{x}) &= 2v^2 + 2a[d_k^x c\theta + d_k^y s\theta] - \\
&\quad 2\frac{v^2}{l_1} t\delta_1[d_k^x s\theta - d_k^y c\theta], \\
\dddot{h}_{1k}(\boldsymbol{x}, \boldsymbol{u}) &= 6va - \frac{6va}{l_1} t\delta_1[d_k^x s\theta - d_k^y c\theta] - 2\frac{v^3}{l_1^2} t^2\delta_1 \\
&\quad [d_k^x c\theta + d_k^y s\theta] + 2[d_k^x c\theta + d_k^y s\theta]J - \\
&\quad 2\frac{v^2}{l_1} sec^2\delta_1[d_k^x s\theta - d_k^y c\theta]\omega_1.
\end{aligned}$$

Similarly, the required derivatives of $h_{2k}(\boldsymbol{x})$ are obtained by considering (4.42) and substituting the derivatives of x_2 and y_2 up to order two in terms of the tractor's reference point

coordinates as follows

$$
\begin{aligned}
\dot{h}_{2k}(\boldsymbol{x}) &= 2v[d_k^x c\theta + d_k^y s\theta] - 2l_2 v c\psi + 2v(t\delta_2 c\psi + s\psi) \\
&\quad [d_k^x s(\theta-\psi) + d_k^y c(\theta-\psi)], \\
\ddot{h}_{2k}(\boldsymbol{x},\boldsymbol{u}) &= 2v^2 - 2v^2(t\delta_2 c\psi + s\psi)s\psi - 2l_2 a c\psi + \\
&\quad 2l_2 v s\psi[\frac{v}{l_1}t\delta_1 - \frac{v}{l_2}(t\delta_2 c\psi + s\psi)] + \\
&\quad 2a[d_k^x c\theta + d_k^y s\theta] - 2\frac{v^2}{l_1}t\delta_1[d_k^x s\theta - d_k^y c\theta] + \\
&\quad 2a(t\delta_2 c\psi + s\psi)[d_k^x s(\theta-\psi) - d_k^y c(\theta-\psi)] - \\
&\quad 2v(t\delta_2 s\psi - c\psi)[\frac{v}{l_1}t\delta_1 - \frac{v}{l_2}(t\delta_2 c\psi + s\psi)] \\
&\quad [d_k^x s(\theta-\psi) - d_k^y c(\theta-\psi)] + \\
&\quad \frac{2v^2}{l_2}(t\delta_2 c\psi + s\psi)^2[d_k^x c(\theta-\psi) + d_k^y s(\theta-\psi)] + \\
&\quad 2v\sec^2\delta_2 c\psi[d_k^x s(\theta-\psi) - d_k^y c(\theta-\psi)]\omega_2,
\end{aligned}
$$

where $s(\cdot) = \sin(\cdot)$, $c(\cdot) = \cos(\cdot)$, and $t(\cdot) = \tan(\cdot)$. We also denote $(x_1 - x_k^o)$ by d_k^x and $(y_1 - y_k^o)$ by d_k^y for $k \in \{1, \ldots, N_o\}$, respectively.

A.2 Discrete Linear Time-varying System

$$
A_k = \begin{bmatrix}
1 & 0 & \cos\theta_r T_s & 0 & -v_r \sin\theta_r T_s & 0 & 0 & 0 \\
0 & 1 & \sin\theta_r T_s & 0 & v_r \cos\theta_r T_s & 0 & 0 & 0 \\
0 & 0 & 1 & T_s & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & \frac{\tan\delta_{1r}}{l_1} T_s & 0 & 1 & 0 & a_{57} & 0 \\
0 & 0 & a_{63} & 0 & 0 & a_{66} & a_{67} & a_{68} \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix},
$$

where

$$
\begin{aligned}
a_{57} &= \frac{v_r}{l_1}(1+\tan^2\delta_{1r})T_s, \\
a_{63} &= (\frac{\tan\delta_{1r}}{l_1} - \frac{(\tan\delta_{2r}\cos\psi_r + \sin\psi_r)}{l_2})T_s, \\
a_{66} &= 1 - \frac{v_r}{l_2}(-\tan\delta_{2r}\sin\psi_r + \cos\psi_r)T_s, \\
a_{67} &= \frac{v_r}{l_1}(1+\tan^2\delta_{1r})T_s, \\
a_{68} &= -\frac{v_r}{l_2}(1+\tan^2\delta_{2r})\cos\psi_r T_s.
\end{aligned}
$$

Also, we have

$$
B_k = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ T_s & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & T_s & 0 \\ 0 & 0 & T_s \end{bmatrix}, C_k = \begin{bmatrix} v_r\theta_r\sin\theta_r \\ -v_r\theta_r\cos\theta_r \\ 0 \\ 0 \\ c_{51} \\ c_{61} \\ 0 \\ 0 \end{bmatrix},
$$

where $c_{51} = -\frac{v_r}{l_1}(1+\tan^2\delta_{1r})\delta_{1r}T_s$, and $c_{61} = \delta_{1r})\delta_{1r} + \frac{v_r}{l_2}\cos\psi_r(1+\tan^2\delta_{2r})\delta_{2r})T_s$.